ACVAE-VC: Non-Parallel Voice Conversion With Auxiliary Classifier Variational Autoencoder

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

***Abstract*—This paper proposes a non-parallel voice conversion (VC) method using a variant of the conditional variational autoencoder (VAE) called an auxiliary classifier VAE. The proposed method has two key features. First, it adopts fully convolutional architectures to construct the encoder and decoder networks so that the networks can learn conversion rules that capture the time dependencies in the acoustic feature sequences of source and target speech. Second, it uses information-theoretic regularization for the model training to ensure that the information in the attribute class label will not be lost in the conversion process. With regular conditional VAEs, the encoder and decoder are free to ignore the attribute class label input. This can be problematic since in such a situation, the attribute class label will have little effect on controlling the voice characteristics of input speech at test time. Such situations can be avoided by introducing an auxiliary classifier and training the encoder and decoder so that the attribute classes of the decoder outputs are correctly predicted by the classifier. We also present several ways to convert the feature sequence of input speech using the trained encoder and decoder and compare them in terms of audio quality through objective and subjective evaluations. We confirmed experimentally that the proposed method outperformed baseline non-parallel VC systems and performed comparably to an open-source parallel VC system trained using a parallel corpus in a speaker identity conversion task.**

***Index Terms*—Voice conversion (VC), variational autoencoder (VAE), non-parallel VC, auxiliary classifier VAE (ACVAE), fully convolutional network.**

I. Introduction

VOICE CONVERSION (VC) is a technique for converting para/non-linguistic information contained in a given utterance without changing the linguistic information. This technique can be applied to various tasks such as speaker-identity modification for text-to-speech (TTS) systems [2], speaking assistance [3], [4], speech enhancement [5]–[7], and pronunciation conversion [8].

One widely studied VC framework involves Gaussian mixture model (GMM)-based approaches [9]–[11], which utilize acoustic models represented by GMMs for feature mapping. Recently, a neural network (NN)-based framework [8], [12]–[20] and an exemplar-based framework using non-negative matrix factorization (NMF) [21], [22] have also attracted particular attention. Examples of the acoustic models for the NN-based framework include restricted Boltzmann machines [12], [13], fully-connected NNs [14], [15], recurrent NNs (RNNs) [16], [17] and convolutional NNs (CNNs) [8]. While many VC methods including those mentioned above require accurately aligned parallel data of source and target speech, in general scenarios, collecting parallel utterances can be a costly and time-consuming process. Even if we were able to collect parallel utterances, we typically need to perform automatic time alignment procedures, which becomes relatively difficult when there is a large acoustic gap between the source and target speech. Since many frameworks are weak with respect to the misalignment found with parallel data, careful pre-screening and manual correction is often required to make these frameworks work reliably. To sidestep these issues, this paper aims to develop a non-parallel VC method that requires no parallel utterances, transcriptions, or time alignment procedures.

The quality and conversion effect obtained with non-parallel methods are generally poorer than with methods using parallel data since there is a disadvantage related to the training condition. Thus, it would be challenging to achieve as high a quality and conversion effect with non-parallel methods as with parallel methods. Several non-parallel methods have already been proposed [19], [20], [23], [24]. For example, a method using automatic speech recognition (ASR) was proposed in [23] where the idea is to convert input speech under a restriction, namely that the posterior state probability of the acoustic model of an ASR system is preserved. Since the performance of this method depends heavily on the quality of the acoustic model of ASR, it can fail to work if ASR does not function reliably. A method using i-vectors [25], which is known to be a powerful feature for speaker verification, was proposed in [24] where the idea is to shift the acoustic features of input speech towards target speech in the i-vector space so that the converted speech is likely to be recognized as the target speaker by a speaker recognizer. While this method is also free of parallel data, one limitation is that it is applicable only to speaker identity conversion tasks.

Recently, a framework based on conditional variational autoencoders (CVAEs) [26], [27] was proposed in [19], [28]. As the name implies, VAEs are a probabilistic counterpart of autoencoders (AEs), consisting of encoder and decoder networks. Conditional VAEs (CVAEs) [27] are an extended version of VAEs with the only difference being that the encoder and decoder networks take an attribute class label c as an additional

Manuscript received November 14, 2018; revised March 18, 2019 and May 10, 2019; accepted May 11, 2019. Date of publication May 20, 2019; date of current version June 24, 2019. This work was supported by JSPS KAKENHI under Grant 17H01763. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Heiga Zen. *(Corresponding author: Hirokazu Kameoka.)*

The authors are with the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation, Atsugi 243-0198, Japan (e-mail: hirokazu.kameoka.uh@hco.ntt.co.jp; kaneko.takuhiro@lab.ntt.co.jp; tanaka.ko@lab.ntt.co.jp; hojo.nobukatsu@lab.ntt.co.jp).

Digital Object Identifier 10.1109/TASLP.2019.2917232

input. By using acoustic features associated with attribute labels as the training examples, the networks learn how to convert an attribute of source speech to a target attribute according to the attribute label fed into the decoder. While this VAE-based VC approach is notable in that it is completely free of parallel data and works even with unaligned corpora, there are three major drawbacks. Firstly, the devised networks are designed to produce acoustic features frame-by-frame, which makes it difficult to learn time dependencies in the acoustic feature sequences of source and target speech. Secondly, one well-known problem as regards VAEs is that outputs from the decoder tend to be oversmoothed. This can be problematic for VC applications since it usually results in poor quality buzzy-sounding speech. One powerful framework that can potentially overcome the weakness of VAEs involves generative adversarial networks (GANs) [29]. GANs offer a general framework for training a data generator in such a way that it can deceive a real/fake discriminator that attempts to distinguish real data and fake data produced by the generator. One natural way of alleviating the oversmoothing effect in VAEs would be to incorporate the GAN concept into VAE [30]. A non-parallel VC method based on this VAEGAN framework has already been proposed in [20]. With this method, the adversarial loss designed using a GAN discriminator is incorporated into the training loss to make the decoder outputs of a CVAE indistinguishable from real speech features. While this method is able to produce more realistic-sounding speech than the regular VAE-based method [19], as will be shown in Section IV, the audio quality and conversion effect are still limited. Thirdly, in the regular CVAEs, the encoder and decoder are free to ignore the additional input c by finding networks that can reconstruct any data without using c. In such a situation, the attribute class label c will have little effect on controlling the voice characteristics of the input speech.

To overcome these drawbacks and limitations, in this paper we describe two modifications to the conventional VAE-based approach. First, we use fully convolutional architectures to design the encoder and decoder networks so that the networks can learn conversion rules that capture short- and long-term dependencies in the acoustic feature sequences of source and target speech. Secondly, we propose using information-theoretic regularization for the model training to ensure that the attribute class information will not be lost in the conversion process. This can be done by introducing an auxiliary classifier whose role is to predict to which attribute class an input acoustic feature sequence belongs and by training the encoder and decoder so that the attribute classes of the decoder outputs are correctly predicted by the classifier. We will show in Section IV that these modifications improve on the VAE and VAEGAN frameworks [19], [20] in terms of audio quality and speaker identity conversion performance. We call the proposed VAE variant an auxiliary classifier VAE (or ACVAE). We also present several ways to convert the feature sequence of input speech using the trained encoder and decoder and compare them in terms of the effect on audio quality.

Meanwhile, we previously proposed a non-parallel VC method using a GAN variant called the cycle-consistent GAN (CycleGAN) [31], which was originally proposed as a method for translating images using unpaired training examples [32]–[34]. This method allows us to learn mappings of acoustic features between two domains through a training loss that combines an adversarial loss and a cycle-consistency loss. The former encourages the output of each mapping to be indistinguishable from real speech samples in the target domain whereas the latter encourages each mapping to preserve linguistic information by requiring that mapping input speech to the target domain and then mapping back to the source domain will result in the original input speech. Although this method was shown to work reasonably well, one major limitation is that it is designed to learn only mappings between two domains. To overcome this limitation, we subsequently proposed in [35] a non-parallel VC method incorporating an extension of CycleGAN called StarGAN [36]. This method is capable of simultaneously learning mappings between multiple domains using a single generator network where the attributes of the generator outputs are controlled by an auxiliary input. As with the CycleGAN-based method, it uses an adversarial loss and a cycle-consistency loss for generator training so that the generator outputs become indistinguishable from real speech samples in the target domain and so that each mapping preserves the linguistic information contained in the input speech. In Section IV, the proposed method is compared with this StarGAN-based VC method.

