This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

ConvS2S-VC: Fully Convolutional Sequence-to-Sequence Voice Conversion

Hirokazu Kameoka, *Senior Member, IEEE,* Kou Tanaka, Damian Kwaśny, Takuhiro Kaneko, and Nobukatsu Hojo

***Abstract*—This paper proposes a voice conversion (VC) method using sequence-to-sequence (seq2seq or S2S) learning, which flexibly converts not only the voice characteristics but also the pitch contour and duration of input speech. The proposed method, called ConvS2S-VC, has three key features. First, it uses a model with a fully convolutional architecture. This is particularly advantageous in that it is suitable for parallel computations using GPUs. It is also beneficial since it enables effective normalization techniques such as batch normalization to be used for all the hidden layers in the networks. Second, it achieves many-to-many conversion by simultaneously learning mappings among multiple speakers using only a single model instead of separately learning mappings between each speaker pair using a different model. This enables the model to fully utilize available training data collected from multiple speakers by capturing common latent features that can be shared across different speakers. Owing to this structure, our model works reasonably well even without source speaker information, thus making it able to handle any-to-many conversion tasks. Third, we introduce a mechanism, called the conditional batch normalization that switches batch normalization layers in accordance with the target speaker. This particular mechanism has been found to be extremely effective for our many-to-many conversion model. We conducted speaker identity conversion experiments and found that ConvS2S-VC obtained higher sound quality and speaker similarity than baseline methods. We also found from audio examples that it could perform well in various tasks including emotional expression conversion, electrolaryngeal speech enhancement, and English accent conversion.**

***Index Terms*—Voice conversion (VC), sequence-to-sequence learning, attention, fully convolutional model, many-to-many VC.**

I. 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 [2], speaking aids [3], [4], speech enhancement [5]–[7], and accent conversion [8].

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) [9]–[11] and deep neural networks (DNNs) [12]–[16]. 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 used with notable success for non-parallel VC tasks [17]–[21].

One limitation of conventional methods including those mentioned above is that they are focused mainly on learning to convert only the local spectral features and less on converting prosodic features such as the fundamental frequency (F_0) contour, duration, and rhythm of the input speech. This is because the acoustic models in these methods are designed to describe mappings between local features only. This prevents a model from discovering word-level or sentence-level suprasegmental conversion rules. In 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 local 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, we need a model that can learn to convert entire feature sequences by capturing and utilizing long-term dependencies in source and target speech. To this end, we adopt a sequence-to-sequence (seq2seq or S2S) learning approach.

The S2S learning approach offers a general and powerful framework for transforming one sequence into another variable length sequence [22], [23]. 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 in accordance with the internal representation. The original S2S model employs recurrent neural networks (RNNs) to model the encoder and decoder networks, where common 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, automatic speech recognition (ASR) [23] and text-to-speech (TTS) [24]–[30].

The original S2S model suffers from the constraint that all input sequences are forced to be encoded into a fixed length

H. Kameoka, K. Tanaka, Damian Kwaśny, T. Kaneko and N. Hojo are with NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation, Atsugi, Kanagawa, 243-0198 Japan (e-mail: kameoka.hirokazu@lab.ntt.co.jp).

Manuscript received ***; revised ***. This work was supported by JSPS KAKENHI 17H01763 and JST CREST Grant Number JPMJCR19A3, Japan. A preliminary version of this work has already been made publicly available [1].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

internal vector. This limits 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" [31] has been introduced, which enables the network to learn where to pay attention in the input sequence for each item in the output sequence.

While RNNs are a natural choice for modeling long sequential data, recent work has shown that convolutional neural networks (CNNs) with gating mechanisms also have excellent potential for capturing long-term dependencies [32], [33]. In addition, they are suitable for parallel computations using GPUs unlike RNNs. To exploit this advantage of CNNs, an S2S model was recently proposed that adopts a fully convolutional architecture [34]. With this model, the decoder is designed using causal convolutions so that it enables the model to generate an output sequence autoregressively. This model with an attention mechanism is called the ConvS2S model and has already been applied successfully to machine translation [34] and TTS [28], [29]. Inspired by its success in these tasks, 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 a wide sense, VC is a task of converting the domain of speech. Here, the types of domain include speaker identities, emotional expressions, speaking styles, and accents, but for concreteness, we will restrict our attention to speaker identity conversion tasks in the following. When we are interested in converting speech among multiple speakers, one naive way of applying the S2S model is to prepare and train a model for each speaker pair. However, this can be inefficient since the model for one pair of speakers fails to use the training data of the other speakers for training, even though there must be a common set of latent features that can be shared across different speakers, especially when the languages are the same. To fully utilize available training data collected from multiple speakers, we further propose an extension of the ConvS2S model that allows for "many-to-many" VC, which can learn mappings among multiple speakers using only a single model.

One important advantage of using fully convolutional networks is that it enables the use of batch normalization in all the hidden layers. This is practically beneficial since batch normalization is known to be significantly effective in not only accelerating training but also improving the generalization ability of the resulting models. Indeed, as described later, it also positively affected our pairwise model. However, as for the many-to-many model, the distributions of the layer inputs can change depending on the source and target speakers, which may affect model training. To stabilize layer input distributions, we introduce a mechanism, called the conditional batch normalization, that switches batch normalization layers in accordance with the source and target speakers. This particular mechanism was experimentally found to work very well.

II. Related work

Note that some attempts have recently been made to apply S2S models to VC problems, including the ones we proposed previously [1], [35]. Although most S2S models typically require sufficiently large parallel corpora for training, collecting a sufficient number of parallel utterances is not always feasible. Thus, particularly in VC tasks, one challenge is how best to train S2S models using a limited amount of training data.

One idea involves using text labels as auxiliary information for model training, assuming they are readily available. For example, Miyoshi et al. proposed combining acoustic models for ASR and TTS with an S2S model [36], where an S2S model is used to convert the context posterior probability sequence produced by the ASR model and the TTS model is finally used to generate a target speech feature sequence. Zhang et al. also proposed an S2S model-based VC method guided by an ASR system, which augments inputs with bottleneck features obtained from a pretrained ASR system [37]. Subsequently, Zhang et al. proposed a shared model for TTS and VC tasks, which enables joint training of the TTS and VC functions [38]. Recently, Biadsy et al. proposed an end-to-end VC system called Parrotron, which is designed to train the encoder and decoder along with an ASR model on the basis of a multitask learning strategy [39]. Our method differs from these methods in that our model does not rely on ASR or TTS models and requires no text annotations for model training. Instead, we introduce several techniques to stabilize training and test prediction.

Haque et al. proposed a method that enables many-to-many VC similar to ours [40]. As detailed in Subsection IV-C, our many-to-many model differs in that it does not necessarily require source speaker information for the encoder, thus enabling it to also handle *any*-to-many VC tasks.

In addition, our method differs from all the methods mentioned above in that it adopts a fully convolutional model, which can be potentially advantageous in several ways, as already mentioned.

III. ConvS2S-VC

In this section, we start by describing a pairwise one-to-one conversion model and then present its multi-speaker extension that enables many-to-many VC. The overall architecture of the pairwise conversion model is illustrated in Fig. 1.

A. Feature extraction and normalization

First, we define acoustic features to be converted. Although one interesting option would be to consider directly converting time-domain signals, given the recent significant advances in high-quality neural vocoder systems [33], [41]–[50], we find it reasonable to consider converting acoustic features such as the mel-cepstral coefficients (MCCs) [51] and log F_0, since we would expect to generate high-fidelity signals by using a neural vocoder if we could obtain a sufficient set of acoustic features. In such systems, the model size for the convertor can be made small enough to enable the system to work well even when a limited amount of training data is available. Hence, in this paper we choose to use the MCCs, log F_0, aperiodicity, and voiced/unvoiced indicator of speech as acoustic features as detailed below.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

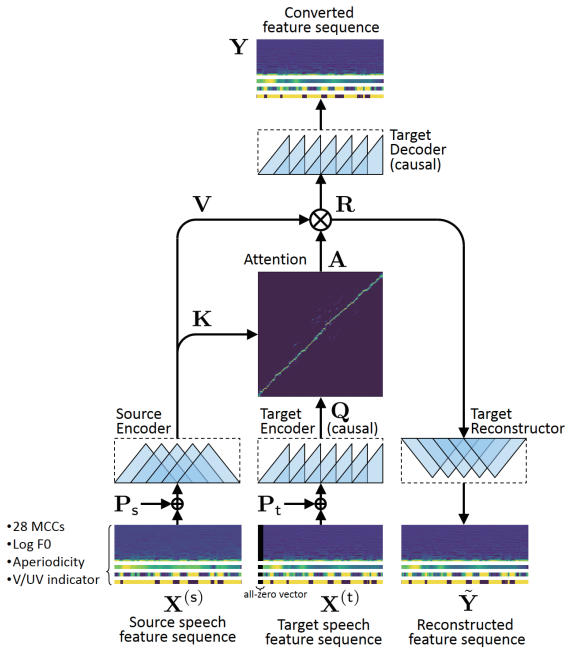

Fig. 1. Overall structure of the pairwise ConvS2S model.

We first use the WORLD analyzer [52] to extract the spectral envelope, the log F_0, the coded aperiodicity, and the voiced/unvoiced indicator within each time frame of a speech utterance, then compute I MCCs from the extracted spectral envelope, and finally construct an acoustic feature vector by stacking the MCCs, the log F_0, the coded aperiodicity, and the voiced/unvoiced indicator. Thus, each acoustic feature vector consists of $I+3$ elements. Here, the log F_0 contour is assumed to be filled with smoothly interpolated values in unvoiced segments. At training time, we normalize each element $x_{i,n}$ $(i = 1, \ldots, I)$ of the MCCs and the log F_0 $x_{I+1,n}$ at frame n to $x_{i,n} \leftarrow (x_{i,n} - \mu_i)/\sigma_i$ where i, μ_i and σ_i denote the feature index, the mean, and the standard deviation of the i-th feature within all the voiced segments of the training samples of the same speaker.

To accelerate and stabilize training and inference, we have found it useful to use a similar trick introduced by Wang et al. [53]. Specifically, we divide the acoustic feature sequence obtained above into non-overlapping segments of equal length r and use the stack of the acoustic feature vectors in each segment as a new feature vector so that the new feature sequence becomes r times shorter than the original feature sequence. Furthermore, we add the sinusoidal position encodings [54] to the reshaped version of the feature sequence before feeding it into the model.

B. Model

We hereafter use $\mathbf{X}^{(\mathsf{s})} = [\mathbf{x}_1^{(\mathsf{s})}, \ldots, \mathbf{x}_{N_\mathsf{s}}^{(\mathsf{s})}] \in \mathbb{R}^{D \times N_\mathsf{s}}$ and $\mathbf{X}^{(\mathsf{t})} = [\mathbf{x}_1^{(\mathsf{t})}, \ldots, \mathbf{x}_{N_\mathsf{t}}^{(\mathsf{t})}] \in \mathbb{R}^{D \times N_\mathsf{t}}$ to denote the source and target speech feature sequences of non-aligned parallel utterances, where N_s and N_t denote the lengths of the two sequences and D denotes the feature dimension. We consider an S2S model that aims to map $\mathbf{X}^{(\mathsf{s})}$ to $\mathbf{X}^{(\mathsf{t})}$. Our pairwise conversion model is inspired by and built upon the models presented by Vaswani et al. [54] and Tachibana et al. [28], with the difference being that it involves an additional network, called a target reconstructor. This network plays an important role in ensuring that the encoders preserve contextual information about the source and target speech, as explained below. Our model thus consists of four networks: source and target encoders, a target decoder, and a target reconstructor.

