CONVS2S-VC: FULLY CONVOLUTIONAL SEQUENCE-TO-SEQUENCE VOICE CONVERSION

Hirokazu Kameoka, Kou Tanaka, Takuhiro Kaneko, Nobukatsu Hojo

NTT Communication Science Laboratories, NTT Corporation
3-1 Morinosato Wakamiya, Atsugi, Kanagawa, Japan

ABSTRACT

This paper proposes a voice conversion method based on fully convolutional sequence-to-sequence (seq2seq) learning. The present method, which we call "ConvS2S-VC", learns the mapping between source and target speech feature sequences using a fully convolutional seq2seq model with an attention mechanism. Owing to the nature of seq2seq learning, our method is particularly noteworthy in that it allows the flexible conversion of not only the voice characteristics but also the pitch contour and duration of the input speech. The current model consists of six networks, namely source and target encoders, a target decoder, source and target reconstructors and a postnet, which are designed using dilated causal convolution networks with gated linear units. Subjective evaluation experiments revealed that the proposed method obtained higher sound quality and speaker similarity than a baseline method.

Index Terms— Voice conversion, sequence-to-sequence learning, attention, fully convolutional network

1. INTRODUCTION

Voice conversion (VC) is a technique for converting para/non-linguistic information contained in a given utterance such as the perceived identity of a speaker while preserving linguistic information. Potential applications of this technique include speaker-identity modification for text-to-speech (TTS) systems [1], speaking aids [2, 3], speech enhancement [4–6], and pronunciation conversion [7].

Many conventional VC methods are designed to use parallel utterances of source and target speech to train acoustic models for feature mapping. A typical pipeline of the training process consists of extracting acoustic features from source and target utterances, performing dynamic time warping (DTW) to obtain time-aligned parallel data, and training an acoustic model that maps the source features to the target features frame-by-frame. Examples of the acoustic model include Gaussian mixture models (GMM) [8–10] and deep neural networks (DNNs) [7, 11–14]. Some attempts have also been made to develop methods that require no parallel utterances, transcriptions, or time alignment procedures. Recently, deep generative models such as variational autoencoders (VAEs), cycle-consistent generative adversarial networks (CycleGAN), and star generative adversarial networks (StarGAN) have been employed with notable success for non-parallel VC tasks [15–19].

One limitation of conventional methods including those mentioned above is that they are mainly focused on learning to convert only the spectral features frame-by-frame and are less focused on converting prosodic features such as the fundamental frequency (F_0) contour, duration and rhythm of the input speech. In particular, with most methods, the entire F_0 contour is simply adjusted using a linear transformation in the logarithmic domain while the duration and rhythm are usually kept unchanged. However, since these features play as important a role as spectral features in characterizing speaker identities and speaking styles, it would be desirable if these features could also be converted more flexibly. To overcome this limitation, this paper proposes adopting a sequence-to-sequence (seq2seq) learning approach.

The seq2seq learning approach offers a general and powerful framework for transforming one sequence into another variable length sequence [20, 21]. This is made possible by using encoder and decoder networks, where the encoder encodes an input sequence to an internal representation whereas the decoder generates an output sequence according to the internal representation. The original seq2seq model employs recurrent neural networks (RNNs) to model the encoder and decoder networks, where popular choices for the RNN architectures involve long short-term memory (LSTM) networks and gated recurrent units (GRU). This approach has attracted a lot of attention in recent years after being introduced and applied with notable success in various tasks such as machine translation in the field of natural language processing. It has also been successfully adopted in state-of-the-art automatic speech recognition (ASR) systems (e.g., [21]) and TTS systems [22–28].

One problem as regards the original seq2seq model is that it suffers from the constraint that all input sequences are forced to be encoded into a fixed length internal vector. This can limit the ability of the model especially when it comes to long input sequences, such as long sentences in text translation problems. To overcome this limitation, a mechanism called "attention" [29] has been introduced, which allows the network to learn where to pay attention in the input sequence for each item in the output sequence.