II. VAE Voice Conversion

A. Variational Autoencoder (VAE)

VAEs [26], [27] are stochastic neural network models consisting of encoder and decoder networks. The encoder network generates a set of parameters for the conditional distribution $q_\phi(\mathbf{z}|\mathbf{x})$ of a latent space variable $\mathbf{z}$ given input data $\mathbf{x}$, whereas the decoder network generates a set of parameters for the conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$ of the data $\mathbf{x}$ given the latent space variable $\mathbf{z}$. Given a training dataset $\mathcal{S} = \{\mathbf{x}_m\}_{m=1}^M$, VAEs learn the parameters of the entire network so that the encoder distribution $q_\phi(\mathbf{z}|\mathbf{x})$ becomes consistent with the posterior $p_\theta(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. By using Jensen's inequality, the log marginal distribution of the data $\mathbf{x}$ can be lower-bounded by

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \log \int q_\phi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ &\geq \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})], \end{aligned} \tag{1}$$

where the difference between the left- and right-hand sides of this inequality is equal to the Kullback-Leibler divergence $\mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})]$, which is minimized when

$$q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x}). \tag{2}$$

This means we can make $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ consistent by maximizing the lower bound of Eq. (1). One typical way of modeling $q_\phi(\mathbf{z}|\mathbf{x})$, $p_\theta(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z})$ is to assume

Gaussian distributions

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x}))), \tag{3}$$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_\theta(\mathbf{z}), \mathrm{diag}(\boldsymbol{\sigma}^2_\theta(\mathbf{z}))), \tag{4}$$

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}), \tag{5}$$

where $\boldsymbol{\mu}_\phi(\mathbf{x})$ and $\boldsymbol{\sigma}^2_\phi(\mathbf{x})$ are the outputs of an encoder network with parameter ϕ, and $\boldsymbol{\mu}_\theta(\mathbf{z})$ and $\boldsymbol{\sigma}^2_\theta(\mathbf{z})$ are the outputs of a decoder network with parameter θ. The first term of the lower bound can be interpreted as an autoencoder reconstruction error. By using a reparameterization $\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \mathbf{I})$, sampling $\mathbf{z}$ from $q_\phi(\mathbf{z}|\mathbf{x})$ can be replaced by sampling $\boldsymbol{\epsilon}$ from the distribution, which is independent of θ. This allows us to compute the gradient of the lower bound with respect to θ by using a Monte Carlo approximation of the expectation $\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\cdot]$. The second term is given as the negative KL divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$. This term can be interpreted as a regularization term that forces each element of the encoder output to be uncorrelated and normally distributed.

Conditional VAEs (CVAEs) [27] are an extended version of VAEs with the only difference being that the encoder and decoder networks can take an auxiliary variable c as an additional input. With CVAEs, Eqs. (3) and (4) are replaced with

$$q_\phi(\mathbf{z}|\mathbf{x}, c) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}, c), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x}, c))), \tag{6}$$

$$p_\theta(\mathbf{x}|\mathbf{z}, c) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_\theta(\mathbf{z}, c), \mathrm{diag}(\boldsymbol{\sigma}^2_\theta(\mathbf{z}, c))), \tag{7}$$

and the variational lower bound to be maximized becomes

$$\mathcal{J}(\phi, \theta) = \mathbb{E}_{(\mathbf{x},c)\sim p_\mathrm{d}(\mathbf{x},c)}\big[\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x},c)}[\log p(\mathbf{x}|\mathbf{z}, c)] - \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, c)\|p(\mathbf{z})]\big], \tag{8}$$

where $\mathbb{E}_{(\mathbf{x},c)\sim p_\mathrm{d}(\mathbf{x},c)}[\cdot]$ denotes the sample mean over the training examples $\{\mathbf{x}_m, c_m\}_{m=1}^M$.

B. Non-Parallel Voice Conversion Using CVAE

By letting $\mathbf{x} \in \mathbb{R}^Q$ and c be an acoustic feature vector and an attribute class label, respectively, a non-parallel VC problem can be formulated using the CVAE [19], [20]. Given a training set of acoustic features with attribute class labels $\{\mathbf{x}_m, c_m\}_{m=1}^M$, the encoder learns to map an input acoustic feature $\mathbf{x}$ and an attribute class label c to a latent space variable $\mathbf{z}$ (expected to represent phonetic information), and then the decoder reconstructs an acoustic feature $\hat{\mathbf{x}}$ conditioned on the encoded latent space variable $\mathbf{z}$ and the attribute class label c. At test time, we can generate a converted feature by feeding an acoustic feature of the input speech into the encoder and a target attribute class label into the decoder.

III. Proposed Method

A. Fully Convolutional VAE

Although the model in [19], [20] is designed to convert acoustic feature vectors frame-by-frame and fails to learn conversion rules that reflect time-dependencies in feature vector sequences, we propose extending it to a sequential version to overcome this limitation. Namely, we consider a CVAE that takes an acoustic feature vector sequence instead of a single-frame feature vector as an input and outputs a feature vector sequence of the same length. Hence, in the following we assume that $\mathbf{x} \in \mathbb{R}^{D\times N}$ is a feature vector sequence of length N and that the encoder and decoder networks are designed to generate the sequences of the means and logarithmic variances of q_ϕ and p_θ:

$$\big[\mu_\phi(\mathbf{x}, c); \log \sigma^2_\phi(\mathbf{x}, c)\big] = \mathsf{Enc}(\mathbf{x}, c), \tag{9}$$

$$\big[\mu_\theta(\mathbf{z}, c); \log \sigma^2_\theta(\mathbf{z}, c)\big] = \mathsf{Dec}(\mathbf{z}, c), \tag{10}$$

where $[;]$ denotes concatenation along the channel dimension. Although RNN-based architectures are a natural choice for modeling sequential data, model training becomes challenging as the network becomes deeper. Furthermore, it is difficult to employ parallel implementations for RNNs, and so both the training and conversion processes can be computationally demanding. Motivated by the recent success of sequential modeling using CNNs in the field of natural language processing [37] and the fact that CNNs are well suited to parallel implementations, we use fully convolutional networks to design Enc and Dec, as detailed in III-D.

B. Auxiliary Classifier VAE

We hereafter assume that a class label comprises one or more categories, each consisting of multiple classes. We thus represent c as a concatenation of one-hot vectors, each of which is filled with 1 at the index of a class in a certain category and with 0 everywhere else. For example, if we consider speaker identities as the only class category, c will be represented as a single one-hot vector, where each element is associated with a different speaker.

The regular CVAEs impose no restrictions on the manner in which the encoder and decoder may use the attribute class label c. Hence, the encoder and decoder are free to ignore c by finding distributions satisfying $q_\phi(\mathbf{z}|\mathbf{x}, c) = q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z}, c) = p_\theta(\mathbf{x}|\mathbf{z})$. This can occur for instance when the encoder and decoder have sufficient capacity to reconstruct any data without using c. In such a situation, c will have little effect on controlling the voice characteristics of input speech. To avoid such situations, we introduce information-theoretic regularization [38] to assist the decoder output to be correlated as far as possible with c.

The mutual information for $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}, c)$ and c conditioned on $\mathbf{z}$ can be written as

$$\begin{aligned}\mathcal{I}(\theta) &= \sum_{c'} \int p(c', \mathbf{x}) \log \frac{p(c', \mathbf{x})}{p(c')p(\mathbf{x})} \mathrm{d}\mathbf{x} \\ &= \sum_{c'} \int p(\mathbf{x})p(c'|\mathbf{x}) \log p(c'|\mathbf{x}) \mathrm{d}\mathbf{x} + H \\ &= \mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z},c), c'\sim p(c|\mathbf{x})}[\log p(c'|\mathbf{x})] + H,\end{aligned} \tag{11}$$

where H represents the entropy of c, which can be considered a constant term. In practice, $\mathcal{I}(\theta)$ is hard to optimize directly since it requires access to the posterior $p(c|\mathbf{x})$. Fortunately, we can obtain a lower bound of the first term of $\mathcal{I}(\theta)$ by introducing

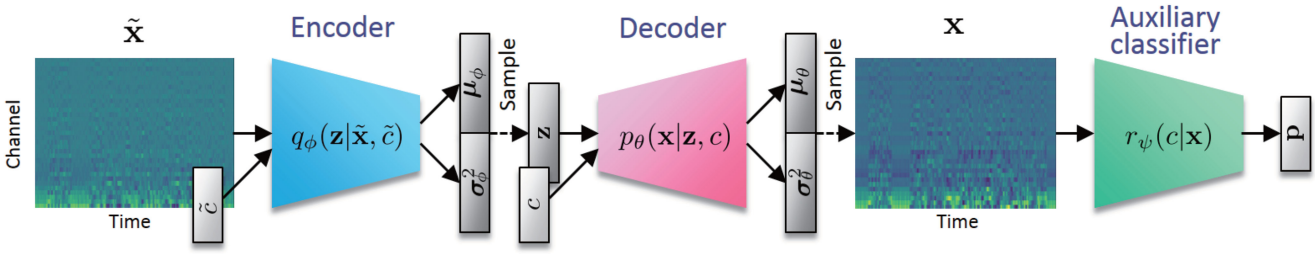


Fig. 1. Illustration of the structure of the proposed ACVAE-VC.