As with many S2S models, our model has an encoder-decoder structure (Fig. 1). The source and target encoders are expected to extract contextual information from source and target speech. Given the contextual vector sequence pair produced by the encoders, we can compute a contextual similarity matrix between the source and target speech, which can be used to warp the time-axis of the source speech. We can then generate the feature sequence of the target speech by letting the target decoder transform each element of the time-warped version of the contextual vector sequence of the source speech. This idea can be formulated as follows.

The source encoder takes $\mathbf{X}^{(\mathsf{s})}$ as the input and produces two internal vector sequences $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{D' \times N_\mathsf{s}}$ and the target encoder takes $\mathbf{X}^{(\mathsf{t})}$ as the input and produces an internal vector sequence $\mathbf{Q} \in \mathbb{R}^{D' \times N_\mathsf{t}}$:

$$[\mathbf{K}; \mathbf{V}] = \mathsf{SrcEnc}(\mathbf{X}^{(\mathsf{s})}), \tag{1}$$

$$\mathbf{Q} = \mathsf{TrgEnc}(\mathbf{X}^{(\mathsf{t})}), \tag{2}$$

where $[;]$ denotes vertical concatenation of matrices (or vectors) with compatible sizes and D' denotes the dimension of the internal vectors. $\mathbf{K}$, $\mathbf{V}$ and $\mathbf{Q}$ can be metaphorically interpreted as the queries and the key-value pairs in a hash table. By using the query and key pair, we can define an attention matrix $\mathbf{A} \in \mathbb{R}^{N_\mathsf{s} \times N_\mathsf{t}}$ as

$$\mathbf{A} = \mathsf{softmax}\big(\mathbf{K}^\mathsf{T}\mathbf{Q}/\sqrt{D'}\big), \tag{3}$$

where $\mathsf{softmax}$ denotes a softmax operation performed on the first axis. $\mathbf{A}$ can be seen as a similarity matrix, where the (n, m)-th element indicates the similarity between the n-th and m-th frames of source and target speech. The peak trajectory of $\mathbf{A}$ can therefore 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 the value vector sequence $\mathbf{V}$ is thus given as

$$\mathbf{R} = \mathbf{V}\mathbf{A}, \tag{4}$$

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

$$\mathbf{Y} = \mathsf{TrgDec}(\mathbf{R}). \tag{5}$$

Since the target speech feature sequence $\mathbf{X}^{(\mathsf{t})}$ is of course not accessible at test time, we 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 recursively. To enable the model to behave in this way, first, we must ensure 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

convolution can be easily implemented by padding the input by $\delta(\kappa-1)$ elements on both the left and right sides with zero vectors and removing $\delta(\kappa-1)$ elements from the end of the convolution output, where κ is the kernel size and δ is the dilation factor. Second, the output sequence $\mathbf{Y}$ must correspond to a time-shifted version of $\mathbf{X}^{(\mathsf{t})}$ 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

$$\mathcal{L}_{\mathsf{dec}} = \tfrac{1}{N_{\mathsf{t}}}\|\mathbf{Y}_{:,1:N_{\mathsf{t}}-1} - \mathbf{X}^{(\mathsf{t})}_{:,2:N_{\mathsf{t}}}\|_1, \tag{6}$$

in the training loss to be minimized, where we have used the colon operator : to specify the range of indices of the elements in a matrix or a vector we wish to extract. For ease of notation, we use : itself to represent all elements along an axis. For example, $\mathbf{X}^{(\mathsf{t})}_{:,2:N_{\mathsf{t}}}$ denotes a submatrix consisting of the elements in all the rows and columns $2,\ldots,N_{\mathsf{t}}$ of $\mathbf{X}^{(\mathsf{t})}$. Third, the first column of $\mathbf{X}^{(\mathsf{t})}$ must correspond to an initial vector with which the recursion is assumed to start. We thus assume that it is always set at an all-zero vector.

The source and target encoders are free to ignore the information contained in the feature vector inputs when finding a time alignment between source and target speech. One natural way to ensure that $\mathbf{K}$, $\mathbf{V}$, and $\mathbf{Q}$ contain necessary information for finding an appropriate time alignment is to assist $\mathbf{K}$, $\mathbf{V}$, and $\mathbf{Q}$ to preserve sufficient information for reconstructing the input feature sequence. To this end, we introduce a target reconstructor that aims to reconstruct the feature sequence of target speech $\mathbf{X}^{(\mathsf{t})}$ from $\mathbf{K}$, $\mathbf{V}$, and $\mathbf{Q}$:

$$\widetilde{\mathbf{Y}} = \mathsf{TrgRec}(\mathbf{R}), \tag{7}$$

and include a reconstruction loss

$$\mathcal{L}_{\mathsf{rec}} = \tfrac{1}{M}\|\widetilde{\mathbf{Y}} - \mathbf{X}^{(\mathsf{t})}\|_1, \tag{8}$$

in the training loss to be minimized. This idea was introduced in our previous work [35]. We call Eq. (8) the context preservation loss. Although the reconstructor and the decoder may appear to have similar roles, the difference is that the reconstructor is only responsible for making each column of $\mathbf{R}$ contain sufficient information about the current value of the target feature sequence so that the decoder can concentrate on predicting the future value using that information.

As detailed in Subsection V-B, all the networks are designed using fully convolutional architectures using gated linear units (GLUs) [32] with residual connections. The output of the GLU block used in the present model is defined as $\mathsf{GLU}(\mathbf{X}) = \mathsf{B}_1(\mathsf{L}_1(\mathbf{X}))\odot\mathsf{sigmoid}(\mathsf{B}_2(\mathsf{L}_2(\mathbf{X})))$ where $\mathbf{X}$ is the layer input, L_1 and L_2 are dilated convolution layers, B_1 and B_2 are batch normalization layers, and sigmoid is a sigmoid gate function. Similar to LSTMs, GLUs can reduce the vanishing gradient problem for deep architectures by providing a linear path for the gradients while retaining non-linear capabilities.

C. Constraints on Attention Matrix

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 always 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, we introduce a diagonal attention loss (DAL) [28]:

$$\mathcal{L}_{\mathsf{dal}} = \tfrac{1}{N_{\mathsf{s}}N_{\mathsf{t}}}\|\mathbf{W}_{N_{\mathsf{s}}\times N_{\mathsf{t}}}(\nu)\odot\mathbf{A}\|_1, \tag{9}$$

where $\odot$ is the elementwise product and $\mathbf{W}_{N_{\mathsf{s}}\times N_{\mathsf{t}}}(\nu)\in\mathbb{R}^{N_{\mathsf{s}}\times N_{\mathsf{t}}}$ is a non-negative weight matrix whose (n,m)-th element $w_{n,m}$ is defined as $w_{n,m} = 1 - e^{-(n/N_{\mathsf{s}}-m/N_{\mathsf{t}})^2/2\nu^2}$. Fig. 2 shows plots of $\mathbf{W}_{N_{\mathsf{s}}\times N_{\mathsf{t}}}(\nu)$.

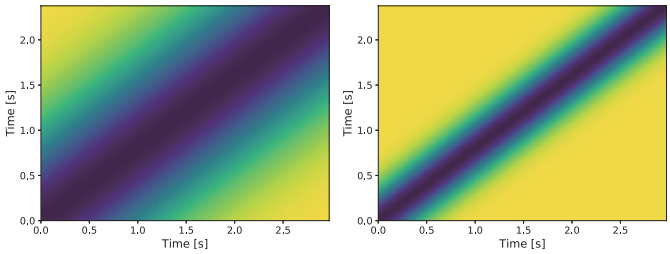

Fig. 2. Plots of $\mathbf{W}_{N_{\mathsf{s}}\times N_{\mathsf{t}}}(0.3)$ (left) and $\mathbf{W}_{N_{\mathsf{s}}\times N_{\mathsf{t}}}(0.1)$ (right) where the lengths of the source and target speech are 2.4[s] and 3.0[s], respectively.

Each time point of the target feature sequence must correspond to only one or at most a few time points of the source feature sequence. This implies that two different columns in $\mathbf{A}$ must be as orthogonal as possible. Although the DAL with a sufficiently small ν value can induce orthogonality, it may also lead to undesirable situations where the time alignment between the two sequences is forced to be always strictly linear. Thus, ν must not be set to a value too small to enable reasonably flexible time alignments. To achieve orthogonality while enabling ν to be a moderately greater value, we propose introducing another loss to constrain $\mathbf{A}$, which we call the orthogonal attention loss (OAL):

$$\mathcal{L}_{\mathsf{oal}} = \tfrac{1}{N_{\mathsf{s}}^2}\|\mathbf{W}_{N_{\mathsf{s}}\times N_{\mathsf{s}}}(\rho)\odot(\mathbf{A}\mathbf{A}^{\mathsf{T}})\|_1. \tag{10}$$

D. Training loss

Given examples of parallel utterances, the total training loss for the ConvS2S-VC model to be minimized is given as

$$\mathcal{L} = \mathbb{E}_{\mathbf{X}^{(\mathsf{s})},\mathbf{X}^{(\mathsf{t})}}\left\{\mathcal{L}_{\mathsf{dec}} + \lambda_{\mathsf{r}}\mathcal{L}_{\mathsf{rec}} + \lambda_{\mathsf{d}}\mathcal{L}_{\mathsf{dal}} + \lambda_{\mathsf{o}}\mathcal{L}_{\mathsf{oal}}\right\}, \tag{11}$$

where $\mathbb{E}_{\mathbf{X}^{(\mathsf{s})},\mathbf{X}^{(\mathsf{t})}}\{\cdot\}$ is the sample mean over all the training examples and $\lambda_{\mathsf{r}}\geq 0$, $\lambda_{\mathsf{d}}\geq 0$ and $\lambda_{\mathsf{o}}\geq 0$ are regularization parameters, which weigh the importances of $\mathcal{L}_{\mathsf{rec}}$, $\mathcal{L}_{\mathsf{dal}}$ and $\mathcal{L}_{\mathsf{oal}}$ relative to $\mathcal{L}_{\mathsf{dec}}$.