Another potential weakness with the original seq2seq model is that training RNNs can be costly and time-consuming since they are unsuitable for parallel computations using GPUs. While RNNs are indeed a natural choice for modeling long sequential data, recent work has shown that CNNs with gating mechanisms also have excellent potential for capturing long-term dependencies [30, 31]. In addition, they are more suitable than RNNs for parallel computations. To exploit this advantage of CNNs, a seq2seq model that adopts a fully convolutional architecture was recently proposed [32]. With this model, the decoder is designed using causal convolutions so that it allows the model to generate an output sequence in an autoregressive manner. This model with an attention mechanism and called the "ConvS2S" model has already been applied and shown to work well in machine translation tasks [32] and TTS [26, 27]. It has also been shown that it can be trained more efficiently than its RNN counterpart.

Inspired by the success of the ConvS2S model in TTS tasks, in this paper we propose a VC method based on the ConvS2S model, which we call "ConvS2S-VC", along with an architecture tailored for use with VC. In addition, we report some of the implementation details that we have found particularly useful in practice.

2. RELATED WORK

It should be noted that some attempts have already been made to apply seq2seq models to VC problems. Miyoshi et al. proposed an acoustic model combining recognition, synthesis and seq2seq models [33]. The recognition and synthesis models can be thought of as ASR and TTS modules, where the recognition model is used to convert a source speech feature sequence into a sequence of context posterior probabilities. An LSTM-based seq2seq model is used to convert the context posterior probability sequence of the source speech into that of target speech and finally the synthesis model is used to generate a target speech feature sequence according to the converted context posterior probability sequence. Since this model relies on the ASR module to ensure that the contextual information of the source speech will be preserved after conversion, the downside is that it requires text annotations for model training in addition to parallel utterances and can fail to work if the ASR module does not function reliably.

Our method differs from the above method in three major respects: First, our model includes an attention mechanism. Secondly, we designed our model to be fully convolutional, and so we hope it can be trained efficiently. Thirdly, it allows the direct conversion of a source speech feature sequence without relying on ASR modules and requires no text annotations for model training, thanks to our newly introduced idea of context preservation loss [34].

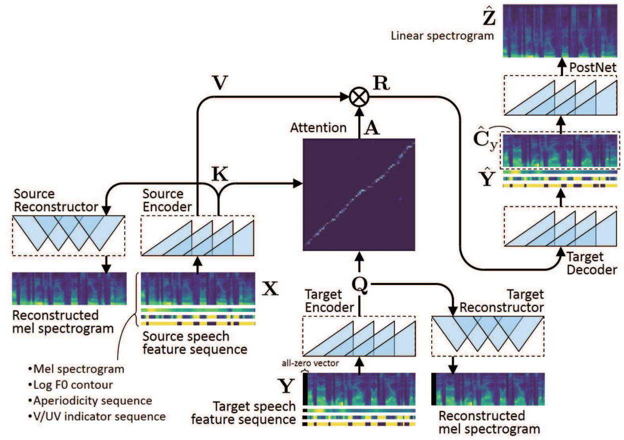

Fig. 1. Model architecture of the present ConvS2S model.

3. CONVS2S-VC

The present model consists of two networks, ConversionNet and PostNet. ConversionNet is a seq2seq model that maps a source speech feature sequence to a target speech feature sequence, whereas PostNet restores the linear-frequency-scaled spectral envelope sequence from its mel-frequency-scaled version included in the converted feature sequence. The overall architecture of our model is illustrated in Fig. 1.

3.1. Feature extraction and normalization

We use the WORLD analyzer [35] to compute linear-frequency-scaled spectral envelope sequences (hereafter referred to as linear spectrograms). For the feature sequence, we use a concatenation of a mel-frequency-scaled (compressed) version of the linear spectrogram (hereafter referred to as a mel spectrogram), a log F_0 contour, an aperiodicity sequence, and a voiced/unvoiced indicator sequence. Here, the log F_0 contour is assumed to be filled with smoothly interpolated values in unvoiced segments. In our preliminary experiments, we also tried appending the sinusoidal position encodings introduced in [36] to the feature vector, however, it tended to lead to poorer performance.