an auxiliary distribution $r(c|\mathbf{x})$

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z},c),c'\sim p(c|\mathbf{x})}[\log p(c'|\mathbf{x})] \\
&\quad = \mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z},c),c'\sim p(c|\mathbf{x})}\left[\log \frac{r(c'|\mathbf{x})p(c'|\mathbf{x})}{r(c'|\mathbf{x})}\right] \\
&\quad \geq \mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z},c),c'\sim p(c|\mathbf{x})}[\log r(c'|\mathbf{x})] \\
&\quad = \mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z},c)}[\log r(c|\mathbf{x})]. \qquad (12)
\end{aligned}$$

This technique of lower bounding mutual information is called variational information maximization [39]. The last line of Eq. (12) follows from the lemma presented in [38]. The equality holds in Eq. (12) when $r(c|\mathbf{x}) = p(c|\mathbf{x})$. Hence, maximizing the lower bound Eq. (12) with respect to $r(c|\mathbf{x})$ corresponds to approximating $p(c|\mathbf{x})$ by $r(c|\mathbf{x})$ as well as approximating $\mathcal{I}(\theta)$ by this lower bound. We can therefore indirectly increase $\mathcal{I}(\theta)$ by increasing the lower bound with respect to $p_\theta(\mathbf{x}|\mathbf{z},c)$ and $r(c|\mathbf{x})$. One way to do this involves expressing $r(c|\mathbf{x})$ using an NN and training it along with $q_\phi(\mathbf{z}|\mathbf{x},c)$ and $p_\theta(\mathbf{x}|\mathbf{z},c)$. Hereafter, we use $r_\psi(c|\mathbf{x})$ to denote the auxiliary classifier NN with parameter ψ. As detailed in III-D, we also design the auxiliary classifier using a fully convolutional network, which takes an acoustic feature sequence as the input and generates a sequence of class probabilities. The regularization term that we would like to maximize with respect to ϕ, θ and ψ becomes

$$\begin{aligned}
&\mathcal{Q}(\phi,\theta,\psi) \\
&= \mathbb{E}_{(\tilde{c},\tilde{x})\sim p_{\mathrm{d}}(\tilde{\mathbf{x}},\tilde{c}),\mathbf{z}\sim q_\phi(\mathbf{z}|\tilde{\mathbf{x}},\tilde{c})}\left[\mathbb{E}_{c\sim p(c),\mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z},c)}[\log r_\psi(c|\mathbf{x})]\right], \qquad (13)
\end{aligned}$$

where $\mathbb{E}_{(\tilde{\mathbf{x}},\tilde{c})\sim p_{\mathrm{d}}(\tilde{\mathbf{x}},\tilde{c})}[\cdot]$ denotes the sample mean over the training examples $\{\tilde{\mathbf{x}}_m,\tilde{c}_m\}_{m=1}^M$. This criterion can be understood as follows. For each training example, the encoder generates $\mathbf{z}$ and then the decoder produces a set of samples $\mathbf{x}$ by using all possible target labels c. The classifier then infers the log probability that each of the generated samples $\mathbf{x}$ correctly belong to the corresponding class. $\mathcal{Q}(\phi,\theta,\psi)$ is defined as the mean of these log probabilities. Here, it should be noted that to compute $\mathcal{Q}(\phi,\theta,\psi)$, we must sample $\mathbf{z}$ from $q_\phi(\mathbf{z}|\mathbf{x},c)$ and $\mathbf{x}$ from $p_\theta(\mathbf{x}|\mathbf{z},c)$. Fortunately, we can use the same reparameterization trick as in II-A to compute the gradients of $\mathcal{Q}(\phi,\theta,\psi)$ with respect to ϕ and θ. Since we can also use the training examples $\{\tilde{\mathbf{x}}_m,\tilde{c}_m\}_{m=1}^M$ to train the auxiliary classifier $r_\psi(c|\mathbf{x})$, we include the cross-entropy

$$\mathcal{R}(\psi) = \mathbb{E}_{(\tilde{\mathbf{x}},\tilde{c})\sim p_{\mathrm{d}}(\tilde{\mathbf{x}},\tilde{c})}[\log r_\psi(\tilde{c}|\tilde{\mathbf{x}})], \qquad (14)$$

in our training criterion. The entire training criterion is thus given by

$$\mathcal{J}(\phi,\theta) + \lambda_{\mathcal{Q}}\mathcal{Q}(\phi,\theta,\psi) + \lambda_{\mathcal{R}}\mathcal{R}(\psi), \qquad (15)$$

where $\lambda_{\mathcal{Q}} \geq 0$ and $\lambda_{\mathcal{R}} \geq 0$ are regularization parameters, which weigh the importances of the regularization terms relative to the VAE training criterion $\mathcal{J}(\phi,\theta)$.

Although the idea of using an auxiliary classifier for GAN-based image synthesis [36], [40] and voice conversion [35] has already been proposed, to the best of our knowledge, it has yet to be proposed for use with the VAE framework. We call our VAE variant an auxiliary classifier VAE (or ACVAE).

C. Conversion Process

Although it would be interesting to develop an end-to-end model by directly using a time-domain signal or a magnitude spectrogram as $\mathbf{x}$, given the recent significant advances in high-quality neural vocoder systems [41]–[51], we still find it useful to develop VC systems that are designed to convert acoustic features such as the mel-cepstral coefficients (MCCs) [52], since we can expect to generate high-fidelity signals by using a neural vocoder once acoustic features are obtained. In such systems, the model size for the convertor can be made small enough to allow the system to work well even when a limited amount of training data is available and possibly allow real-time implementations. Motivated by this, in this paper we use as $\mathbf{x}$ a sequence of MCCs computed from a spectral envelope sequence obtained using WORLD [53].

There are several ways to convert an input feature sequence $\mathbf{x}$ at test time after training ϕ and θ. One simple way involves using the means of the encoder and decoder distributions

$$\hat{\mathbf{x}}_{\mathrm{mean}} = \boldsymbol{\mu}_\theta(\boldsymbol{\mu}_\phi(\mathbf{x},c),\hat{c}), \qquad (16)$$

where c and $\hat{c}$ denote the source and target attribute class labels, respectively. Once a feature sequence is obtained, we can reconstruct a time-domain signal with a vocoder. However, the converted feature sequence $\hat{\mathbf{x}}_{\mathrm{mean}}$ obtained with this procedure can be over-smoothed as with other conventional VC methods and can result in buzzy-sounding synthetic speech. Since the

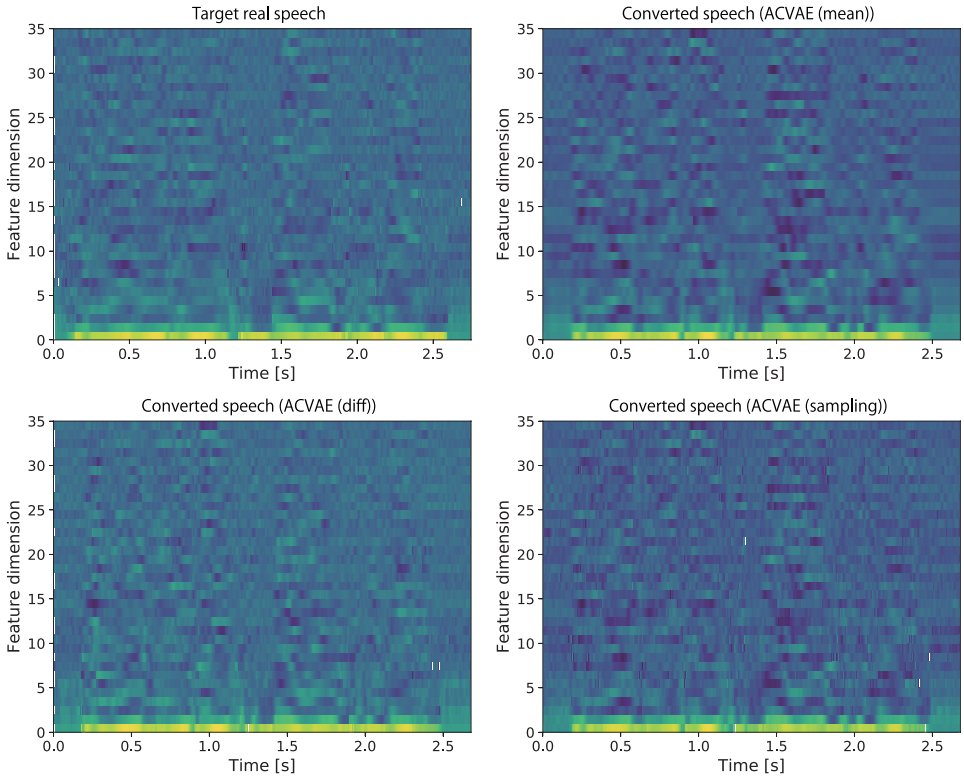


Fig. 2. MCC sequences generated using Eqs. (16), (18) and (20) along with those of the real speech of the target speaker reading the same sentence.