E. Conversion process

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

$[\mathbf{K};\mathbf{V}] = \mathsf{SrcEnc}(\mathbf{X})$, $\mathbf{Y}\leftarrow\mathbf{0}$
for $m = 1$ to M' **do**
  $\mathbf{Q} = \mathsf{TrgEnc}(\mathbf{Y})$
  $\mathbf{A} = \mathsf{softmax}\big(\mathbf{K}^{\mathsf{T}}\mathbf{Q}/\sqrt{D'}\big)$
  $\mathbf{R} = \mathbf{V}\mathbf{A}$
  $\mathbf{Y} = \mathsf{TrgDec}(\mathbf{R})$
  $\mathbf{Y}\leftarrow[\mathbf{0},\mathbf{Y}]$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

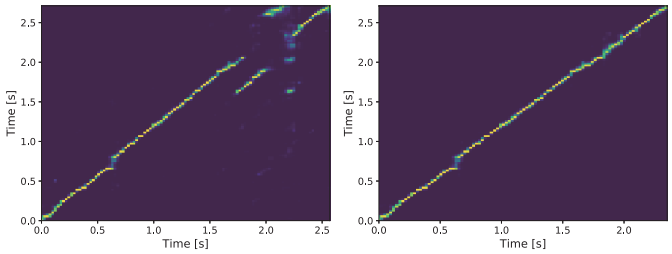

Fig. 3. Attention matrices predicted without (left) and with (right) forward attention.

end for
return $\mathbf{Y}$

However, as Fig. 3 shows, it transpired that with this algorithm the attended time point does not always move forward monotonically and continuously at test time and can occasionally become stuck at the same time point or suddenly jump to a distant time point even though the diagonal and orthogonal losses are considered in training. To assist the attended point to move forward monotonically and continuously, we limit the paths through which the attended point is allowed to move by forcing the attentions to the time points distant from the peak of the attention distribution obtained at the previous time step to zeros. This can be implemented for instance as follows:

$[\mathbf{K}; \mathbf{V}] = \mathsf{SrcEnc}(\mathbf{X})$, $\mathbf{Y} \leftarrow \mathbf{0}$
for $m = 1$ to M **do**
  $\mathbf{Q} = \mathsf{TrgEnc}(\mathbf{Y})$
  $\mathbf{A} = \mathsf{softmax}\big(\mathbf{K}^{\mathsf{T}}\mathbf{Q}/\sqrt{D'}\big)$
  if $m > 1$ **then**
    $\mathbf{a} = \mathbf{A}_{1:N,m}$
    $\mathbf{a}_{1:\max(1,\hat{n}-N_0)} = 0$, $\mathbf{a}_{\min(\hat{n}+N_1,N):N} = 0$
    $\mathbf{a} \leftarrow \mathbf{a}/\mathrm{sum}(\mathbf{a})$
    $\mathbf{A} \leftarrow [\mathbf{A}_{1:N,1:m-1}, \mathbf{a}]$
  end if
  $\hat{n} = \mathrm{argmax}_n\, \mathbf{A}_{n,m}$
  $\mathbf{R} = \mathbf{V}\mathbf{A}$
  $\mathbf{Y} = \mathsf{TrgDec}(\mathbf{R})$
  $\mathbf{Y} \leftarrow [\mathbf{0}, \mathbf{Y}]$
end for
return $\mathbf{Y}$

where $\mathrm{sum}(\cdot)$ denotes the sum of all the elements in a vector. Note that we set N_0 and N_1 at the nearest integers that correspond to 160[ms] and 320[ms], respectively. Fig. 3 shows an example of how attention matrices look different when this procedure has been undertaken. After we obtain $\mathbf{R}$ with the above algorithm, we can use the target reconstructor to compute $\widetilde{\mathbf{Y}} = \mathsf{TrgRec}(\mathbf{R})$ and use it instead of $\mathbf{Y}$ as the feature sequence of the converted speech.

Once $\mathbf{Y}$ or $\widetilde{\mathbf{Y}}$ has been obtained, we adjust the mean and variance of the generated feature sequence so that they match the pretrained mean and variance of the feature vectors of the target speaker. We can then generate a time-domain signal using the WORLD vocoder or any recently developed neural vocoder [33], [41]–[50]. Note that in the following experiments, we chose to use $\widetilde{\mathbf{Y}}$ for final waveform generation as it resulted in better-sounding speech.

F. Real-Time System Design

Real-time requirements must be considered when building VC systems. If we want our model to work in real-time, first, we must not allow the source encoder to use future information as with the target encoder and decoder during training. This requirement can easily be implemented by constraining the convolution layers in the source encoder (and the target reconstructor, if we assume it is used to generate the converted feature sequence) to be causal. Another point we must consider is that the speaking rate and rhythm of input speech cannot be changed drastically at test time. One simple way of keeping them unchanged is to set $\mathbf{A}$ to an identity matrix. In this way, the autoregressive recursion will be no longer needed and the conversion can be performed in a sliding-window fashion as:

$[\mathbf{K}; \mathbf{V}] = \mathsf{SrcEnc}(\mathbf{X})$
$\mathbf{Y} = \mathsf{TrgDec}(\mathbf{V})$ or $\mathbf{Y} = \mathsf{TrgRec}(\mathbf{V})$
return $\mathbf{Y}$

We will show later how these modifications can affect the VC performance. Note that even under this setting, the ability to learn and apply conversion rules that capture long-term dependencies is still effective.

G. Impact of Batch Normalization

As mentioned earlier, using fully convolutional architectures allows the use of batch normalization for all the hidden layers in the networks, which is not straightforward for architectures including recurrent modules. One benefit of using batch normalization layers is that it enables the networks to use a higher learning rate without vanishing or exploding gradients. It is also believed to help regularize the networks such that it is easier to generalize and mitigate overfitting. The effect of batch normalization will be verified experimentally in Section V.

IV. Many-to-Many ConvS2S-VC

A. Model and Training Loss

We now describe an extension of the ConvS2S model that enables many-to-many VC. Here, the idea is to use a single model to achieve mappings among multiple speakers. The model consists of the same set of the networks as the pairwise model. The only difference is that each network takes a speaker index as an additional input.

Let $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(K)}$ be examples of the acoustic feature sequences of speech in different speakers reading the same sentence. Given a single pair of parallel utterances $\mathbf{X}^{(k)}$ and $\mathbf{X}^{(k')}$, where k and k' denote the source and target speaker indices (integers), the source encoder takes $\mathbf{X}^{(k)}$ and the source speaker index k as the inputs and produces two internal vector sequences $\mathbf{K}^{(k)}, \mathbf{V}^{(k)}$, whereas the target encoder takes $\mathbf{X}^{(k')}$ and the target speaker index k' as the inputs and produces an internal vector sequence $\mathbf{Q}^{(k')}$:

$$[\mathbf{K}^{(k)}; \mathbf{V}^{(k)}] = \mathsf{SrcEnc}(\mathbf{X}^{(k)}, k), \tag{12}$$

$$\mathbf{Q}^{(k')} = \mathsf{TrgEnc}(\mathbf{X}^{(k')}, k'). \tag{13}$$

The attention matrix $\mathbf{A}^{(k,k')}$ and the time-warped version of $\mathbf{V}^{(k)}$ are then computed using $\mathbf{K}^{(k)}$ and $\mathbf{Q}^{(k')}$:

$$\mathbf{A}^{(k,k')} = \mathsf{softmax}_n\big(\mathbf{K}^{(k)\mathsf{T}}\mathbf{Q}^{(k')}/\sqrt{D'}\big), \tag{14}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

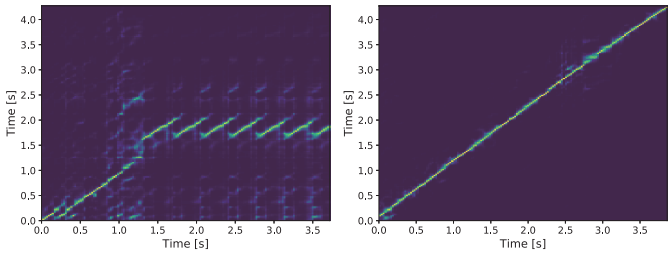

Fig. 4. Examples of the attention matrices predicted from test input female speech using the many-to-many model: with batch normalization (left) and with conditional batch normalization (right).

$$\mathbf{R}^{(k,k')} = \mathbf{V}^{(k)}\mathbf{A}^{(k,k')}. \tag{15}$$

The outputs of the reconstructor and decoder given the input $\mathbf{R}^{(k,k')}$ with target speaker conditioning are finally given as

$$\widetilde{\mathbf{Y}}^{(k,k')} = \mathsf{TrgRec}(\mathbf{R}^{(k,k')}, k'), \tag{16}$$

$$\mathbf{Y}^{(k,k')} = \mathsf{TrgDec}(\mathbf{R}^{(k,k')}, k'). \tag{17}$$

The loss functions to be minimized given this single training example are given as

$$\mathcal{L}_{\mathsf{dec}}^{(k,k')} = \tfrac{1}{N_{k'}}\|\mathbf{Y}_{:,1:N_{k'}-1}^{(k,k')} - \mathbf{X}_{:,2:N_{k'}}^{(k')}\|_1, \tag{18}$$

$$\mathcal{L}_{\mathsf{dal}}^{(k,k')} = \tfrac{1}{N_k N_{k'}}\|\mathbf{W}_{N_k\times N_{k'}}(\nu) \odot \mathbf{A}^{(k,k')}\|_1, \tag{19}$$

$$\mathcal{L}_{\mathsf{oal}}^{(k,k')} = \tfrac{1}{N_k{}^2}\|\mathbf{W}_{N_k\times N_k}(\rho) \odot (\mathbf{A}^{(k,k')}\mathbf{A}^{(k,k')\mathsf{T}})\|_1, \tag{20}$$

$$\mathcal{L}_{\mathsf{rec}}^{(k,k')} = \tfrac{1}{N_{k'}}\|\widetilde{\mathbf{Y}}^{(k,k')} - \mathbf{X}^{(k')}\|_1. \tag{21}$$

With the above model, the case where $k = k'$ would be reasonable to also consider. Minimizing this loss corresponds to ensuring that the input feature sequence $\mathbf{X}^{(k)}$ will remain unchanged when the source and target speakers are the same. We call this loss the "identity mapping loss (IML)". The effect given by this loss will be shown later. Hence, the total training loss to be minimized becomes

$$\mathcal{L} = \sum_{k,k'\neq k} \mathbb{E}_{\mathbf{X}^{(k)},\mathbf{X}^{(k')}}\{\mathcal{L}_{\mathsf{all}}^{(k,k')}\} + \lambda_{\mathsf{i}}\sum_{k}\mathbb{E}_{\mathbf{X}^{(k)}}\{\mathcal{L}_{\mathsf{all}}^{(k,k)}\},$$

$$\mathcal{L}_{\mathsf{all}}^{(k,k')} = \mathcal{L}_{\mathsf{dec}}^{(k,k')} + \lambda_{\mathsf{r}}\mathcal{L}_{\mathsf{rec}}^{(k,k')} + \lambda_{\mathsf{d}}\mathcal{L}_{\mathsf{dal}}^{(k,k')} + \lambda_{\mathsf{o}}\mathcal{L}_{\mathsf{oal}}^{(k,k')}, \tag{22}$$

where $\mathbb{E}_{\mathbf{X}^{(k)},\mathbf{X}^{(k')}}[\cdot]$ and $\mathbb{E}_{\mathbf{X}^{(k)}}[\cdot]$ denote the sample means over all the training examples of parallel utterances in speakers k and k', and $\lambda_{\mathsf{i}} \geq 0$ is a regularization parameter, which weighs the importance of the IML.

B. Conditional Batch Normalization

The left figure in Fig. 4 shows the attention matrix predicted from input female speech using the many-to-many model with regular batch normalization layers. As this example shows, attention matrices predicted by the many-to-many model tended to become blurry, mostly resulting in unintelligible speech. We conjecture that this was caused by the fact that the distributions of the inputs to the hidden layers can change in accordance with the source and/or target speakers. To normalize layer input distributions on a speaker-dependent basis, we propose using conditional batch normalization layers for the many-to-many model. Each element $y_{b,d,n}$ of the output of a regular batch normalization layer $\mathbf{Y} = \mathsf{B}(\mathbf{X})$ is defined as $y_{b,d,n} = \gamma_d \frac{x_{b,d,n}-\mu_d(\mathbf{X})}{\sigma_d(\mathbf{X})} + \beta_d$, where $\mathbf{X}$ denotes the layer input given by a three-way array with batch, channel, and time axes, $x_{b,d,n}$ denotes its (b,d,n)-th element, $\mu_d(\mathbf{X})$ and $\sigma_d(\mathbf{X})$ denote the mean and standard deviation of the d-th channel components of $\mathbf{X}$ computed along the batch and time axes, and $\boldsymbol{\gamma} = [\gamma_1,\ldots,\gamma_D]$ and $\boldsymbol{\beta} = [\beta_1,\ldots,\beta_D]$ denote the parameters to be learned. In contrast, the output of a conditional batch normalization layer $\mathbf{Y} = \mathsf{B}^k(\mathbf{X})$ is defined as $y_{b,d,n} = \gamma_d^k \frac{x_{b,d,n}-\mu_d(\mathbf{X})}{\sigma_d(\mathbf{X})} + \beta_d^k$, where the only difference is that the parameters $\boldsymbol{\gamma}^k = [\gamma_1^k,\ldots,\gamma_D^k]$ and $\boldsymbol{\beta}^k = [\beta_1^k,\ldots,\beta_D^k]$ are conditioned on speaker k. Note that a similar idea, called the conditional instance normalization, has been introduced to modify the instance normalization process for image style transfer [55] and non-parallel VC [56].