We normalize the linear and mel spectrograms and the log F_0 contour as follows to ensure that each element lies within the range $[0, 1]$:

$$z_{i,n} \leftarrow (z_{i,n}/\{\max_{i',n'} z_{i',n'}\})^{\gamma}, \tag{1}$$
$$c_{j,n} \leftarrow (c_{j,n}/\{\max_{j',n'} c_{j',n'}\})^{\gamma}, \tag{2}$$
$$f_n \leftarrow (f_n - f_{\min})/(f_{\max} - f_{\min}), \tag{3}$$

where $1 \leq i \leq I$ and $1 \leq j \leq J$ denote the frequency indices, n denotes the frame index, and $z_{i,n}$, $c_{j,n}$ and f_n denote elements of the linear and mel spectrograms and the log F_0 contour of a particular utterance. Here, we set γ, $f_{\max}$ and $f_{\min}$ to 0.3, $\log(500)$ and $\log(50)$, respectively.

3.2. Model

We hereafter use $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M] \in \mathbb{R}^{D \times M}$ to denote the source and target speech feature sequences of parallel utterances. ConversionNet is a seq2seq model that aims to map $\mathbf{X}$ to $\mathbf{Y}$. Our model is inspired by and built upon the models presented in [26, 36], with the difference being that it involves two additional networks, called source and target reconstructors. These networks play an important role in ensuring that the encoders preserve contextual (phoneme) information about the source and target speech, as explained below. ConversionNet thus consists of five networks, namely source and target encoders, a target decoder, and source and target reconstructors.

As with many seq2seq models, ConversionNet has an encoder-decoder structure. Here, the source encoder takes $\mathbf{X}$ as the input and produces two internal vector sequences $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{d \times N}$, whereas the target encoder takes $\mathbf{Y}$ as the input and produces an internal vector sequence $\mathbf{Q} \in \mathbb{R}^{d \times M}$:

$$(\mathbf{K}, \mathbf{V}) = \mathsf{SrcEnc}(\mathbf{X}), \tag{4}$$

$$\mathbf{Q} = \mathsf{TrgEnc}(\mathbf{Y}), \tag{5}$$

where $\mathbf{K}$, $\mathbf{V}$ and $\mathbf{Q}$ are called *key*, *value* and *query*, respectively, and d denotes the dimension of the internal vectors. We now define an attention matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$ as the product of $\mathbf{K}$ and $\mathbf{Q}$ divided by $\sqrt{d}$ and followed by a softmax operation:

$$\mathbf{A} = \mathsf{softmax}_n\big(\mathbf{K}^\mathsf{T}\mathbf{Q}/\sqrt{d}\big), \tag{6}$$

where $\mathsf{softmax}_n$ denotes a softmax operation performed on the n-axis. $\mathbf{A}$ can be thought of as a similarity matrix, where the (n, m)-th element is expected to indicate the similarity between the n-th and m-th frames of source and target speech. The peak trajectory of $\mathbf{A}$ can thus be interpreted as a time-warping function that associates the frames of the source speech with those of the target speech. The time-warped version of $\mathbf{V}$ can thus be written as

$$\mathbf{R} = \mathbf{A}\mathbf{V}, \tag{7}$$

which will be passed to the target decoder to generate an output sequence:

$$\hat{\mathbf{Y}} = \mathsf{TrgDec}(\mathbf{R}). \tag{8}$$

Since the target speech feature sequence $\mathbf{Y}$ is of course not accessible at test time, we would want to use a feature vector that the target decoder has generated as the input to the target encoder for the next time step so that feature vectors can be generated one-by-one in a recursive manner. To allow the model to behave in this way, first we must take care that the target encoder and decoder must not use future information when producing an output vector at each time step. This can be ensured by simply constraining the convolution layers in the target encoder and decoder to be causal. Note that causal convolution can be easily implemented by padding the input by $\delta(k-1)$ elements on both the left and right side with zero vectors and remove $\delta(k-1)$ elements from the end of the convolution output, where k is the kernel size and δ is the dilation factor. Secondly, the output sequence $\hat{\mathbf{Y}}$ must correspond to a time-shifted version of $\mathbf{Y}$ so that at each time step the decoder will be able to predict the target speech feature vector that is likely to be generated at the next time step. To this end, we include an L_1 loss

$$L_{\mathsf{dec}} = \|\hat{\mathbf{Y}}_{1:D,1:M-1} - \mathbf{Y}_{1:D,2:M}\|_1 \tag{9}$$

in the training loss to be minimized, where $\mathbf{Y}_{d:d',m:m'}$ denotes a submatrix consisting of the elements in rows $d, d+1, \ldots, d'$ and columns $m, m+1, \ldots, m'$ of $\mathbf{Y}$. Thirdly, the first column of $\mathbf{Y}$ must correspond to an initial vector with which the recursion is assumed to start. We thus assume that the first column of $\mathbf{Y}$ is always set at an all-zero vector.