reconstructed feature sequence

$$\bar{\mathbf{x}}_{\mathrm{mean}} = \boldsymbol{\mu}_\theta(\boldsymbol{\mu}_\phi(\mathbf{x}, c), c), \tag{17}$$

can also be over-smoothed, one reasonable way of avoiding buzzy-sounding speech would be to add the difference between $\mathbf{x}$ and $\bar{\mathbf{x}}_{\mathrm{mean}}$ to $\hat{\mathbf{x}}_{\mathrm{mean}}$

$$\hat{\mathbf{x}}_{\mathrm{diff}} = \mathbf{x} - \bar{\mathbf{x}}_{\mathrm{mean}} + \hat{\mathbf{x}}_{\mathrm{mean}}, \tag{18}$$

so that the spectral details of the input speech are transplanted into its converted version. While $\mathbf{x} - \bar{\mathbf{x}}_{\mathrm{mean}}$ can be thought of as the spectral detail contained in the input speech, $\hat{\mathbf{x}}_{\mathrm{mean}} - \bar{\mathbf{x}}_{\mathrm{mean}}$ can be viewed as the predicted spectral difference between source and target speech. Thus, Eq. (18) can also be interpreted as the process of adding the spectral difference $\hat{\mathbf{x}}_{\mathrm{mean}} - \bar{\mathbf{x}}_{\mathrm{mean}}$ to the raw input $\mathbf{x}$. It should be noted that a similar idea has already been introduced in the GMM-based framework [54]. Another way to produce feature sequences with realistic spectral details would be to use random sampling according to the encoder and decoder distributions

$$\hat{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x}, c), \tag{19}$$

$$\hat{\mathbf{x}}_{\mathrm{samp}} \sim p_\theta(\mathbf{x}|\hat{\mathbf{z}}, \hat{c}). \tag{20}$$

Examples of the MCC sequences generated using Eqs. (16), (18) and (20) are shown in Fig. 2. As can be seen from these examples, while $\hat{\mathbf{x}}_{\mathrm{mean}}$ has been over-smoothed, both $\hat{\mathbf{x}}_{\mathrm{diff}}$ and $\hat{\mathbf{x}}_{\mathrm{samp}}$ have succeeded in recovering spectral details that resemble those found in real speech. The effects of these methods on audio quality are evaluated in Section IV.

D. Network Architectures

As detailed in Figs. 3–6, all the networks are designed using fully convolutional architectures with gated linear units (GLUs) [37]. The output of the GLU block used in the proposed model is defined as $\mathsf{GLU}(\mathbf{X}) = \mathsf{B}_1(\mathsf{L}_1(\mathbf{X})) \odot \sigma(\mathsf{B}_2(\mathsf{L}_2(\mathbf{X})))$ where $\mathbf{X}$ is the layer input, L_1 and L_2 denote convolution layers, B_1 and B_2 denote batch normalization layers, and σ denotes a sigmoid gate function. We used 2D or 1D convolutions to design the convolution layers in the encoder, decoder and auxiliary classifier, where we treated $\mathbf{x}$ as an image of size $D \times N$ with 1 channel in the 2D case and as an image of size $1 \times N$ with D channel in the 1D case. The dimension of the latent variable space was set at J and so $\mathbf{z}$ is treated as an image of size $1 \times N'$ with J channels. At each GLU block in the encoder and decoder, a broadcast version of c is appended along the channel dimension to the output of the previous GLU block. It should be noted that since the entire architecture is fully convolutional with no fully-connected layers, it can take an entire sequence with an arbitrary length as an input and generate an acoustic feature sequence of the same length. This can be ensured by properly padding zeros before each convolution. The final output of the auxiliary classifier is given by the product of all the elements of the output produced from a convolution layer and followed by a softmax operation.

It should be noted that the conventional VAE-based methods [19], [20] are designed so that the encoder is not conditioned on a speaker label and our method can also be designed in the same way. In our preliminary experiments, we tested both the unconditional and conditional versions and found their performance to be comparable.

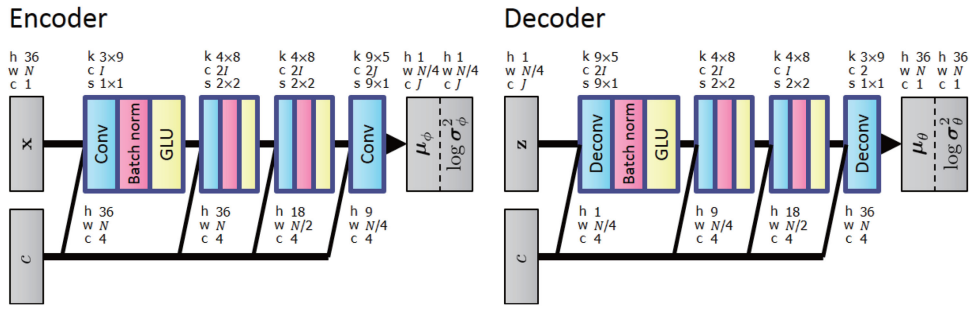


Fig. 3. Network architectures of the encoder and decoder designed using 2D convolution layers. Here, the input and output of each of the networks are interpreted as images, where "h", "w" and "c" denote the height, width and channel number, respectively. "Conv", "Batch norm", "GLU", "Deconv" "Softmax" and "Product" denote convolution, batch normalization, gated linear unit, transposed convolution, softmax, and product pooling layers, respectively. "k", "c" and "s" denote the kernel size, output channel number and stride size of a convolution layer, respectively.

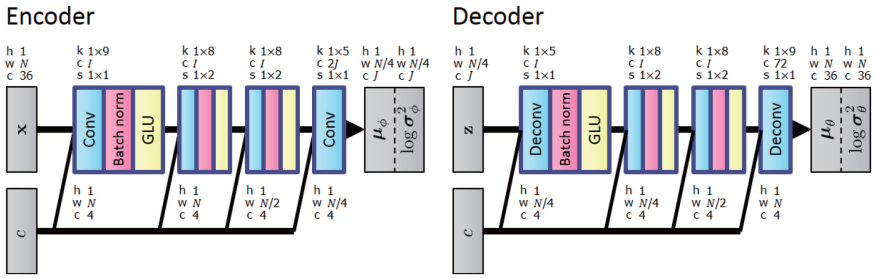


Fig. 4. Network architectures of the encoder and decoder designed using 1D convolution layers. The notation follows that in Fig. 3.

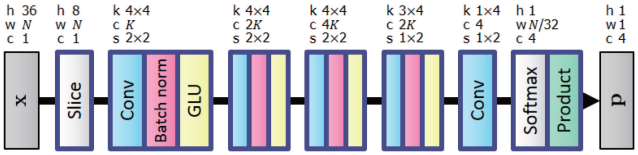


Fig. 5. Network architecture of the auxiliary classifier designed using 2D convolution layers. "Slice" denotes an operation of extracting only the lower region of an input. The notation follows that in Fig. 3.

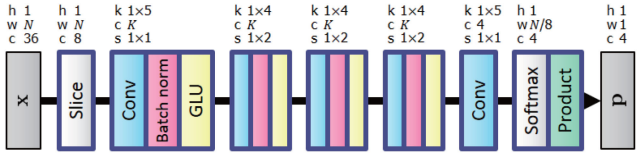


Fig. 6. Network architecture of the auxiliary classifier designed using 1D convolution layers. "Slice" denotes an operation of extracting only the lower channels of an input. The notation follows that in Fig. 3.