C. Any-to-many Conversion

With the models presented above, the source speaker must be known and specified during both training and inference. However, there can be certain situations where the source speaker is unknown or arbitrary. We call VC tasks in such scenarios any-to-one or any-to-many VC. Our many-to-many model can be modified to handle any-to-many VC tasks by not allowing the source encoder to take the source speaker index k as an input at both training and test time. The modified version can be formulated by simply replacing Eq. (12) in the many-to-many model with

$$[\mathbf{K}^{(k)}; \mathbf{V}^{(k)}] = \mathsf{SrcEnc}(\mathbf{X}^{(k)}). \tag{23}$$

V. Experiments

A. Experimental Settings

To evaluate the effects of the ideas presented in Sections III and IV, we conducted objective and subjective evaluation experiments involving a speaker identity conversion task. For the experiment, we used the CMU Arctic database [57], which consists of recordings of 1132 phonetically balanced English utterances spoken by four US English speakers. We used all the speakers (clb (female), bdl (male), slt (female), and rms (male)) for training and evaluation. Thus, in total there were 12 different combinations of source and 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. As already detailed in Subsection III-A, for each utterance, the spectral envelope, log F_0, coded aperiodicity, and voiced/unvoiced information were extracted every 8 ms using the WORLD analyzer [52]. Then, 28 MCCs were extracted from each spectral envelope using the Speech Processing Toolkit (SPTK) [58]. The reduction factor r was set to 3. Hence, the dimension of the acoustic feature was $D = (28+3)\times 3 = 93$.

B. Network architectures

We use the notations in Tab. I to describe the network architectures. The architectures of all the networks in the pairwise and many-to-many models are detailed in Tab. II.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE I
NOTATIONS FOR NETWORK ARCHITECTURE DESCRIPTIONS

Notation	Meaning
$\mathbf{X}$ (bold symbol)	two-way array with channel and time axes
$(f \circ g)(\mathbf{X})$	function composition $f(g(\mathbf{X}))$
$(\bigcirc_{n=0}^{N} f_n)(\mathbf{X})$	multiple compositions $(f_N \circ \cdots \circ f_1 \circ f_0)(\mathbf{X})$
$\mathsf{RL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})$	1D regular (non-causal) convolution (κ: kernel size, δ: dilation factor, i: input channel size, o: output channel size)
$\mathsf{CL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})$	1D causal convolution (κ: kernel size, δ: dilation factor, i: input channel size, o: output channel size)
$\mathsf{B}(\mathbf{X})$	normalization (BN or IN)
$\mathsf{B}^k(\mathbf{X})$	conditional normalization (CBN or CIN) conditioned on speaker k
$\mathsf{ResRGLU}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})$	$\mathsf{B}(\mathsf{RL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})) \odot \mathsf{sigmoid}(\mathsf{B}(\mathsf{RL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X}))) + \mathbf{X}_{1:o,:}$
$\mathsf{ResCGLU}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})$	$\mathsf{B}(\mathsf{CL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})) \odot \mathsf{sigmoid}(\mathsf{B}(\mathsf{CL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X}))) + \mathbf{X}_{1:o,:}$
$\mathsf{ResRGLU}^{k\,o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})$	$\mathsf{B}^k(\mathsf{RL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})) \odot \mathsf{sigmoid}(\mathsf{B}^k(\mathsf{RL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X}))) + \mathbf{X}_{1:o,:}$
$\mathsf{ResCGLU}^{k\,o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})$	$\mathsf{B}^k(\mathsf{CL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X})) \odot \mathsf{sigmoid}(\mathsf{B}^k(\mathsf{CL}^{o\leftarrow i}_{\kappa\star\delta}(\mathbf{X}))) + \mathbf{X}_{1:o,:}$
$\mathsf{drop}(\mathbf{X})$	dropout with ratio 0.1
$\mathsf{embed}^o(k)$	retrieving an o-dimensional embedding vector from an integer k
$\mathsf{A}_{\mathbf{Z}}(\mathbf{X})$	appending a broadcast version of $\mathbf{Z}$ (expanded along the time axis) to $\mathbf{X}$ along the channel dimension

TABLE II
NETWORK ARCHITECTURES OF PAIRWISE AND MANY-TO-MANY MODELS

Model	Network	Architecture
pairwise	$\mathsf{SrcEnc}(\mathbf{X})$	$(\mathsf{RL}^{512\leftarrow 256}_{1\star 1} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}\mathsf{ResRGLU}^{256\leftarrow 256}_{5\star 3^l})) \circ \mathsf{B} \circ \mathsf{RL}^{256\leftarrow 93}_{1\star 1} \circ \mathsf{drop})(\mathbf{X})$
	$\mathsf{TrgEnc}(\mathbf{Y})$	$(\mathsf{CL}^{256\leftarrow 256}_{1\star 1} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}\mathsf{ResCGLU}^{256\leftarrow 256}_{3\star 3^l})) \circ \mathsf{B} \circ \mathsf{CL}^{256\leftarrow 93}_{1\star 1} \circ \mathsf{drop})(\mathbf{Y})$
	$\mathsf{TrgRec}(\mathbf{R})$	$(\mathsf{RL}^{93\leftarrow 256}_{1\star 1} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}\mathsf{ResRGLU}^{256\leftarrow 256}_{5\star 3^l})) \circ \mathsf{B} \circ \mathsf{RL}^{256\leftarrow 256}_{1\star 1} \circ \mathsf{drop})(\mathbf{R})$
	$\mathsf{TrgDec}(\mathbf{R})$	$(\mathsf{CL}^{93\leftarrow 256}_{1\star 1} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}\mathsf{ResCGLU}^{256\leftarrow 256}_{3\star 3^l})) \circ \mathsf{B} \circ \mathsf{CL}^{256\leftarrow 256}_{1\star 1} \circ \mathsf{drop})(\mathbf{R})$
many-to-many (batch)	$\mathsf{SrcEnc}(\mathbf{X}, k)$	$(\mathsf{RL}^{1024\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}(\mathsf{ResRGLU}^{k\,512\leftarrow 544}_{5\star 3^l} \circ \mathsf{A}_{\mathbf{Z}}))) \circ \mathsf{B}^k \circ \mathsf{RL}^{512\leftarrow 125}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}} \circ \mathsf{drop})(\mathbf{X})$
	$\mathsf{TrgEnc}(\mathbf{Y}, k')$	$(\mathsf{CL}^{512\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}'} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}(\mathsf{ResCGLU}^{k'\,512\leftarrow 544}_{3\star 3^l} \circ \mathsf{A}_{\mathbf{Z}'}))) \circ \mathsf{B}^{k'} \circ \mathsf{CL}^{512\leftarrow 125}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}'} \circ \mathsf{drop})(\mathbf{Y})$
	$\mathsf{TrgRec}(\mathbf{R}, k')$	$(\mathsf{RL}^{93\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}'} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}(\mathsf{ResRGLU}^{k'\,512\leftarrow 544}_{5\star 3^l} \circ \mathsf{A}_{\mathbf{Z}'}))) \circ \mathsf{B}^{k'} \circ \mathsf{RL}^{512\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}'} \circ \mathsf{drop})(\mathbf{R})$
	$\mathsf{TrgDec}(\mathbf{R}, k')$	$(\mathsf{CL}^{93\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}'} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}(\mathsf{ResCGLU}^{k'\,512\leftarrow 544}_{3\star 3^l} \circ \mathsf{A}_{\mathbf{Z}'}))) \circ \mathsf{B}^{k'} \circ \mathsf{CL}^{512\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}'} \circ \mathsf{drop})(\mathbf{R})$
		where $\mathbf{Z} = \mathsf{embed}^{32}(k)$ and $\mathbf{Z}' = \mathsf{embed}^{32}(k')$
any-to-many	$\mathsf{SrcEnc}(\mathbf{X})$	$(\mathsf{RL}^{1024\leftarrow 512}_{1\star 1} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}\mathsf{ResRGLU}^{512\leftarrow 512}_{5\star 3^l})) \circ \mathsf{B} \circ \mathsf{RL}^{512\leftarrow 93}_{1\star 1} \circ \mathsf{drop})(\mathbf{X})$
	others	same as above
many-to-many (real-time)	$\mathsf{SrcEnc}(\mathbf{X}, k)$	$(\mathsf{CL}^{1024\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}(\mathsf{ResCGLU}^{k\,512\leftarrow 544}_{3\star 3^l} \circ \mathsf{A}_{\mathbf{Z}}))) \circ \mathsf{B}^k \circ \mathsf{CL}^{512\leftarrow 125}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}} \circ \mathsf{drop})(\mathbf{X})$
	$\mathsf{TrgRec}(\mathbf{R}, k')$	$(\mathsf{CL}^{93\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}'} \circ (\bigcirc_{l'=0}^{2}(\bigcirc_{l=0}^{3}(\mathsf{ResCGLU}^{k'\,512\leftarrow 544}_{3\star 3^l} \circ \mathsf{A}_{\mathbf{Z}'}))) \circ \mathsf{B}^{k'} \circ \mathsf{CL}^{512\leftarrow 544}_{1\star 1} \circ \mathsf{A}_{\mathbf{Z}'} \circ \mathsf{drop})(\mathbf{R})$
	$\mathsf{TrgDec}(\mathbf{R}, k')$	same as above

Note that in Tab. II the layer index is omitted for simplicity of notation and each layer has a different set of free parameters even though the same symbol is used.

C. Hyperparameter Settings

λ_{r}, λ_{d}, λ_{o}, and λ_{i} were set at 1, 2000, 2000, and 1, respectively. ν and ρ were set at 0.3 and 0.3 for both the pairwise and many-to-many models. The L_1 norm $\|\mathbf{X}\|_1$ used in (6), (8), (18), and (21) was defined as a weighted norm

$$\|\mathbf{X}\|_1 = \sum_{n=1}^{N} \frac{1}{r} \sum_{j=1}^{r} \sum_{i=1}^{31} \alpha_i |x_{ij,n}|,$$

where $x_{1j,n}, \ldots, x_{28j,n}$, $x_{29j,n}$, $x_{30j,n}$ and $x_{31j,n}$ denote the entries of $\mathbf{X}$ corresponding to the 28 MCCs, log F_0, coded aperiodicity and voiced/unvoiced indicator at time n, and the weights were set at $\alpha_1 = \cdots = \alpha_{28} = \frac{1}{28}$, $\alpha_{29} = \frac{1}{10}$, and $\alpha_{30} = \alpha_{31} = \frac{1}{50}$, respectively.

All the networks were trained simultaneously with random initialization. Adam optimization [59] was used for model training where the mini-batch size was 16 and 25,000 iterations were run. The learning rate and the exponential decay rate for the first moment for Adam were set at 0.00015 and 0.9.

D. Objective Performance Measures

The test dataset consists of speech samples of each speaker reading the same sentences. Thus, the quality of a converted feature sequence can be assessed by comparing it with the feature sequence of the reference utterance.

1) Mel-cepstral distortion (MCD): Given two mel-cepstra, $\hat{\mathbf{x}} = [\hat{x}_1, \ldots, \hat{x}_{28}]^{\mathsf{T}}$ and $\mathbf{x} = [x_1, \ldots, x_{28}]^{\mathsf{T}}$, we can use the mel-cepstral distortion (MCD):

$$\mathrm{MCD[dB]} = \frac{10}{\ln 10} \sqrt{2 \sum_{i=2}^{28} (\hat{x}_i - x_i)^2}, \tag{24}$$

to measure their difference. Here, we used the average of the MCDs taken along the DTW path between converted and reference feature sequences as the objective performance measure for each test utterance.