The source and target encoders are free to ignore the phoneme information contained in the mel spectrogram inputs when finding a time alignment between source and target speech. One natural way to ensure that $\mathbf{K}$ and $\mathbf{Q}$ contain necessary information for finding an appropriate time alignment would be to assist $\mathbf{K}$ and $\mathbf{Q}$ to preserve sufficient information for reconstructing the mel spectrogram inputs. To this end, we introduce source and target reconstructors that aim to reconstruct the mel spectrograms of source and target speech, denoted by $\mathbf{C}_\mathrm{x}$ and $\mathbf{C}_\mathrm{y}$, from $\mathbf{K}$ and $\mathbf{Q}$:

$$\widetilde{\mathbf{C}}_\mathrm{x} = \mathsf{SrcRec}(\mathbf{K}), \tag{10}$$

$$\widetilde{\mathbf{C}}_\mathrm{y} = \mathsf{TrgRec}(\mathbf{Q}), \tag{11}$$

and include a reconstruction loss

$$L_{\mathsf{rec}} = \|\widetilde{\mathbf{C}}_\mathrm{x} - \mathbf{C}_\mathrm{x}\|_1 + \|\widetilde{\mathbf{C}}_\mathrm{y} - \mathbf{C}_\mathrm{y}\|_1, \tag{12}$$

in the training loss to be minimized. We call (12) the "context preservation loss".

PostNet aims to restore the linear spectrogram from its mel-scaled version

$$\hat{\mathbf{Z}} = \mathsf{PostNet}(\mathbf{C}_\mathrm{y}), \tag{13}$$

where $\mathbf{C}_\mathrm{y}$ denotes the mel spectrogram of the target speech. By using $\mathbf{Z}$ to denote the linear spectrogram associated with $\mathbf{C}_\mathrm{y}$, we include an L_1 loss

$$L_{\mathsf{post}} = \|\mathsf{PostNet}(\mathbf{C}_\mathrm{y}) - \mathbf{Z}\|_1 + \|\mathsf{PostNet}(\hat{\mathbf{C}}_\mathrm{y}) - \mathbf{Z}\|_1, \tag{14}$$

in the training loss to be minimized, where $\hat{\mathbf{C}}_\mathrm{y}$ denotes the mel spectrogram produced by the target decoder.

As detailed in Fig. 2, all the networks are designed using fully convolutional architectures with gated linear units (GLUs) [30]. Although we also tested highway blocks [37] for the architecture design, it transpired that GLU blocks performed better in our preliminary experiments. Since it is important to be aware of real-time requirements when building

VC systems, we used causal convolutions to design all the convolution layers in the encoders and postnet as well as those in the target decoder. The output of the GLU block used in the present model is defined as $\mathsf{GLU}(\mathbf{X}) = \mathsf{B}_1(\mathsf{L}_1(\mathbf{X})) \odot \sigma(\mathsf{B}_2(\mathsf{L}_2(\mathbf{X})))$ where $\mathbf{X}$ is the layer input, L_1 and L_2 denote dilated convolution layers, B_1 and B_2 denote batch normalization layers, and σ denotes a sigmoid gate function.

It would be natural to assume that the time alignment between parallel utterances is usually monotonic and nearly linear. This implies that the diagonal region in the attention matrix $\mathbf{A}$ should be dominant. We expect that imposing such restrictions on $\mathbf{A}$ can significantly reduce the training effort since the search space for $\mathbf{A}$ can be greatly reduced. To penalize $\mathbf{A}$ for not having a diagonally dominant structure, Tachibana et al. proposed introducing a "guided attention loss" [26]:

$$L_{\mathsf{att}} = \|\mathbf{G} \odot \mathbf{A}\|_1, \tag{15}$$

where $\odot$ denotes elementwise multiplication and $\mathbf{G} \in \mathbb{R}^{N\times M}$ is a non-negative weight matrix whose (n,m)-th element $g_{n,m}$ is defined as $g_{n,m} = 1 - e^{-(n/N-m/M)^2/2\nu^2}$.