IV. Experiments

A. Experimental Settings

To confirm the effects of the ideas presented in III-A, III-C and III-B, we conducted objective and subjective evaluation experiments involving a non-parallel speaker identity conversion task. We used the Voice Conversion Challenge (VCC) 2018 dataset [55], which consists of recordings of six female and six male US English speakers. We used a subset of speakers for training and evaluation. Specifically, we selected two female speakers, 'SF1' and 'SF2', and two male speakers, 'SM1' and 'SM2'. Thus, c is represented as a four-dimensional one-hot vector and in total there were twelve different combinations of source and target speakers. The audio files for each speaker were manually segmented into 116 short sentences (about 7 minutes long in total) where 81 and 35 sentences (respectively, about 5 and 2 minutes long in total) were provided as training and evaluation sets, respectively. The training and test datasets consisted of speech samples of each speaker reading the same sentences. Although this means we could actually construct a parallel corpus, we took care not to take advantage of it to simulate a non-parallel training scenario. All the speech signals were sampled at 22,050 Hz. For each utterance, a spectral envelope, a logarithmic fundamental frequency (log F_0), and aperiodicities (APs) were extracted every 5 ms using the WORLD analyzer [53], [56]. 36 mel-cepstral coefficients (MCCs) were then extracted from each spectral envelope using the Speech Processing Toolkit (SPTK) [57]. The F_0 contours were converted using the logarithm Gaussian normalized transformation described in [58]. The aperiodicities were used directly without modification. The network architectures we investigated in these experiments are shown in detail in Figs. 3–6. The signals of the converted speech were obtained using the methods described in III-C.

B. Pre- and Post-Processing

At training time, each element $x_{d,n}$ of the feature sequence $\mathbf{x}$ of each speaker was normalized to

$$x_{d,n} \leftarrow \frac{x_{d,n} - \alpha_d}{\beta_d}, \tag{21}$$

where α_d and β_d are the mean and standard deviation of the d-th dimension of the feature vectors within all the voiced segments of the training samples of the same speaker. At test time, the mean and variance of the generated feature sequence were adjusted so that they matched the pretrained mean and variance of the feature vectors of the target speaker.

C. Hyperparameter Settings

The regularization parameters were set at $\lambda_{\mathcal{Q}} = \lambda_{\mathcal{R}} = 1$ when θ and ϕ were updated, and at $\lambda_{\mathcal{Q}} = 0$, $\lambda_{\mathcal{R}} = 1$ when ψ was updated. All the networks were trained simultaneously with random initialization. Adam optimization [59] was used for model training where the mini-batch size was 8 and 12,000 iterations were run. The learning rate for Adam was set at 0.001 for the encoder/decoder and at 2.5×10^{-5} for the auxiliary classifier. The exponential decay rate for the first moment was set at 0.9 for the encoder/decoder and at 0.5 for the auxiliary classifier. The network architectures we tested in our experiments are shown in Figs. 3–6.

D. Objective Performance Measure

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 target speaker reading the same sentence.

Given two mel-cepstra, $[x_1, \ldots, x_D]^{\mathsf{T}}$ and $[y_1, \ldots, y_D]^{\mathsf{T}}$, we can use the mel-cepstral distortion (MCD):

$$\text{MCD [dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{d=2}^{D}(x_d - y_d)^2}, \tag{22}$$

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

E. Baseline Methods

We chose the VAE-based [19], VAEGAN-based [20] and our previously proposed StarGAN-based [35] non-parallel VC methods for comparison. To clarify how close the proposed method can get to the performance achieved by one of the best performing parallel VC methods, we also chose an open-source system called "sprocket" [60] for comparison. Since sprocket is a parallel VC system, we used a parallel corpus only for the model training for sprocket. To run these methods, we used the source codes provided by the authors [61]–[63].

F. Objective Evaluations

1) Comparison of Different Network Configurations: First, we evaluated the performance of the proposed method with six different network configurations. The detailed settings for these configurations are shown in Figs. 3–6 and Table I, where I, J and K are the channel numbers of the middle layers in the encoder, decoder and auxiliary classifier, respectively. Types 1, 2 and 6 adopt 2D convolution layers and Types 3, 4 and 5 adopt 1D convolution layers in the encoder and decoder. Types 1, 2, 3 and 4 adopt 2D convolution layers and Types 5 and 6 adopt 1D convolution layers in the auxiliary classifier.

TABLE I
CONFIGURATIONS FOR ARCHITECTURE TYPES 1–6

Type	encoder/decoder			auxiliary classifier	
	convolution	I	J	convolution	K
1	2D (Fig. 3)	5	8	2D (Fig. 5)	8
2	2D (Fig. 3)	8	16	2D (Fig. 5)	16
3	1D (Fig. 4)	5	32	2D (Fig. 5)	16
4	1D (Fig. 4)	8	32	2D (Fig. 5)	16
5	1D (Fig. 4)	5	32	1D (Fig. 6)	16
6	2D (Fig. 3)	5	8	1D (Fig. 6)	16

Table II shows the average MCDs with 95% confidence intervals obtained using the six network configurations for each of the source and target speaker combinations. As the results show, the MCDs were all comparable, indicating that the choices between the 2D and 1D models did not significantly affect the quality of the converted speech.

2) Effect of Fully Convolutional Architecture Design: The proposed model of Architecture Type 5 reduces to a frame-independent model that uses fully-connected networks to convert feature vectors frame-by-frame, when the kernel sizes of all the convolutions in the encoder, decoder and auxiliary classifier are set at 1×1. To confirm the effect of our fully convolutional architecture design, we compared the MCDs obtained with the proposed model of Architecture Type 5 and its frame-independent counterpart. Table III shows the results obtained with these models. As the results show, the proposed model obtained significantly smaller MCDs than the frame-independent model for all speaker combinations, thus showing the advantage of capturing time dependencies.

3) Effect of Auxiliary Classifier: To confirm the effect of the incorporation of the auxiliary classifier, we compared the proposed model with and without the auxiliary classifier. Table IV shows the MCDs obtained using the proposed model of Architecture Type 1. As the results show, the proposed model performed significantly better when using the auxiliary classifier, thus demonstrating its positive effect.

4) Comparison of Conversion Methods: We compared the performance of the proposed method obtained when using Eqs. (16) and (18). Table IV compares the MCDs obtained using the model of Architecture Type 1. As the results show, it transpired that using the spectral detail transplantation technique did not have a positive impact as regards improving the MCDs. This was not unexpected since the spectral details of source and target speech are usually different.

5) Comparisons With Baseline Methods: According to the results of the objective evaluations described above, we chose the proposed method using Architecture Type 1 with no spectral detail transplantation for comparison in the following experiments. Table VI shows the MCDs obtained with the proposed and baseline methods. As the results show, the proposed method significantly outperformed the other non-parallel methods for all the source and target speaker pairs. It is also worth noting that the proposed method performed better than even the parallel VC method for most of the speaker pairs.

TABLE II
MCDs [dB] WITH 95% CONFIDENCE INTERVALS OBTAINED WITH THE PROPOSED METHOD USING DIFFERENT ARCHITECTURE TYPES

Speakers		Architecture type					
source	target	1	2	3	4	5	6
SF1	SM1	$\mathbf{6.70 \pm 0.129}$	6.80 ± 0.127	6.75 ± 0.134	6.96 ± 0.143	6.79 ± 0.126	6.87 ± 0.148
	SF2	6.57 ± 0.134	6.63 ± 0.126	6.73 ± 0.133	6.66 ± 0.128	$\mathbf{6.51 \pm 0.138}$	6.57 ± 0.135
	SM2	$\mathbf{6.97 \pm 0.124}$	7.10 ± 0.122	7.10 ± 0.106	7.19 ± 0.117	7.05 ± 0.119	7.08 ± 0.113
SM1	SF1	$\mathbf{7.01 \pm 0.108}$	7.09 ± 0.108	7.08 ± 0.112	7.23 ± 0.125	7.03 ± 0.128	7.13 ± 0.128
	SF2	$\mathbf{6.29 \pm 0.133}$	6.43 ± 0.126	6.49 ± 0.109	6.41 ± 0.113	$\mathbf{6.29 \pm 0.135}$	6.37 ± 0.104
	SM2	$\mathbf{6.64 \pm 0.111}$	6.76 ± 0.107	6.74 ± 0.116	6.69 ± 0.123	6.67 ± 0.107	6.66 ± 0.115
SF2	SF1	$\mathbf{6.94 \pm 0.115}$	7.02 ± 0.106	6.99 ± 0.124	6.98 ± 0.122	6.95 ± 0.127	6.98 ± 0.116
	SM1	$\mathbf{6.36 \pm 0.108}$	6.55 ± 0.114	6.38 ± 0.108	6.46 ± 0.112	6.45 ± 0.124	6.48 ± 0.108
	SM2	$\mathbf{6.85 \pm 0.137}$	6.96 ± 0.140	6.89 ± 0.133	6.96 ± 0.146	6.87 ± 0.120	6.92 ± 0.146
SM2	SF1	7.24 ± 0.151	7.30 ± 0.164	7.26 ± 0.154	7.38 ± 0.165	$\mathbf{7.17 \pm 0.145}$	7.34 ± 0.159
	SM1	6.66 ± 0.133	6.72 ± 0.126	6.67 ± 0.128	6.69 ± 0.126	$\mathbf{6.63 \pm 0.129}$	6.70 ± 0.125
	SF2	6.64 ± 0.152	6.73 ± 0.158	6.84 ± 0.142	6.81 ± 0.147	$\mathbf{6.58 \pm 0.136}$	6.72 ± 0.136