2) Log F_0 Correlation Coefficient (LFC): To evaluate the F_0 contour of converted speech, we used the correlation coefficient between the predicted and target log F_0 contours [60] as the objective performance measure. Since the converted and reference utterances were not necessarily aligned in time, we computed the correlation coefficient after properly aligning them. Here, we used the MCC sequences $\hat{\mathbf{X}}_{1:28,1:N}$, $\mathbf{X}_{1:28,1:M}$ of converted and reference utterances to find phoneme-based

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

alignment, assuming that the predicted and reference MCCs at the corresponding frames were sufficiently close. Given the log F_0 contours $\hat{\mathbf{X}}_{29,1:N}$, $\mathbf{X}_{29,1:M}$ and the voiced/unvoiced indicator sequences $\hat{\mathbf{X}}_{31,1:N}$, $\mathbf{X}_{31,1:M}$ of converted and reference utterances, we first warp the time axis of $\hat{\mathbf{X}}_{29,1:N}$ and $\hat{\mathbf{X}}_{31,1:N}$ in accordance with the DTW path between the MCC sequences $\hat{\mathbf{X}}_{1:28,1:N}$, $\mathbf{X}_{1:28,1:M}$ of the two utterances and obtain their time-warped versions, $\tilde{\mathbf{X}}_{29,1:M}$, $\tilde{\mathbf{X}}_{31,1:M}$. We then extract the elements of $\tilde{\mathbf{X}}_{29,1:M}$ and $\mathbf{X}_{29,1:M}$ at all the time points corresponding to the voiced segments such that $\{m|\tilde{\mathbf{X}}_{31,m} = \mathbf{X}_{31,m} = 1\}$. If we use $\tilde{\mathbf{y}} = [\tilde{y}_1, \ldots, \tilde{y}_{M'}]$ and $\mathbf{y} = [y_1, \ldots, y_{M'}]$ to denote the vectors consisting of the elements extracted from $\tilde{\mathbf{X}}_{29,1:M}$ and $\mathbf{X}_{29,1:M}$, we can use the correlation coefficient between $\tilde{\mathbf{y}}$ and $\mathbf{y}$

$$R = \frac{\sum_{m'=1}^{M'}(\tilde{y}_{m'} - \tilde{\varphi})(y_{m'} - \varphi)}{\sqrt{\sum_{m'=1}^{M'}(\tilde{y}_{m'} - \tilde{\varphi})^2}\sqrt{\sum_{m'=1}^{M'}(y_{m'} - \varphi)^2}}, \qquad (25)$$

where $\tilde{\varphi} = \frac{1}{M'}\sum_{m'=1}^{M'}\tilde{y}_{m'}$ and $\varphi = \frac{1}{M'}\sum_{m'=1}^{M'}y_{m'}$, to measure the similarity between the two log F_0 contours. In the current experiment, we used the average of the correlation coefficients taken over all the test utterances as the objective performance measure for log F_0 prediction. Thus, the closer it is to 1, the better the performance. We call this measure the log F_0 correlation coefficient (LFC).

3) Local Duration Ratio (LDR): To evaluate the speaking rate and the rhythm of converted speech, we used the local slopes of the DTW path between converted and reference utterances to determine the objective performance measure. If the speaking rate and the rhythm of the two utterances are exactly the same, all the local slopes should be 1. Hence, the better the conversion, the closer the local slopes become to 1. To compute the local slopes, we undertook the following process. Given the MCC sequences $\hat{\mathbf{X}}_{1:28,1:N}$, $\mathbf{X}_{1:28,1:M}$ of converted and reference utterances, we first performed DTW on $\hat{\mathbf{X}}_{1:28,1:N}$ and $\mathbf{X}_{1:28,1:M}$. If we use $(p_1, q_1), \ldots, (p_j, q_j), \ldots, (p_J, q_J)$ to denote the obtained DTW path where $(p_1, q_1) = (1, 1)$ and $(p_J, q_J) = (M, N)$, we computed the slope of the regression line fitted to the 33 local consecutive points for each j:

$$s_j = \frac{\sum_{j'=j-16}^{j+16}(p_{j'} - \bar{p}_j)(q_{j'} - \bar{q}_j)}{\sum_{j'=j-16}^{j+16}(p_{j'} - \bar{p}_j)^2}, \qquad (26)$$

where $\bar{p}_j = \frac{1}{33}\sum_{j'=j-16}^{j+16}p_{j'}$ and $\bar{q}_j = \frac{1}{33}\sum_{j'=j-16}^{j+16}q_{j'}$, and then computed the median of $s_1, \ldots, s_J$. We call this measure the local duration ratio (LDR). The greater this ratio, the longer the duration of the converted utterance is relative to the reference utterance. In the following, we use the mean absolute difference between the LDRs and 1 (in percentages) as the overall measure for the LDRs. Thus, the closer it is to zero, the better the performance. For example, if the converted speech is 2 times faster than the reference speech, the LDR will be 0.5 everywhere, and so its mean absolute difference from 1 will be 50%.

E. Baseline Methods

1) sprocket: We chose the open-source VC system called sprocket [61] for comparison in our experiments. To run this method, we used the source code provided by its author [62]. Note that this system was used as a baseline system in the Voice Conversion Challenge (VCC) 2018 [63].

2) RNN-S2S-VC: To evaluate the effect of the fully convolutional architecture adopted in ConvS2S-VC, we implemented its recurrent counterpart [35], inspired by the architecture introduced in a S2S model-based TTS system called Tacotron [24] and considered it as another baseline. Although the original Tacotron used mel-spectra as the acoustic features, the baseline system was designed to use the same acoustic features as our system. The architecture was specifically designed as follows. The encoder consisted of a bottleneck fully-connected prenet followed by a stack of 1×1 1D GLU convolutions and a bi-directional LSTM layer. The decoder was an autoregressive content-based attention network, consisting of a bottleneck fully-connected prenet followed by a stateful LSTM layer producing the attention query, which was then passed to a stack of two uni-directional residual LSTM layers, followed by a linear projection to generate the features. Note that we replaced all rectified linear unit (ReLU) activations with GLUs as with our model. We also designed and implemented a many-to-many extension of the above RNN-based model.

F. Objective Evaluations

1) Effect of regularization: First, we evaluated the individual effects of the regularization techniques presented in Subsections III-B and III-C on both the pairwise and many-to-many models. Tabs. III, IV, and V show the average MCDs (with 95% confidence intervals), LFCs, and LDR deviations of the converted speech obtained using the pairwise and many-to-many models under different $(\lambda_{\mathsf{r}}, \lambda_{\mathsf{o}})$ settings $(0, 0)$, $(1, 0)$, and $(1, 2000)$ for the pairwise conversion model and different $(\lambda_{\mathsf{r}}, \lambda_{\mathsf{i}}, \lambda_{\mathsf{o}})$ settings $(0, 0, 0)$, $(1, 0, 0)$, $(1, 1, 0)$, and $(1, 1, 2000)$ for the many-to-many model. Owing to the limited amount of training data, the models trained without DAL did not successfully produce recognizable speech. Thus, we omit the results obtained when $\lambda_{\mathsf{d}} = 0$. As the results show, although there are a few exceptions, both the pairwise and many-to-many models performed better for most speaker pairs in terms of the MCD measure when all the regularization terms were simultaneously taken into account during training. We also found that the effects of L_{rec} and L_{oal} on the LFC and LDR measures were less significant than on the MCD measure. Fig. 5 shows examples of how each of the regularization techniques can affect the prediction of the attention matrices by the many-to-many model at test time. As these examples show, the CPL tended to have a notable effect on promoting monotonicity and continuity of the attention prediction. However, it also had a negative effect of blurring the predicted attention distributions. The OAL and IML contributed to counteracting this negative effect by sharpening the attention matrices while keeping them monotonic and continuous.

2) Comparison of normalization methods: For normalization, there are several choices including instance normalization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE III
AVERAGE MCDS [DB] OBTAINED WITH THE PAIRWISE AND MANY-TO-MANY MODELS TRAINED WITH AND WITHOUT REGULARIZATION.

Speakers		Pairwise			many-to-many			
source	target	$\lambda_r=\lambda_o=0$	$\lambda_o=0$	full version	$\lambda_r=\lambda_i=\lambda_o=0$	$\lambda_i=\lambda_o=0$	$\lambda_o=0$	full version
clb	bdl	$7.00 \pm .09$	$\mathbf{6.84 \pm .09}$	$6.93 \pm .10$	$6.98 \pm .10$	$6.76 \pm .12$	$6.67 \pm .14$	$\mathbf{6.57 \pm .12}$
	slt	$6.37 \pm .09$	$\mathbf{6.34 \pm .10}$	$6.37 \pm .08$	$6.34 \pm .08$	$6.11 \pm .05$	$6.01 \pm .09$	$\mathbf{5.98 \pm .08}$
	rms	$6.54 \pm .10$	$\mathbf{6.52 \pm .13}$	$6.53 \pm .16$	$6.27 \pm .05$	$6.23 \pm .06$	$6.44 \pm .12$	$\mathbf{6.11 \pm .08}$
bdl	clb	$6.30 \pm .06$	$6.09 \pm .08$	$\mathbf{6.03 \pm .09}$	$6.03 \pm .07$	$6.05 \pm .11$	$5.91 \pm .08$	$\mathbf{5.86 \pm .09}$
	slt	$6.61 \pm .10$	$6.66 \pm .10$	$\mathbf{6.51 \pm .12}$	$6.58 \pm .12$	$6.47 \pm .06$	$6.25 \pm .09$	$\mathbf{6.19 \pm .11}$
	rms	$6.75 \pm .11$	$\mathbf{6.68 \pm .13}$	$6.79 \pm .16$	$6.65 \pm .14$	$6.45 \pm .07$	$6.44 \pm .11$	$\mathbf{6.24 \pm .12}$
slt	clb	$6.21 \pm .12$	$6.12 \pm .08$	$\mathbf{6.03 \pm .06}$	$6.08 \pm .07$	$6.14 \pm .09$	$5.79 \pm .09$	$\mathbf{5.78 \pm .16}$
	bdl	$7.25 \pm .16$	$7.21 \pm .18$	$\mathbf{7.07 \pm .16}$	$7.10 \pm .14$	$6.99 \pm .14$	$6.80 \pm .14$	$\mathbf{6.72 \pm .14}$
	rms	$6.61 \pm .07$	$\mathbf{6.56 \pm .08}$	$6.61 \pm .09$	$6.50 \pm .09$	$6.44 \pm .12$	$6.51 \pm .10$	$\mathbf{6.27 \pm .08}$
rms	clb	$6.42 \pm .14$	$6.37 \pm .18$	$\mathbf{6.30 \pm .11}$	$6.05 \pm .06$	$6.10 \pm .11$	$5.96 \pm .09$	$\mathbf{5.93 \pm .11}$
	bdl	$7.16 \pm .10$	$7.14 \pm .14$	$\mathbf{7.08 \pm .15}$	$7.07 \pm .10$	$6.91 \pm .11$	$6.74 \pm .11$	$\mathbf{6.67 \pm .12}$
	slt	$6.78 \pm .19$	$6.72 \pm .23$	$\mathbf{6.50 \pm .07}$	$6.49 \pm .11$	$\mathbf{6.28 \pm .11}$	$6.35 \pm .16$	$6.32 \pm .16$
All pairs		$6.67 \pm .04$	$6.61 \pm .05$	$\mathbf{6.56 \pm .05}$	$6.51 \pm .04$	$6.41 \pm .04$	$6.32 \pm .04$	$\mathbf{6.22 \pm .04}$

TABLE IV
AVERAGE LFCS OBTAINED WITH THE PAIRWISE AND MANY-TO-MANY MODELS TRAINED WITH AND WITHOUT REGULARIZATION.