To summarize, the total training loss for the present ConvS2S-VC model to be minimized is given as

$$L_{\mathsf{dec}} + \lambda_{\mathsf{post}} L_{\mathsf{post}} + \lambda_{\mathsf{rec}} L_{\mathsf{rec}} + \lambda_{\mathsf{att}} L_{\mathsf{att}}, \tag{16}$$

where $\lambda_{\mathsf{post}} \geq 0$, $\lambda_{\mathsf{rec}} \geq 0$ and $\lambda_{\mathsf{att}} \geq 0$ are regularization parameters, which weigh the importances of L_{post}, L_{rec} and L_{post} relative to L_{dec}.

3.3. Conversion process

At test time, we can convert a source speech feature sequence $\mathbf{X}$ via the following recursion:

$(\mathbf{K}, \mathbf{V}) = \mathsf{SrcEnc}(\mathbf{X})$, $\mathbf{Y} = \mathbf{0}$
for $m = 1$ to M' **do**
  $\mathbf{Q} = \mathsf{TrgEnc}(\mathbf{Y})$
  $\mathbf{A} = \mathsf{softmax}_n\big(\mathbf{K}^{\mathsf{T}}\mathbf{Q}/\sqrt{d}\big)$
  $\mathbf{R} = \mathbf{AV}$
  $\hat{\mathbf{Y}} = \mathsf{TrgDec}(\mathbf{R})$
  $\mathbf{Y} = [\mathbf{0}, \hat{\mathbf{Y}}]$
end for
$\mathbf{C}_{\mathrm{y}} = \hat{\mathbf{Y}}_{1:J,1:M'}$, $\hat{\mathbf{Z}} = \mathsf{PostNet}(\mathbf{C}_{\mathrm{y}})$
return $\hat{\mathbf{Z}}$, $\hat{\mathbf{Y}}$

When computing $\mathbf{A}$, we used the same heuristics employed in [26] to ensure that $\mathbf{A}$ becomes diagonally dominant.

Once $\hat{\mathbf{Z}}$ and $\hat{\mathbf{Y}}$ have been obtained, we can generate a time-domain signal using the WORLD vocoder.

4. EXPERIMENTS

To confirm the performance of our proposed method, we conducted subjective evaluation experiment involving a speaker identity conversion task. For the experiment, we used the CMU Arctic database [38], which consists of 1132 phonetically balanced English utterances spoken by four US English speakers. We selected "clb" (female) and "rms" (male) as the source speakers and "slt" (female) and "bdl" (male) as the target speakers. The audio files for each speaker were manually divided into 1000 and 132 files, which were provided as training and evaluation sets, respectively. All the speech signals were sampled at 16 kHz. For each utterance, the spectral envelope (513 dimensions), log F_0, aperiodicity, and voiced/unvoiced information were extracted every 8 ms using the WORLD analyzer [35]. The spectral envelope sequences were then converted into 80-dimensional mel-frequency-scaled spectrograms. Namely, $I = 513$, $D = 83$ and $J = 80$. Adam optimization [39] was used for model training.

We chose the open-source VC system presented in [40] for comparison with our experiments. It should be noted that this system was one of the best performing systems in the Voice Conversion Challenge (VCC) 2016 [41] and VCC 2018 [42] in terms of both sound quality and speaker similarity. We conducted an AB test to compare the sound quality of the converted speech samples and an ABX test to compare the similarity to the target speaker of the converted speech samples, where "A" and "B" were converted speech samples obtained with the proposed and baseline methods and "X" was a real speech sample obtained from a target speaker. With these listening tests, "A" and "B" were presented in random orders to eliminate bias in the order of the stimuli. Nine listeners participated in our listening tests. Each listener was presented with {"A","B"} $\times$ 20 utterances for the AB test of sound quality and {"A","B","X"} $\times$ 20 utterances for the ABX test of speaker similarity. Each listener was then asked to select "A", "B" or "fair" for each utterance. The results are shown in Fig. 3. As the results reveal, the proposed method outperformed the baseline method in terms of both sound quality and speaker similarity. Audio samples are provided at http://www.kecl.ntt.co.jp/people/kameoka.hirokazu/Demos/.