TABLE III
MCDs [dB] OBTAINED WITH FULLY CONVOLUTIONAL AND FRAME-INDEPENDENT MODELS

Speakers		Layer type	
source	target	frame-independent	fully convolutional
SF1	SM1	9.07 ± 0.197	$\mathbf{6.79 \pm 0.088}$
	SF2	8.73 ± 0.121	$\mathbf{6.51 \pm 0.096}$
	SM2	9.25 ± 0.189	$\mathbf{7.05 \pm 0.081}$
SM1	SF1	8.94 ± 0.166	$\mathbf{7.03 \pm 0.090}$
	SF2	8.33 ± 0.214	$\mathbf{6.29 \pm 0.095}$
	SM2	8.68 ± 0.165	$\mathbf{6.67 \pm 0.071}$
SF2	SF1	8.78 ± 0.211	$\mathbf{6.95 \pm 0.104}$
	SM1	8.54 ± 0.198	$\mathbf{6.45 \pm 0.103}$
	SM2	8.75 ± 0.183	$\mathbf{6.87 \pm 0.102}$
SM2	SF1	9.03 ± 0.202	$\mathbf{7.17 \pm 0.098}$
	SM1	8.65 ± 0.182	$\mathbf{6.63 \pm 0.083}$
	SF2	8.43 ± 0.213	$\mathbf{6.58 \pm 0.088}$

TABLE IV
MCDs [dB] OBTAINED WITH THE PROPOSED MODEL WITH AND WITHOUT THE AUXILIARY CLASSIFIER

Speakers		Auxiliary classifier	
source	target	not included	included
SF1	SM1	7.48 ± 0.150	$\mathbf{6.70 \pm 0.129}$
	SF2	7.38 ± 0.163	$\mathbf{6.57 \pm 0.134}$
	SM2	7.70 ± 0.140	$\mathbf{6.97 \pm 0.124}$
SM1	SF1	7.64 ± 0.144	$\mathbf{7.01 \pm 0.108}$
	SF2	6.93 ± 0.148	$\mathbf{6.29 \pm 0.133}$
	SM2	7.25 ± 0.136	$\mathbf{6.64 \pm 0.111}$
SF2	SF1	7.83 ± 0.164	$\mathbf{6.94 \pm 0.115}$
	SM1	7.25 ± 0.151	$\mathbf{6.36 \pm 0.108}$
	SM2	7.49 ± 0.167	$\mathbf{6.85 \pm 0.137}$
SM2	SF1	7.82 ± 0.176	$\mathbf{7.24 \pm 0.151}$
	SM1	7.22 ± 0.150	$\mathbf{6.66 \pm 0.133}$
	SF2	7.15 ± 0.170	$\mathbf{6.64 \pm 0.152}$

6) Comparison of Modulation Spectra: The modulation spectra of MCC sequences are known to be quantities that are closely related to perceived quality and naturalness of speech [64]. By definition, the modulation spectrum of a feature sequence represents the interdependencies of the elements in the sequence. Thus, our fully convolutional architecture design is expected to be effective in generating feature sequences with modulation spectra similar to those of real speech.

In the following, we use the abbreviations ACVAE (mean), ACVAE (diff) and ACVAE (sampling) to indicate the proposed method using Eqs. (16), (18) and (20) for the conversion process. Fig. 7 shows an example of the average modulation spectra of the converted MCC sequences obtained with the proposed and baseline methods along with those of the real speech of the target speaker. As can be seen from these graphs, the modulation spectra obtained with the proposed method provide a good match with those of the target speaker below 20 Hz, however they tend to deviate in the higher frequency range. Meanwhile, it transpired that the modulation spectra obtained with the StarGAN-based method were relatively close to those of real speech over the entire frequency range, thanks to both an adversarial training strategy and a fully convolutional architecture design. By contrast, the modulation spectra obtained with the VAE-based and VAEGAN-based methods were relatively distant from those of real speech. This could be because these methods employed frame-independent models, which were incapable of capturing temporal dependencies. Even though the VAEGAN-based method uses adversarial training for model training, it will have a limited effect on obtaining realistic modulation spectra as long as the model is incapable of learning temporal dependencies.

It is interesting to compare the modulation spectra obtained with ACVAE (mean/diff/sampling). As can be seen from Fig. 7, the spectral detail transplanting process and the random sampling process exhibited similar effects in recovering realistic modulation spectra especially in the higher frequency range.

TABLE V
MCDs [dB] OBTAINED WITH THE PROPOSED METHOD USING DIFFERENT CONVERSION METHODS

Speakers		Conversion method		
source	target	Eq. (16)	Eq. (18)	Eq. (20)
SF1	SM1	$\mathbf{6.70 \pm 0.129}$	7.26 ± 0.130	7.17 ± 0.124
	SF2	$\mathbf{6.57 \pm 0.134}$	7.01 ± 0.130	7.07 ± 0.131
	SM2	$\mathbf{6.97 \pm 0.124}$	7.58 ± 0.135	7.50 ± 0.113
SM1	SF1	$\mathbf{7.01 \pm 0.108}$	7.52 ± 0.111	7.55 ± 0.097
	SF2	$\mathbf{6.29 \pm 0.133}$	6.74 ± 0.118	6.87 ± 0.136
	SM2	$\mathbf{6.64 \pm 0.111}$	7.16 ± 0.110	7.20 ± 0.108
SF2	SF1	$\mathbf{6.94 \pm 0.115}$	7.36 ± 0.104	7.52 ± 0.111
	SM1	$\mathbf{6.36 \pm 0.108}$	6.88 ± 0.119	6.88 ± 0.115
	SM2	$\mathbf{6.85 \pm 0.137}$	7.42 ± 0.112	7.38 ± 0.118
SM2	SF1	$\mathbf{7.24 \pm 0.151}$	7.69 ± 0.130	7.74 ± 0.139
	SM1	$\mathbf{6.66 \pm 0.133}$	7.06 ± 0.126	7.12 ± 0.128
	SF2	$\mathbf{6.64 \pm 0.152}$	7.11 ± 0.144	7.14 ± 0.148

G. Subjective Listening Tests

We conducted a mean opinion score (MOS) test to compare the sound quality of the converted speech samples obtained

TABLE VI
COMPARISONS WITH THE CONVENTIONAL NON-PARALLEL AND PARALLEL METHODS

Speakers		non-parallel methods				parallel method
source	target	VAE [19]	VAEGAN [20]	StarGAN [35]	Proposed	sprocket [61]
SF1	SM1	7.66 ± 0.123	7.70 ± 0.122	7.81 ± 0.126	$\mathbf{6.70 \pm 0.129}$	6.91 ± 0.119
	SF2	7.53 ± 0.118	7.43 ± 0.124	7.54 ± 0.146	$\mathbf{6.57 \pm 0.134}$	6.70 ± 0.125
	SM2	8.06 ± 0.143	8.04 ± 0.145	8.11 ± 0.123	$\mathbf{6.97 \pm 0.124}$	7.06 ± 0.118
SM1	SF1	8.25 ± 0.104	8.20 ± 0.128	8.27 ± 0.119	$\mathbf{7.01 \pm 0.108}$	$\mathbf{7.01 \pm 0.114}$
	SF2	7.43 ± 0.111	7.23 ± 0.117	7.27 ± 0.134	$\mathbf{6.29 \pm 0.133}$	6.30 ± 0.108
	SM2	7.92 ± 0.106	7.82 ± 0.103	7.56 ± 0.106	6.64 ± 0.111	$\mathbf{6.58 \pm 0.099}$
SF2	SF1	7.97 ± 0.127	7.83 ± 0.121	7.99 ± 0.144	$\mathbf{6.94 \pm 0.115}$	7.21 ± 0.111
	SM1	7.38 ± 0.108	7.37 ± 0.097	7.28 ± 0.112	$\mathbf{6.36 \pm 0.108}$	6.77 ± 0.108
	SM2	7.92 ± 0.122	7.78 ± 0.109	7.75 ± 0.124	$\mathbf{6.85 \pm 0.137}$	$\mathbf{6.85 \pm 0.115}$
SM2	SF1	8.33 ± 0.148	8.20 ± 0.158	8.30 ± 0.189	$\mathbf{7.24 \pm 0.151}$	7.31 ± 0.116
	SM1	7.73 ± 0.138	7.66 ± 0.142	7.44 ± 0.122	$\mathbf{6.66 \pm 0.133}$	6.88 ± 0.114
	SF2	7.74 ± 0.135	7.65 ± 0.137	7.53 ± 0.154	$\mathbf{6.64 \pm 0.152}$	6.78 ± 0.146