Speakers		Pairwise			many-to-many			
source	target	$\lambda_r=\lambda_o=0$	$\lambda_o=0$	full version	$\lambda_r=\lambda_i=\lambda_o=0$	$\lambda_i=\lambda_o=0$	$\lambda_o=0$	full version
clb	bdl	0.852	0.856	**0.869**	**0.876**	0.874	0.851	0.845
	slt	**0.846**	0.835	0.833	0.834	**0.852**	0.831	0.841
	rms	**0.829**	0.805	0.771	**0.835**	0.741	0.751	0.811
bdl	clb	**0.844**	0.815	0.810	**0.846**	0.835	0.831	0.823
	slt	0.768	0.799	**0.815**	0.775	0.792	**0.875**	0.864
	rms	**0.805**	0.750	0.800	0.759	0.742	0.788	**0.855**
slt	clb	0.838	0.796	**0.861**	0.795	0.770	**0.830**	0.812
	bdl	0.821	0.832	**0.850**	0.860	0.859	**0.861**	0.850
	rms	**0.804**	0.797	0.785	0.789	0.783	0.759	**0.799**
rms	clb	**0.850**	0.830	0.821	0.833	0.794	**0.834**	0.819
	bdl	**0.854**	0.853	0.825	0.864	0.845	**0.877**	0.870
	slt	0.792	**0.825**	0.809	0.776	0.794	**0.866**	0.826
All pairs		**0.829**	0.818	0.826	0.833	0.822	0.834	**0.838**

TABLE V
AVERAGE LDR DEVIATIONS (%) OBTAINED WITH THE PAIRWISE AND MANY-TO-MANY MODELS TRAINED WITH AND WITHOUT REGULARIZATION.

Speakers		Pairwise			many-to-many			
source	target	$\lambda_r=\lambda_o=0$	$\lambda_o=0$	full version	$\lambda_r=\lambda_i=\lambda_o=0$	$\lambda_i=\lambda_o=0$	$\lambda_o=0$	full version
clb	bdl	5.28	**3.54**	5.16	3.96	**3.48**	6.14	8.58
	slt	2.42	3.76	**1.96**	2.61	3.43	**0.55**	5.85
	rms	**2.59**	4.25	5.86	**2.34**	10.44	3.01	4.20
bdl	clb	4.96	3.87	**1.97**	5.66	5.20	3.78	**3.47**
	slt	6.16	**4.30**	6.78	6.43	4.09	**3.56**	5.53
	rms	4.22	6.41	**0.79**	**2.45**	5.43	5.25	6.06
slt	clb	2.59	**0.60**	0.81	**0.31**	0.75	2.45	0.66
	bdl	7.36	**5.59**	6.70	**1.73**	3.64	4.25	5.49
	rms	4.49	4.30	**4.08**	4.17	11.40	**2.76**	4.66
rms	clb	**1.04**	3.07	1.99	5.40	4.83	**2.94**	3.55
	bdl	**2.87**	5.10	6.87	**2.59**	3.63	5.13	10.85
	slt	**2.88**	7.16	3.29	6.93	4.62	3.38	**2.77**
All pairs		4.17	4.21	**3.47**	**3.51**	4.48	4.01	4.65

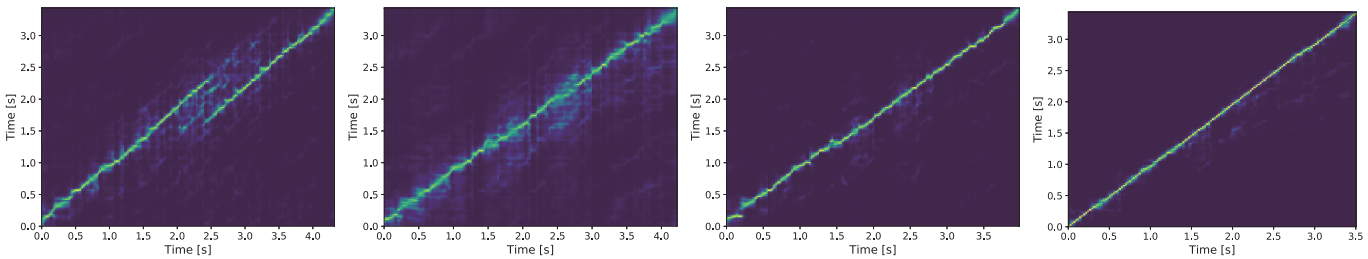

Fig. 5. Examples of the attention matrices predicted from test input female speech using the many-to-many model trained under four settings of $(\lambda_r, \lambda_o, \lambda_i)$: $(0, 0, 0)$, $(1, 0, 0)$, $(1, 2000, 0)$, and $(1, 2000, 1)$ (from left to right).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE VI
MCD [dB] COMPARISON OF NORMALIZATION METHODS.

Speakers		Pairwise			many-to-many				
source	target	IN	WN	BN	IN	CIN	WN	BN	CBN
clb	bdl	$7.76 \pm .19$	$7.10 \pm .11$	$\mathbf{6.93 \pm .10}$	$9.04 \pm .21$	$9.00 \pm .25$	$6.83 \pm .13$	$7.42 \pm .10$	$\mathbf{6.57 \pm .12}$
	slt	$6.79 \pm .10$	$6.63 \pm .10$	$\mathbf{6.37 \pm .08}$	$8.16 \pm .15$	$7.96 \pm .17$	$6.16 \pm .07$	$6.96 \pm .09$	$\mathbf{5.98 \pm .08}$
	rms	$7.57 \pm .25$	$6.71 \pm .12$	$\mathbf{6.53 \pm .16}$	$7.72 \pm .23$	$7.96 \pm .21$	$6.20 \pm .08$	$8.01 \pm .05$	$\mathbf{6.11 \pm .08}$
bdl	clb	$6.85 \pm .12$	$6.38 \pm .07$	$\mathbf{6.03 \pm .09}$	$7.93 \pm .26$	$7.52 \pm .30$	$\mathbf{5.85 \pm .07}$	$8.04 \pm .20$	$5.86 \pm .09$
	slt	$7.32 \pm .20$	$6.69 \pm .06$	$\mathbf{6.51 \pm .12}$	$7.97 \pm .18$	$7.90 \pm .21$	$6.36 \pm .08$	$8.03 \pm .25$	$\mathbf{6.19 \pm .11}$
	rms	$7.49 \pm .16$	$6.93 \pm .12$	$\mathbf{6.79 \pm .16}$	$7.86 \pm .18$	$7.69 \pm .24$	$6.31 \pm .09$	$8.06 \pm .13$	$\mathbf{6.24 \pm .12}$
slt	clb	$6.64 \pm .15$	$6.24 \pm .05$	$\mathbf{6.03 \pm .06}$	$7.68 \pm .28$	$8.26 \pm .26$	$5.82 \pm .09$	$8.06 \pm .21$	$\mathbf{5.78 \pm .16}$
	bdl	$7.95 \pm .24$	$7.28 \pm .16$	$\mathbf{7.07 \pm .16}$	$9.14 \pm .23$	$9.61 \pm .26$	$6.95 \pm .13$	$7.73 \pm .20$	$\mathbf{6.72 \pm .14}$
	rms	$7.32 \pm .19$	$6.76 \pm .12$	$\mathbf{6.61 \pm .09}$	$7.60 \pm .14$	$7.69 \pm .14$	$6.42 \pm .14$	$8.54 \pm .09$	$\mathbf{6.27 \pm .08}$
rms	clb	$6.96 \pm .15$	$6.43 \pm .12$	$\mathbf{6.30 \pm .11}$	$7.69 \pm .20$	$7.99 \pm .26$	$6.04 \pm .13$	$7.93 \pm .20$	$\mathbf{5.93 \pm .11}$
	bdl	$8.04 \pm .19$	$7.25 \pm .12$	$\mathbf{7.08 \pm .15}$	$8.97 \pm .26$	$9.16 \pm .28$	$6.93 \pm .11$	$7.88 \pm .39$	$\mathbf{6.67 \pm .12}$
	slt	$7.48 \pm .34$	$6.92 \pm .12$	$\mathbf{6.50 \pm .07}$	$8.32 \pm .21$	$8.15 \pm .25$	$6.47 \pm .17$	$8.43 \pm .62$	$\mathbf{6.32 \pm .16}$
All pairs		$7.34 \pm .07$	$6.78 \pm .05$	$\mathbf{6.56 \pm .05}$	$8.17 \pm .08$	$8.24 \pm .90$	$6.36 \pm .05$	$7.92 \pm .08$	$\mathbf{6.22 \pm .04}$

TABLE VII
LFC COMPARISON OF NORMALIZATION METHODS.

Speakers		Pairwise			many-to-many				
source	target	IN	WN	BN	IN	CIN	WN	BN	CBN
clb	bdl	0.827	0.840	**0.869**	0.556	0.456	**0.873**	0.769	0.845
	slt	0.819	0.813	**0.833**	0.547	0.794	0.830	0.815	**0.841**
	rms	0.657	0.770	**0.771**	0.475	0.338	**0.831**	0.494	0.811
bdl	clb	0.781	0.779	**0.810**	0.766	0.725	**0.829**	0.652	0.823
	slt	0.746	**0.831**	0.815	0.787	0.693	0.789	0.617	**0.864**
	rms	0.634	0.738	**0.800**	0.588	0.608	0.763	0.444	**0.855**
slt	clb	0.811	0.835	**0.861**	0.594	0.647	0.803	0.745	**0.812**
	bdl	0.792	0.819	**0.850**	0.537	0.356	**0.852**	0.748	**0.850**
	rms	0.680	0.729	**0.785**	0.504	0.327	0.783	0.293	**0.799**
rms	clb	0.738	0.761	**0.821**	0.530	0.471	0.794	0.723	**0.819**
	bdl	**0.841**	0.787	0.825	0.675	0.479	0.853	0.709	**0.870**
	slt	0.678	0.768	**0.809**	0.525	0.657	0.781	0.554	**0.826**
All pairs		0.772	0.800	**0.826**	0.582	0.572	0.818	0.672	**0.838**

TABLE VIII
LDR DEVIATION (%) COMPARISON OF NORMALIZATION METHODS.

Speakers		Pairwise			many-to-many				
source	target	IN	WN	BN	IN	CIN	WN	BN	CBN
clb	bdl	**2.10**	5.72	5.16	9.69	15.10	**4.76**	6.91	8.58
	slt	3.46	**1.36**	1.96	13.38	13.56	**2.04**	3.68	5.85
	rms	4.20	**2.69**	5.86	15.22	23.06	6.01	5.62	**4.20**
bdl	clb	4.61	4.36	**1.97**	15.59	25.97	4.83	7.91	**3.47**
	slt	9.36	**3.72**	6.78	15.49	13.45	4.14	**2.24**	5.53
	rms	3.09	2.62	**0.79**	19.81	26.39	**5.37**	8.64	6.06
slt	clb	2.01	1.52	**0.81**	10.67	15.91	4.13	3.46	**0.66**
	bdl	**3.91**	8.31	6.70	16.96	19.57	**3.12**	7.78	5.49
	rms	6.16	4.14	**4.08**	18.91	24.02	4.68	**0.01**	4.66
rms	clb	2.22	2.51	**1.99**	14.70	20.74	**2.48**	7.44	3.55
	bdl	5.23	**4.38**	6.87	16.41	13.97	**5.26**	9.68	10.85
	slt	**3.25**	5.35	3.29	10.76	13.63	4.31	5.23	**2.77**
All pairs		3.87	3.87	**3.47**	13.77	18.03	**4.46**	4.77	4.65

(IN) [64], weight normalization (WN) [65], and batch normalization (BN) [66]. For our many-to-many model, other choices include conditional IN (CIN) and conditional BN (CBN). We compared the effects of these normalization methods on both the pairwise and many-to-many models on the basis of the MCD, LFC, and LDR measures. Note that all the normalization layers in Tab. II are excluded in the WN counterparts. The average MCDs, LFCs, and LDR deviations obtained using these normalization methods are demonstrated in Tabs. VI, VII, and VIII. As the results show, BN worked better than IN and WN when applied to the pairwise conversion model especially in terms of the MCD and LFC measures. However, naively applying it directly to the many-to-many model did not work satisfactorily, as expected in Subsection IV-B. This was also the case with IN. Although CIN was found to perform poorly, CBN worked significantly better.