5. CONCLUSIONS

This paper proposed a VC method based on a fully convolutional seq2seq model, which we call "ConvS2S-VC".

There is a lot of future work to be done. Although we chose only one conventional method as the baseline in the present experiment, we plan to compare our method with other state-of-the-art methods. In addition, we plan to conduct more thorough evaluations in order to validate each of the choices we made as regards our model, such as the network architecture, with or without the guided attention loss, and with or without the context preservation mechanism, and report the results in forthcoming papers. As with the best performing systems [43] in VCC 2018, we are interested in incorporating the WaveNet vocoder [31, 44] into our system in place of the WORLD vocoder to realize further improvements in sound quality. Recently, we have also been developing a VC system using an LSTM-based seq2seq model [34] in parallel with this work. It would be interesting

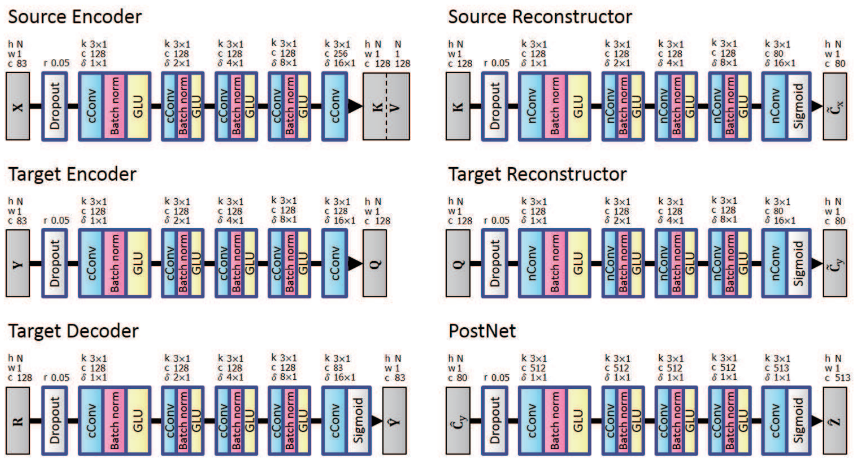

Fig. 2. Network architectures of the source and target encoders, source and target reconstructors, target decoders and postnet. Here, the input and output of each of the networks are interpreted as images, where “h”, “w” and “c” denote the height, width and channel number, respectively. “cConv”, “nConv”, “Dropout”, “Batch norm”, “GLU”, and “Sigmoid” denote causal convolution, normal convolution, dropout, batch normalization, gated linear unit, and sigmoid layers, respectively. “k”, “c”, “δ” denote the kernel size, output channel number and dilation factor of a causal/normal convolution layer, respectively. “r” denotes the dropout ratio.

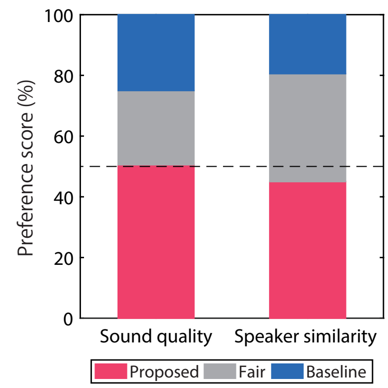

Fig. 3. Results of the AB test for sound quality and the ABX test for speaker similarity.

to investigate which of the two methods performs better in a similar setting.

Acknowledgements: This work was supported by JSPS KAKENHI 17H01763.

6. REFERENCES

[1] A. Kain and M. W. Macon, “Spectral voice conversion for text-to-speech synthesis,” in *Proc. ICASSP*, 1998, pp. 285–288.

[2] 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 Commun.*, vol. 49, no. 9, pp. 743–759, 2007.

[3] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, “Speaking-aid systems using GMM-based voice conversion for electrolaryngeal speech,” *Speech Commun.*, vol. 54, no. 1, pp. 134–146, 2012.