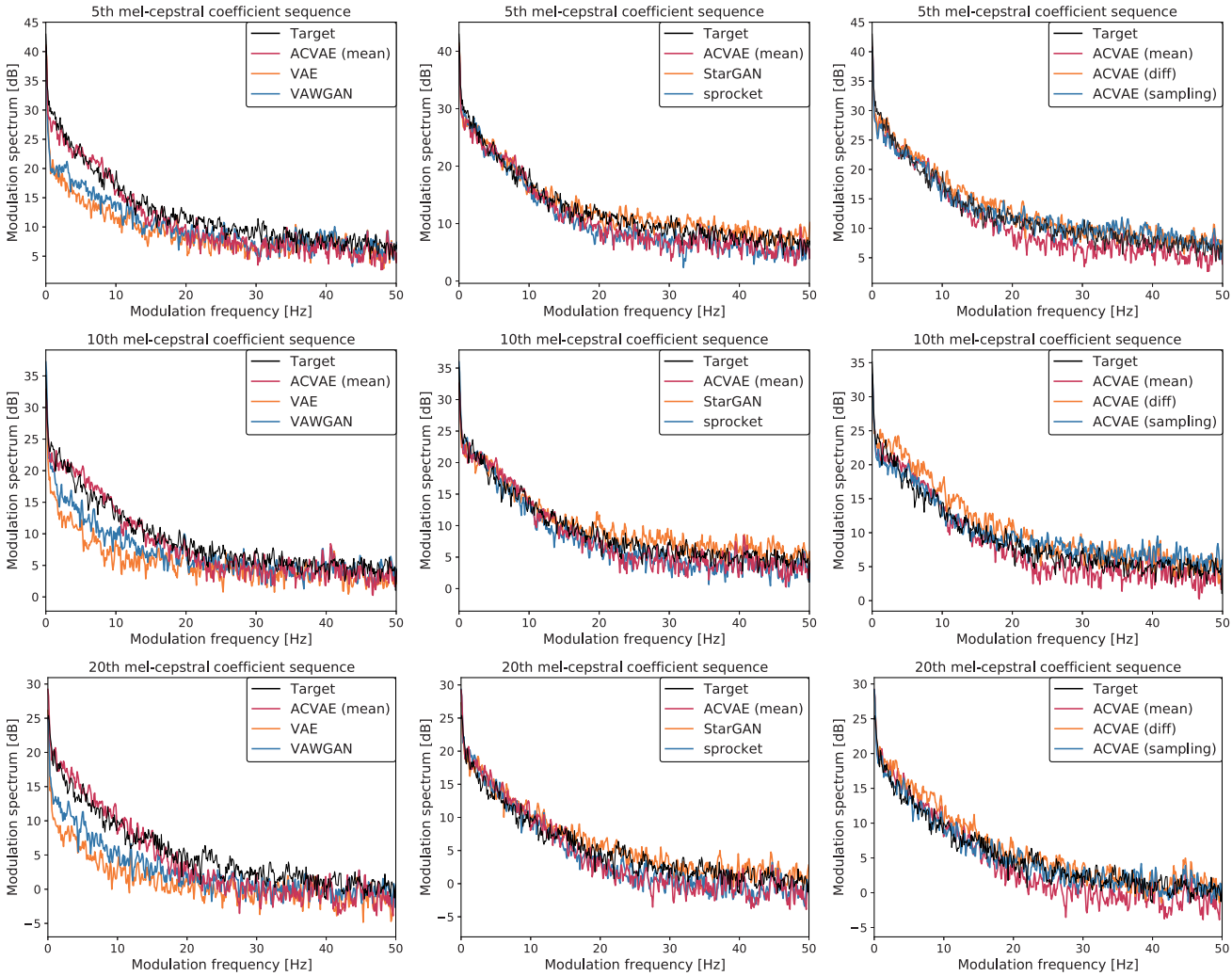


Fig. 7. Average modulation spectra of the 5-th, 10-th and 20-th dimensions of the converted MCC sequences obtained with the baseline methods and the proposed method with and without spectral detail transplantation.

with the proposed and baseline methods. We also conducted ABX tests to compare the similarity to the target speaker of the converted speech samples, where "A" and "B" were converted speech samples obtained with the proposed and baseline methods, respectively, and "X" was a real speech sample of the target speaker. Here, we use the abbreviations ACVAE and ACVAE+ to indicate the proposed method using Eqs. (16) and (18) for the conversion process. With the sound quality test, we included real speech samples and the converted speech samples obtained with the proposed method with and without spectral

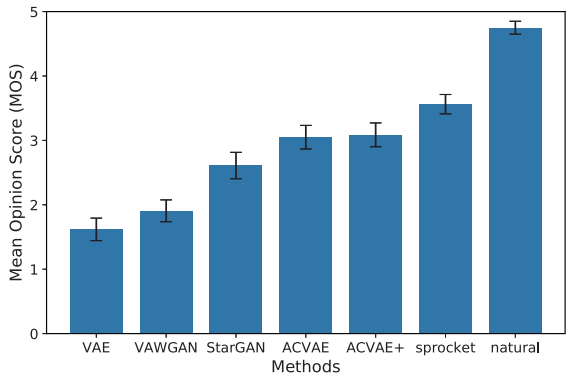


Fig. 8. Results of the MOS test for sound quality.

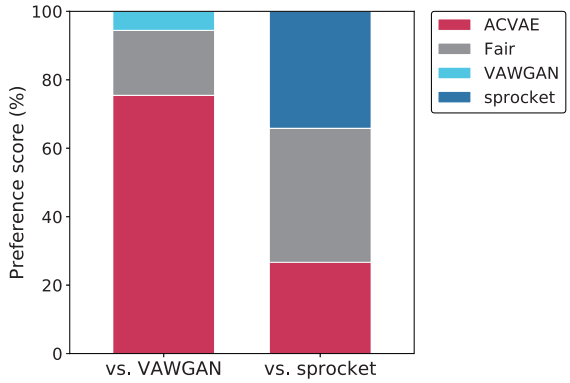


Fig. 9. Results of the ABX test for speaker similarity.

detail transplantation, namely ACVAE and ACVAE+, respectively, and all the baseline methods, namely the VAE-based, VAEGAN-based and StarGAN-based methods and sprocket in the stimuli. With the speaker similarity tests, we chose the VAEGAN-based method and sprocket as the baseline methods. With all these listening tests, speech samples were presented in random orders to eliminate bias as regards the order of the stimuli. Ten listeners participated in our listening tests. For the MOS test of sound quality, each listener was presented 7×10 utterances and asked to evaluate the naturalness by selecting "5: Excellent", "4: Good", "3: Fair", "2: Poor", or "1: Bad" for each utterance. For the ABX tests of speaker similarity, each listener was presented {"A", "B", "X"} $\times$ 24 utterances and asked to listen beyond any audio distortion, concentrate on identifying the voice, and evaluate which of the two was more likely to be produced by the speaker of X by selecting "A", "B" or "fair" for each utterance. The results are shown in Figs. 8 and 9. According to the two-sided Mann-Whitney test performed on the MOS scores for each method pair, the p-values for all the pairs except for the ACVAE and ACVEA+ pair were less than 0.05, indicating that the proposed method significantly outperformed all the baseline non-parallel VC methods (namely, VAE, VAWGAN and StarGAN) in terms of sound quality. It should be noted that the best choice for the architecture design and hyperparameter setting for the StarGAN-based method are currently under investigation, and so the current performance may not do it full justice. We also confirmed that the proposed method could not yield higher sound quality than sprocket. Given the fact that the proposed method outperformed sprocket in terms of the MCD measure, this result may indicate the possibility that the use of different implementations for the vocoding systems including the F_0 estimation modules has caused a difference in the sound quality. Another finding was that the MOS scores obtained with ACVAE and ACVAE+ were comparable to each other, indicating that the effect of spectral detail transplantation was marginal. According to a binomial test with a $1/3$ test proportion performed on the result of the target speaker similarity comparison between the proposed method and VAWGAN, the p-values for the choices of A and B were less than 0.05, indicating that the preference for the proposed method was statistically significant. As for the comparison between the proposed method and sprocket, the p-value for each of the choices was greater than 0.05, indicating that none of the choices were statistically significant. This result is noteworthy considering the fact that sprocket had the advantage of using parallel data for the model training. Since the proposed method is already advantageous in that it can be applied in non-parallel training scenarios, we consider the current result to be promising.