3) Comparisons with baseline methods: Tabs. IX, X, and XI show the average MCDs, LFCs, and LDRs obtained with the proposed and baseline methods. As Tabs. IX and X show, the pairwise versions of ConvS2S-VC and RNN-S2S-VC performed comparably to each other and significantly better than sprocket. The effect of the many-to-many extension

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE IX
AVERAGE MCDS (DB) WITH 95% CONFIDENCE INTERVALS OBTAINED WITH THE BASELINE AND PROPOSED METHODS

Speakers		sprocket	RNN-S2S		proposed (ConvS2S)	
source	target		pairwise	many-to-many	pairwise	many-to-many
clb	bdl	$6.98 \pm .10$	$6.87 \pm .09$	$6.94 \pm .15$	$6.93 \pm .10$	$\mathbf{6.57 \pm .12}$
	slt	$6.34 \pm .06$	$6.22 \pm .07$	$6.26 \pm .09$	$6.37 \pm .08$	$\mathbf{5.98 \pm .08}$
	rms	$6.84 \pm .07$	$6.45 \pm .09$	$6.23 \pm .06$	$6.53 \pm .16$	$\mathbf{6.11 \pm .08}$
bdl	clb	$6.44 \pm .10$	$6.21 \pm .13$	$6.02 \pm .12$	$6.03 \pm .09$	$\mathbf{5.86 \pm .09}$
	slt	$6.46 \pm .04$	$6.68 \pm .11$	$6.38 \pm .14$	$6.51 \pm .12$	$\mathbf{6.19 \pm .11}$
	rms	$7.24 \pm .12$	$6.69 \pm .21$	$6.35 \pm .09$	$6.79 \pm .16$	$\mathbf{6.24 \pm .12}$
slt	clb	$6.21 \pm .06$	$6.13 \pm .10$	$6.03 \pm .10$	$6.03 \pm .06$	$\mathbf{5.78 \pm .16}$
	bdl	$6.80 \pm .05$	$7.08 \pm .11$	$7.09 \pm .12$	$7.07 \pm .16$	$\mathbf{6.72 \pm .14}$
	rms	$6.87 \pm .10$	$6.64 \pm .13$	$6.38 \pm .07$	$6.61 \pm .09$	$\mathbf{6.27 \pm .08}$
rms	clb	$6.43 \pm .06$	$6.26 \pm .14$	$6.23 \pm .12$	$6.30 \pm .11$	$\mathbf{5.93 \pm .11}$
	bdl	$7.40 \pm .15$	$7.11 \pm .16$	$7.22 \pm .16$	$7.08 \pm .15$	$\mathbf{6.67 \pm .12}$
	slt	$6.76 \pm .09$	$6.53 \pm .11$	$6.41 \pm .12$	$6.50 \pm .07$	$\mathbf{6.32 \pm .16}$
All pairs		$6.73 \pm .03$	$6.57 \pm .05$	$6.46 \pm .05$	$6.56 \pm .05$	$\mathbf{6.22 \pm .04}$

TABLE X
LFCS OBTAINED WITH THE BASELINE AND PROPOSED METHODS

Speakers		sprocket	RNN-S2S		proposed (ConvS2S)	
source	target		pairwise	many-to-many	pairwise	many-to-many
clb	bdl	0.643	0.851	**0.875**	0.869	0.847
	slt	0.790	0.765	0.815	0.833	**0.845**
	rms	0.556	0.784	0.787	0.771	**0.795**
bdl	clb	0.642	0.748	**0.840**	0.810	0.826
	slt	0.632	0.738	0.797	0.815	**0.863**
	rms	0.467	0.719	0.715	0.800	**0.829**
slt	clb	0.820	0.847	0.776	**0.861**	0.827
	bdl	0.663	0.812	0.834	0.850	**0.852**
	rms	0.611	0.753	0.773	0.785	**0.806**
rms	clb	0.632	0.753	0.818	**0.821**	0.796
	bdl	0.648	0.817	0.854	0.825	**0.877**
	slt	0.674	0.783	0.785	0.809	**0.838**
All pairs		0.653	0.798	0.808	0.826	**0.836**

TABLE XI
LDR DEVIATIONS (%) OBTAINED WITH THE BASELINE AND PROPOSED METHODS

Speakers		sprocket	RNN-S2S		proposed (ConvS2S)	
source	target		pairwise	many-to-many	pairwise	many-to-many
clb	bdl	17.66	**0.52**	1.30	5.16	8.58
	slt	9.74	2.95	**1.24**	1.96	5.85
	rms	3.24	**2.27**	4.92	5.86	4.20
bdl	clb	16.65	3.52	4.94	**1.97**	3.47
	slt	**4.58**	7.76	7.18	6.78	5.53
	rms	15.20	2.65	3.72	**0.79**	6.06
slt	clb	9.25	2.63	3.49	**0.81**	0.66
	bdl	5.52	4.61	**0.01**	6.70	5.49
	rms	11.46	**3.36**	3.92	4.08	4.66
rms	clb	2.84	2.80	5.40	**1.99**	3.55
	bdl	17.76	4.53	**3.19**	6.87	10.85
	slt	11.95	6.84	4.15	3.29	**2.77**
All pairs		10.60	3.62	3.56	**3.47**	4.65

was noticeable for both ConvS2S-VC and RNN-S2S-VC, revealing the advantage of exploiting the training data of all the speakers. The many-to-many ConvS2S-VC performed better than its RNN counterpart. This demonstrates the effect of the convolutional architecture. Since sprocket is designed to keep the speaking rate and rhythm of input speech unchanged, the performance gains over sprocket in terms of the LDR measure show how well the competing methods are able to predict the speaking rate and rhythm of target speech. As Tab. XI shows, both the pairwise and many-to-many versions of RNN-S2S-VC and ConvS2S-VC obtained LDR deviations closer to 0 than sprocket.

As mentioned earlier, one important advantage of the proposed model over its RNN counterpart is that it can be trained efficiently thanks to the nature of the convolutional architectures. In fact, whereas the pairwise and many-to-many versions of the RNN-based model took about 30 and 50 hours to train, the two versions of the proposed model only took about 4 and 7 hours to train under the current experimental settings. We implemented all the algorithms in PyTorch and used a single Tesla V100 GPU with a 32.0 GB memory for training each model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE XII
AVERAGE MCDS (DB), LFCS, AND LDR DEVIATIONS (%) OBTAINED WITH THE ANY-TO-MANY SETTING UNDER A CLOSED-SET CONDITION.

Speaker pair		Measures		
source	target	MCD(dB)	LFC	LDR(%)
clb	bdl	$6.81 \pm .10$	0.907	11.33
	slt	$6.27 \pm .06$	0.842	3.70
	rms	$6.26 \pm .07$	0.794	7.54
bdl	clb	$6.10 \pm .10$	0.849	3.68
	slt	$6.55 \pm .13$	0.826	8.59
	rms	$6.49 \pm .13$	0.811	2.48
slt	clb	$6.02 \pm .09$	0.813	1.90
	bdl	$7.03 \pm .13$	0.878	5.87
	rms	$6.44 \pm .07$	0.810	4.17
rms	clb	$6.33 \pm .13$	0.815	4.73
	bdl	$6.95 \pm .09$	0.825	10.77
	slt	$6.51 \pm .13$	0.868	3.62
All pairs		$6.48 \pm .04$	0.829	4.67

TABLE XIII
AVERAGE MCDS (DB), LFCS, AND LDR DEVIATIONS (%) OBTAINED WITH THE ANY-TO-MANY SETTING UNDER AN OPEN-SET CONDITION.

Speaker pair		Measures		
source	target	MCD(dB)	LFC	LDR(%)
lnh	clb	$6.29 \pm .12$	0.778	2.74
	bdl	$7.12 \pm .12$	0.803	9.06
	slt	$6.44 \pm .07$	0.694	0.45
	rms	$6.67 \pm .09$	0.720	4.44
All pairs		$6.63 \pm .07$	0.747	4.36

4) Performance of any-to-many setting: The modifications described in Subsection IV-C make it possible to handle any-to-many VC tasks. We evaluated how these modifications actually affected the performance. Tab. XII shows the average MCDs, LFCs, and LDR deviations obtained with the any-to-many setting under a closed-set condition, where the speaker of input speech is unknown but is seen in the training data. Whereas the pairwise and the default many-to-many versions must be informed about the speaker of each input utterance at test time, the any-to-many version requires no information. This can be convenient in practical scenarios of VC applications, but because of the disadvantage in the test condition, the problem becomes more challenging. As the results show, the MCDs and LFCs obtained with the any-to-many version were only slightly worse than those obtained with the default many-to-many model despite this disadvantage. It is also worth noting that they were better than those obtained with sprocket and the pairwise versions of ConvS2S-VC and RNN-S2S, all of which were trained under a speaker-dependent closed-set condition.

We further evaluated the performance of the any-to-many model under an open-set condition where the speaker of the test utterances is unseen in the training data. We used the utterances of the speaker lnh (female) as the test input speech. The results are shown in Tab. XIII. For comparison, Tab. XIV shows results of sprocket performed on the same speaker pairs under a speaker-dependent closed-set condition. As these results show, the proposed model with the open-set any-to-many setting still performed better than sprocket, even though sprocket had an advantage in both the training and test conditions.

TABLE XIV
AVERAGE MCDS (DB), LFCS, AND LDR DEVIATIONS (%) OBTAINED WITH SPROCKET UNDER A SPEAKER-DEPENDENT CONDITION.

Speaker pair		Measures		
source	target	MCD(dB)	LFC	LDR%
lnh	clb	$6.76 \pm .08$	0.716	6.61
	bdl	$8.26 \pm .35$	0.523	13.38
	slt	$6.62 \pm .11$	0.771	5.72
	rms	$7.22 \pm .10$	0.480	4.87
All pairs		$7.21 \pm .14$	0.579	7.61

TABLE XV
MCDS AND LFCS OBTAINED WITH THE REAL-TIME SYSTEM SETTINGS.

Speaker pair		Measures	
source	target	MCD(dB)	LFC
clb	bdl	$6.77 \pm .10$	0.821
	slt	$5.95 \pm .08$	0.829
	rms	$6.35 \pm .10$	0.764
bdl	clb	$5.99 \pm .10$	0.820
	slt	$6.32 \pm .10$	0.821
	rms	$6.54 \pm .15$	0.797
slt	clb	$5.84 \pm .10$	0.818
	bdl	$6.92 \pm .16$	0.810
	rms	$6.48 \pm .09$	0.789
rms	clb	$6.17 \pm .11$	0.789
	bdl	$6.74 \pm .13$	0.856
	slt	$6.42 \pm .12$	0.833
All pairs		$6.37 \pm .04$	0.812

5) Performance with real-time system settings: We evaluated the MCDs and LFCs obtained with the many-to-many model under the real-time system setting described in Subsection III-F. The results are shown in Tab. XV. As the results show, the MCDs and LFCs were only slightly worse than those obtained with the default setting despite the disadvantage of using causal convolutions for all the networks and forcing attention matrices to be exactly diagonal (instead of having them be predicted). A comparison of Tab. XV with the results obtained with sprocket in Tabs. IX and X may also show how well the proposed method can perform with the real-time system setting.