[4] Z. Inanoglu and S. Young, “Data-driven emotion conversion in spoken English,” *Speech Commun.*, vol. 51, no. 3, pp. 268–283, 2009.

[5] O. Türk and M. Schröder, “Evaluation of expressive speech synthesis with voice conversion and copy resynthesis techniques,” *IEEE Trans. ASLP*, vol. 18, no. 5, pp. 965–973, 2010.

[6] T. Toda, M. Nakagiri, and K. Shikano, “Statistical voice conversion techniques for body-conducted unvoiced speech enhancement,” *IEEE Trans. ASLP*, vol. 20, no. 9, pp. 2505–2517, 2012.

[7] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, “Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks,” in *Proc. Interspeech*, 2017, pp. 1283–1287.

[8] Y. Stylianou, O. Cappé, and E. Moulines, “Continuous probabilistic transform for voice conversion,” *IEEE Trans. SAP*, vol. 6, no. 2, pp. 131–142, 1998.

[9] T. Toda, A. W. Black, and K. Tokuda, “Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory,” *IEEE Trans. ASLP*, vol. 15, no. 8, pp. 2222–2235, 2007.

[10] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, “Voice conversion using partial least squares regression,” *IEEE Trans. ASLP*, vol. 18, no. 5, pp. 912–921, 2010.

[11] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, “Spectral mapping using artificial neural networks for voice conversion,” *IEEE Trans. ASLP*, vol. 18, no. 5, pp. 954–964, 2010.

[12] S. H. Mohammadi and A. Kain, “Voice conversion using deep neural networks with speaker-independent pre-training,” in *Proc. SLT*, 2014, pp. 19–23.

[13] Y. Saito, S. Takamichi, and H. Saruwatari, “Voice conversion using input-to-output highway networks,” *IEICE Trans Inf. Syst.*, vol. E100-D, no. 8, pp. 1925–1928, 2017.

[14] L. Sun, S. Kang, K. Li, and H. Meng, “Voice conversion using deep bidirectional long short-term memory based recurrent neural networks,” in *Proc. ICASSP*, 2015, pp. 4869–4873.

[15] 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 *Proc. APSIPA*, 2016.

[16] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, “Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks,” in *Proc. Interspeech*, 2017, pp. 3364–3368.

[17] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "ACVAE-VC: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder," *arXiv:1808.05092 [stat.ML]*, Aug. 2018.

[18] T. Kaneko and H. Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv:1711.11293 [stat.ML]*, Nov. 2017.

[19] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks," *arXiv:1806.02169 [cs.SD]*, June 2018.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Adv. NIPS*, 2014, pp. 3104–3112.

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

[22] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[23] S. Ö. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep voice: Real-time neural text-to-speech," in *Proc. ICML*, 2017.

[24] S. Ö. Arık, G. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Proc. NIPS*, 2017.

[25] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *Proc. ICLR*, 2017.

[26] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. ICASSP*, 2018, pp. 4784–4788.

[27] W. Ping, K. Peng, A. Gibiansky, S. Ö. Arık, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, 2018.

[28] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[29] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015.

[30] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. ICML*, 2017, pp. 933–941.

[31] A. van den 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," *arXiv:1609.03499 [cs.SD]*, Sept. 2016.

[32] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. ICML*, 2017.

[33] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," in *Proc. Interspeech*, 2017, pp. 1268–1272.

[34] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "AttS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," in *Proc. ICASSP*, 2019, submitted.

[35] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. Inf. Syst.*, vol. E99-D, no. 7, pp. 1877–1884, 2016.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. NIPS*, 2017.

[37] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv:1505.00387 [cs.LG]*, May 2015.

[38] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Proc. SSW*, 2004, pp. 223–224.

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

[40] K. Kobayashi and T. Toda, "sprocket: Open-source voice conversion software," in *Proc. Odyssey*, 2018, pp. 203–210.

[41] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The Voice Conversion Challenge 2016," in *Proc. Interspeech*, 2016, pp. 1632–1636.

[42] 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," *arXiv:1804.04262 [eess.AS]*, Apr. 2018.

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

[44] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. Interspeech*, 2017, pp. 1118–1122.