Audio samples generated using the proposed method are provided at [65].

V. Conclusions

This paper proposed a non-parallel VC method using a VAE variant called an auxiliary classifier VAE (ACVAE). The proposed method has two key features. First, we adopted fully convolutional architectures to construct the encoder and decoder networks so that the networks could learn conversion rules that capture time dependencies in the acoustic feature sequences of source and target speech. Second, we proposed using information-theoretic regularization for the model training to ensure that the information in the latent attribute label would not be lost in the generation process. We also presented several ways of converting the feature sequence of input speech using the trained encoder and decoder. Through objective evaluation experiments on a non-parallel speaker identity conversion task, we confirmed the individual effect of each of these ideas and showed that the proposed method obtained smaller MCDs than baseline methods including a parallel VC method. By undertaking subjective evaluation experiments, we showed that the proposed method obtained higher sound quality and speaker similarity than the VAEGAN-based method.

As with the best performing systems [66] in VCC 2018, we are interested in incorporating a neural vocoder in our system in place of the WORLD vocoder to realize further improvements in sound quality.

Note that a preprint version of this work is provided at [1].

References

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

[2] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 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. Audio, Speech, Lang. Process.*, vol. 18, no. 5, pp. 965–973, Jul. 2010.

[7] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 9, pp. 2505–2517, Nov. 2012.

[8] 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.

[9] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, Mar. 1998.

[10] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[11] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 5, pp. 912–921, Jul. 2010.

[12] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1859–1872, Dec. 2014.

[13] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion based on speaker-dependent restricted Boltzmann machines," *IEICE Trans. Inf. Syst.*, vol. 97, no. 6, pp. 1403–1410, 2014.

[14] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 5, pp. 954–964, Jul. 2010.

[15] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014, pp. 19–23.

[16] T. Nakashika, T. Takiguchi, and Y. Ariki, "High-order sequence modeling using speaker-dependent recurrent temporal restricted Boltzmann machines for voice conversion," in *Proc. Interspeech*, 2014, pp. 2278–2282.

[17] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4869–4873.

[18] M. Blaauw and J. Bonada, "Modeling and transforming speech using variational autoencoders," in *Proc. Interspeech*, 2016, pp. 1770–1774.

[19] 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. Asia–Pac. Signal Inf. Process. Assoc.*, 2016.

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

[21] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion using sparse representation in noisy environments," *IEICE Trans. Inf. Syst.*, vol. E96-A, no. 10, pp. 1946–1953, 2013.

[22] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 10, pp. 1506–1521, Oct. 2014.

[23] F.-L. Xie, F. K. Soong, and H. Li, "A KL divergence and DNN-based approach to voice conversion without parallel training sentences," in *Proc. Interspeech*, 2016, pp. 287–291.

[24] T. Kinnunen, L. Juvela, P. Alku, and J. Yamagishi, "Non-parallel voice conversion using i-vector PLDA: Towards unifying speaker verification and transformation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 5535–5539.

[25] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[26] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014.

[27] D. P. Kingma, D. J. Rezendey, S. Mohamedy, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[28] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5274–5278.

[29] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[30] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1558–1566.

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

[32] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[33] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1857–1865.

[34] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2849–2857.

[35] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks," in *Proc. IEEE Spoken Lang. Technol. Workshop*, Jun. 2018, pp. 266–273.

[36] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," Nov. 2017, arXiv:1711.09020.

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

[38] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[39] D. Barber and F. V. Agakov, "Information maximization in noisy channels: A variational approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 201–208.

[40] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 2642–2651.

[41] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," Sep. 2016, arXiv:1609.03499.

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

[43] N. Kalchbrenner *et al.*, "Efficient neural audio synthesis," Feb. 2018, arXiv:1802.08435.

[44] S. Mehri *et al.*, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[45] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "FFTNet: A real-time speaker-dependent neural vocoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 2251–2255.

[46] A. van den Oord *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3918–3926.

[47] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," Feb. 2019, arXiv:1807.07281.

[48] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 3617–3621.

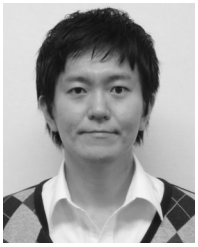
[49] S. Kim, S. Lee, J. Song, and S. Yoon, "FloWaveNet: A generative flow for raw audio," Nov. 2018, arXiv:1811.02155.

[50] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5916–5920.

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

[52] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1992, pp. 137–140.

[53] 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.
[54] F. Villavicencio, J. Bonada, and Y. Hisaminato, "Observation-model error compensation for enhanced spectral envelope transformation in voice conversion," in *Proc. Int. Workshop Mach. Learn. Signal Process.*, 2015.
[55] J. Lorenzo-Trueba *et al.*, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," in *Proc. Speaker Lang. Recognit. Workshop*, 2018, pp. 195–202.
[56] 2016. [Online]. Available: https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder
[57] 2015. [Online]. Available: https://github.com/r9y9/pysptk
[58] K. Liu, J. Zhang, and Y. Yan, "High quality voice conversion through phoneme-based linear mapping functions with STRAIGHT for Mandarin," in *Proc. Int. Conf. Fuzzy Syst. Knowl. Discovery*, 2007, pp. 410–414.
[59] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[60] K. Kobayashi and T. Toda, "sprocket: Open-source voice conversion software," in *Proc. Odyssey, Speaker Lang. Recognit. Workshop*, 2018, pp. 203–210.
[61] 2017. [Online]. Available: https://github.com/JeremyCCHsu/vae-npvc. Accessed on: Jan. 25, 2019.
[62] 2017. [Online]. Available: https://github.com/JeremyCCHsu/vae-npvc/tree/vawgan. Accessed on: Jan. 25, 2019.
[63] 2017. [Online]. Available: https://github.com/k2kobayashi/sprocket. Accessed on: Jan. 28, 2019.
[64] S. Takamichi, T. Toda, A. W. Black, G. Neubig, S. Sakti, and S. Nakamura, "Post-filters to modify the modulation spectrum for statistical parametric speech synthesis," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 4, pp. 755–767, Apr. 2016.
[65] 2016. [Online]. Available: http://www.kecl.ntt.co.jp/people/kameoka.hirokazu/Demos/acvae-vc/
[66] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "WaveNet vocoder with limited training data for voice conversion," in *Proc. Interspeech*, 2018, pp. 1983–1987.



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

Takuhiro Kaneko received the B.E. and M.S. degrees in 2012 and 2014, respectively, from the University of Tokyo, Tokyo, Japan, where he has been working toward the Ph.D. degree since 2017. He is currently a Research Scientist with the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation, Atsugi, Japan. 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 was the recipient of the ICPR2012 Best Student Paper Award at the 21st International Conference on Pattern Recognition in 2012.

Kou Tanaka received the B.E. degree from Kobe University, Kobe, Japan, in 2012, and the M.E. and D.E. degrees from Nara Institute of Science and Technology, Ikoma, Japan, in 2014 and 2017, respectively. From 2015 to 2017, he was a Research Fellow with the Japan Society for the Promotion of Science. He is currently a Research Scientist with the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation, Atsugi, Japan. His current research focuses on speech signal processing with a strong focus on deep generative models.

Nobukatsu Hojo received the B.E and M.E. degrees from the University of Tokyo, Tokyo, Japan, in 2012 and 2014, respectively. In 2014, he joined the NTT Media Intelligence Laboratories, Nippon Telegraph and Telephone Corporation, Atsugi, Japan, where he engaged in the research and development of speech synthesis. He is currently a Research Scientist with the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation. He is a member of the Acoustical Society of Japan, the International Speech Communication Association, and the Institute of Electronics, Information and Communication Engineers of Japan.