G. Subjective Listening Tests

We conducted mean opinion score (MOS) tests to compare the sound quality and speaker similarity of the converted speech samples obtained with the proposed and baseline methods.

With the sound quality test, we included the speech samples synthesized in the same way as the proposed and baseline methods (namely the WORLD synthesizer) using the acoustic features directly extracted from real speech samples. Hence, the scores of these samples are expected to show the upper limit of the performance. We also included speech samples produced using the pairwise and many-to-many versions of RNN-S2S-VC and sprocket in the stimuli. Speech samples were presented in random orders to eliminate bias as regards the order of the stimuli. Ten listeners participated in our listening tests. Each listener was presented 6×10 utterances and asked to evaluate the naturalness by selecting 5: Excellent, 4: Good, 3: Fair, 2: Poor, or 1: Bad for each utterance. The results are shown in Fig. 6. As the results show, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

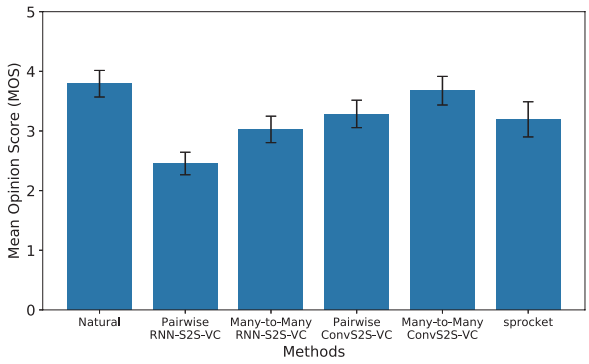

Fig. 6. Results of the MOS test for sound quality.

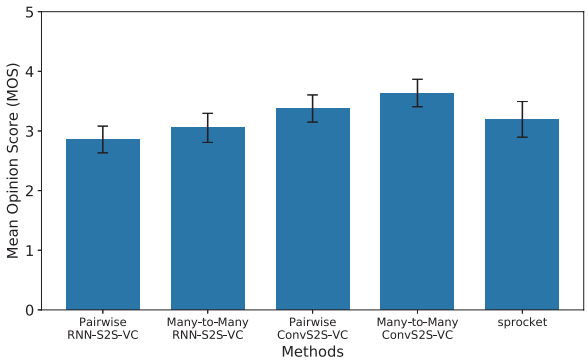

Fig. 7. Results of the MOS test for speaker similarity.

pairwise ConvS2S-VC performed slightly better than sprocket and significantly better than the two versions of RNN-S2S-VC. The many-to-many ConvS2S-VC performed better than all other methods, revealing the effect of the many-to-many extension, and reached close to the upper limit obtained with the analysis and synthesis technique.

In the speaker similarity test, each subject was given a converted speech sample and a real speech sample of the corresponding target speaker and was asked to evaluate how likely they are to be produced by the same speaker by selecting 5: Definitely, 4: Likely, 3: Fair, 2: Not very likely, or 1: Unlikely. We used converted speech samples generated by the pairwise and many-to-many versions of RNN-S2S-VC and sprocket for comparison as with the sound quality test. Each listener was presented 5×10 pairs of utterances. As the results in Fig. 7 show, both the pairwise and many-to-many versions of ConvS2S-VC performed better than all other methods.

H. Audio examples of various conversion tasks

Although we only considered a speaker identity conversion task in the above experiments, ConvS2S-VC can also be applied to other tasks. Audio samples of ConvS2S-VC tested on several tasks, including speaker identity conversion, emotional expression conversion, electrolaryngeal speech enhancement, and English accent conversion, are provided at [67]. From these examples, we can expect that ConvS2S-VC can also perform reasonably well in various tasks other than speaker identity conversion.

VI. Conclusions

This paper proposed a voice conversion (VC) method based on the ConvS2S learning framework. The proposed method provides a natural way of converting the F_0 contour, speaking rate, and rhythm as well as the voice characteristics of input speech and the flexibility of handling many-to-many, any-to-many, and real-time VC tasks without relying on automatic speech recognition (ASR) models and text annotations. Through ablation studies, we demonstrated the individual effect of each of the ideas introduced in the proposed method. Objective and subjective evaluation experiments on a speaker identity conversion task showed that the proposed method could perform better than baseline methods. Furthermore, audio examples showed the potential of the proposed method to perform well in various tasks including emotional expression conversion, electrolaryngeal speech enhancement, and English accent conversion.

Acknowledgments

This work was supported by JSPS KAKENHI 17H01763 and JST CREST Grant Number JPMJCR19A3, Japan.

References

[1] H. Kameoka, K. Tanaka, T. Kaneko, and N. Hojo, "ConvS2S-VC: Fully convolutional sequence-to-sequence voice conversion," *arXiv*, Nov. 2018.
[2] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. ICASSP*, 1998, pp. 285–288.
[3] 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.
[4] 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.
[5] Z. Inanoglu and S. Young, "Data-driven emotion conversion in spoken English," *Speech Commun.*, vol. 51, no. 3, pp. 268–283, 2009.
[6] 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.
[7] 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.
[8] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech Communication*, vol. 51, no. 10, pp. 920–932, 2009.
[9] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. SAP*, vol. 6, no. 2, pp. 131–142, 1998.
[10] 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.
[11] 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.
[12] 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.
[13] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *Proc. SLT*, 2014, pp. 19–23.
[14] 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.
[15] 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.
[16] 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.

[17] 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.

[18] ——, "Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks," in *Proc. Interspeech*, 2017, pp. 3364–3368.

[19] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "ACVAE-VC: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE Trans. ASLP*, vol. 27, no. 9, pp. 1432–1443, 2019.

[20] T. Kaneko and H. Kameoka, "Non-parallel voice conversion using cycle-consistent adversarial networks," in *Proc. EUSIPCO*, 2018, pp. 2114–2118.

[21] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *Proc. SLT*, 2018, pp. 266–273.

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

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

[24] 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.

[25] S. O. 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.

[26] S. O. 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.

[27] 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.

[28] 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.

[29] W. Ping, K. Peng, A. Gibiansky, S. O. 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.

[30] 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.

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

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

[33] 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]*, Sep. 2016.

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

[35] 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, pp. 6805–6809.

[36] 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.

[37] J.-X. Zhang, Z.-H. Ling, L.-J. Liu, Y. Jiang, and L.-R. Dai, "Sequence-to-sequence acoustic modeling for voice conversion," *IEEE/ACM Trans. ASLP*, pp. 631–644, 2019.

[38] M. Zhang, X. Wang, F. Fang, H. Li, and J. Yamagishi, "Joint training framework for text-to-speech and voice conversion using multi-source Tacotron and WaveNet," in *Proc. Interspeech*, 2019, pp. 1298–1302.

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

[40] A. Haque, M. Guo, and P. Verma, "Conditional end-to-end audio transforms," in *Proc. Interspeech*, 2018, pp. 2295–2299.

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

[42] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. MLR*, 2018, pp. 2410–2419.

[43] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. ICLR*, 2017.

[44] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "FFTNet: A real-time speaker-dependent neural vocoder," in *Proc. ICASSP*, 2018, pp. 2251–2255.

[45] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. MLR*, 2018, pp. 3918–3926.

[46] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. ICLR*, 2019.

[47] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, 2019, pp. 3617–3621.

[48] S. Kim, S. Lee, J. Song, and S. Yoon, "FloWaveNet: A generative flow for raw audio," in *Proc. MLR*, 2019, pp. 3370–3378.

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

[50] K. Tanaka, T. Kaneko, N. Hojo, and H. Kameoka, "Synthetic-to-natural speech waveform conversion using cycle-consistent adversarial networks," in *Proc. SLT*, 2018, pp. 632–639.

[51] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP*, 1992, pp. 137–140.

[52] 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.

[53] 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.

[54] 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.

[55] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *Proc. ICLR*, 2017.

[56] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "StarGAN-VC2: Rethinking conditional methods for StarGAN-based voice conversion," in *Proc. Interspeech*, 2019, pp. 679–683.

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

[58] https://github.com/r9y9/pysptk.

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

[60] D. J. Hermes, "Measuring the perceptual similarity of pitch contours," *J. Speech Lang. Hear. Res.*, vol. 41, no. 1, pp. 73–82, 1998.

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

[62] https://github.com/k2kobayashi/sprocket, (Accessed on 01/28/2019).

[63] 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 *Proc. Odyssey*, 2019.

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

[65] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Adv, NIPS*, 2016, pp. 901–909.

[66] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[67] http://www.kecl.ntt.co.jp/people/kameoka.hirokazu/Demos/convs2s-vc/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3001456, IEEE/ACM Transactions on Audio, Speech, and Language Processing

Hirokazu Kameoka Hirokazu Kameoka received B.E., M.S. and Ph.D. degrees all from the University of Tokyo, Japan, in 2002, 2004 and 2007, respectively. He is currently a Senior Distinguished Researcher at NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and an Adjunct Associate Professor at the National Institute of Informatics. From 2011 to 2016, he was an Adjunct Associate Professor at the University of Tokyo. His research interests include audio, speech, and music signal processing and machine learning. He has been an associate editor of the IEEE/ACM Transactions on Audio, Speech, and Language Processing since 2015, a Member of IEEE Audio and Acoustic Signal Processing Technical Committee since 2017, and a Member of IEEE Machine Learning for Signal Processing Technical Committee since 2019. He received 17 awards, including the IEEE Signal Processing Society 2008 SPS Young Author Best Paper Award. He is the author or co-author of about 150 articles in journal papers and peer-reviewed conference proceedings.

Kou Tanaka Kou Tanaka received his B.E. degree from Kobe University, Japan, in 2012 and his M.E. and D.E. degrees from Nara Institute of Science and Technology (NAIST), Japan, in 2014 and 2017, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science from 2015 to 2017. Since 2017, he has been a research associate at NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation. His research interests include speech signal processing with a strong focus on deep generative models.

Damian Kwaśny Damian Kwaśny Received B.E. from AGH University of Science and Technology, Poland, in 2018. In 2019 in the framework of the Vulcanus in Japan exchange program he was working as an intern at the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation. He is pursuing his M.S. degree at the AGH University of Science and Technology. His research interests include signal processing and machine learning with a focus on the deep neural networks applied to speech signal processing tasks.

Takuhiro Kaneko Takuhiro Kaneko received B.E. and M.S. from the University of Tokyo, Japan, in 2012 and 2014, respectively. He is currently a Distinguished Researcher at NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation. He is pursuing his Ph.D. at the University of Tokyo from 2017. His research interests include computer vision, signal processing, and machine learning. In particular, He is currently working on image generation, speech synthesis, and voice conversion using deep generative models. He received the ICPR2012 Best Student Paper Award at the 21st International Conference on Pattern Recognition in 2012.

Nobukatsu Hojo Nobukatsu Hojo recieved the B.E and M.E. degrees from the University of Tokyo, Japan, in 2012 and 2014, respectively. He joined NTT Media Intelligence Laboratories, Nippon Telegraph and Telephone Corporation in 2014, where he engaged in the research and development of speech synthesis. He is currently a research scientist at NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation. He is a member of Acoustical Society of Japan (ASJ), the International Speech Communication Association (ISCA) and the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan