IEICE TRANS. INF. & SYST., VOL.E103–D, NO.9 SEPTEMBER 2020

PAPER

Joint Adversarial Training of Speech Recognition and Synthesis Models for Many-to-One Voice Conversion Using Phonetic Posteriorgrams

Yuki SAITO[†,††a)], ***Student Member***, **Kei AKUZAWA**[†,††], ***and*** **Kentaro TACHIBANA**[†b)], ***Nonmembers***

SUMMARY This paper presents a method for many-to-one voice conversion using phonetic posteriorgrams (PPGs) based on an adversarial training of deep neural networks (DNNs). A conventional method for many-to-one VC can learn a mapping function from input acoustic features to target acoustic features through separately trained DNN-based speech recognition and synthesis models. However, 1) the differences among speakers observed in PPGs and 2) an over-smoothing effect of generated acoustic features degrade the converted speech quality. Our method performs a domain-adversarial training of the recognition model for reducing the PPG differences. In addition, it incorporates a generative adversarial network into the training of the synthesis model for alleviating the over-smoothing effect. Unlike the conventional method, ours jointly trains the recognition and synthesis models so that they are optimized for many-to-one VC. Experimental evaluation demonstrates that the proposed method significantly improves the converted speech quality compared with conventional VC methods.

key words: *many-to-one voice conversion, phonetic posteriorgrams, deep neural networks, over-smoothing, domain-adversarial training, generative adversarial networks*

1. Introduction

Voice conversion (VC) [1] is a technique for transforming the characteristics of input speech to those of target speech while keeping its phonetic content. We can use VC for various applications such as speaking aids [2] and entertainment such as singing VC [3].

In statistical VC [4], we train a VC model that converts acoustic features of an input speaker into those of a target speaker. Specifically, parallel VC constructs a VC model by using parallel speech corpora (i.e., pairs of the same utterances spoken by input and target speakers), which achieves high-quality VC by learning the mapping function from input to target acoustic features in frame-wise [5]–[7] or sequence-wise [8], [9] manners. However, recording parallel speech corpora requires significant time and cost, and it is difficult to extend the VC to more practical ones such as many-to-one VC and many-to-many VC. On the other hand, non-parallel VC trains a VC model without using parallel speech corpora, which enables us to easily increase the amount of training data and to realize the more practical VC frameworks. Although the converted speech quality of non-parallel VC is still limited, recent development of VC techniques based on deep neural networks (DNNs) has made the quality closer to that of parallel VC. Restricted Boltzmann machines [10] and variational autoencoders [11], [12] are well-known examples of VC models that can be used for non-parallel VC. The use of DNNs also offers a way to incorporate other speech processing techniques such as text-to-speech synthesis [13], automatic speech recognition [14], and speaker verification [15] into training of a VC model.

As an example of a VC method that utilizes knowledge for other DNN-based speech processing, Sun et al. [16] proposed a method that utilizes pre-trained speech recognition and synthesis models for building a many-to-one VC model without using parallel speech corpora. In their method, phonetic posteriorgrams (PPGs) predicted by a DNN-based speech recognition model act as intermediate representations for learning the many-to-one mapping from input to target speech. Their method takes several steps to build the VC model with different two corpora. Firstly, the recognition model is trained to predict the phonetic content of input speech by using a speech corpus including many speakers. Secondly, a DNN-based speech synthesis model is trained to generate acoustic features from PPGs with a speech corpus including only one target speaker. Finally, the two models are concatenated for building the many-to-one model. This paper follows the PPG-based many-to-one VC method since it has a potential for realizing high-quality many-to-one VC applications without requiring pre-recording of input speakers' voices whose quality was not always sufficient to adapt or fine-tune the pre-trained VC model.

Although the conventional PPG-based VC can easily construct the many-to-one VC model, its *separate* training of the recognition and synthesis models cannot deal with the differences among speakers such as speaking style and recording conditions. In practice, there is a domain mismatch between the two training data for the recognition and synthesis models. The one for the recognition model training includes a large amount of speakers whose speaking style and recording conditions are not necessarily well-controlled, while the other consists of many utterances spoken by a single speaker with studio-recording quality. As

Manuscript received November 8, 2019.
Manuscript revised March 27, 2020.
Manuscript publicized June 12, 2020.

[†]The authors are with DeNA Co., Ltd., Tokyo, 150–8510 Japan.
[††]The authors are with The University of Tokyo, Tokyo, 113–8656 Japan.
a) E-mail: yuuki_saito@ipc.i.u-tokyo.ac.jp
b) E-mail: kentaro.tachibana@dena.com

DOI: 10.1587/transinf.2019EDP7297

reported in [17], the differences among speakers can affect PPGs and significantly degrade the converted speech quality. Also, the statistical differences between the target and generated acoustic features such as an over-smoothing effect [5], [18] considerably deteriorate the quality.

To deal with the issues in the conventional DNN-based many-to-one VC using PPGs, this paper proposes a joint adversarial training for the recognition and synthesis models. The method introduces two DNN-based discriminative models to the training. One is a domain classification model in a domain-adversarial training (DAT) [19] that distinguishes the target speaker from others by using their latent variables extracted from the recognition model. The loss function for training the recognition model is the weighted sum of the softmax cross-entropy for the phoneme prediction and the loss for fooling the domain classifier that makes PPGs more invariant to input speakers. The other is a speaker verification model that discriminates between the target and generated acoustic features. The loss function for training the synthesis model is the weighted sum of the mean squared error for the acoustic feature generation and the loss for deceiving the speaker verification that alleviates the over-smoothing effect as in the same manner as a training method for speech synthesis [20] based on generative adversarial network (GANs) [21]. Unlike the conventional method for many-to-one VC, ours *jointly* trains the recognition and synthesis models with a unified framework so that they are optimized for generating the target acoustic features from PPGs predicted from input acoustic features. As a result of the joint training, the proposed algorithm can take advantage of both the DAT for reducing the speaker differences (i.e., improving the speaker similarity) and the GAN for enhancing the naturalness of the converted speech. Experimental evaluation demonstrates that our method significantly improves the converted speech quality compared with the conventional ones.

The rest of this paper is organized as follows. Section 2 describes the conventional algorithm for many-to-one VC. Section 3 explains the proposed joint adversarial training for many-to-one VC. Section 4 presents experimental evaluation. Section 5 concludes this paper with a summary.

2. Conventional Many-to-One VC Method

In the conventional method [16], DNN-based speech recognition and synthesis models are trained to represent mapping from any arbitrary input speech to the target speech with using PPGs as the intermediate representation of the VC process.

2.1 Training of the Speech Recognition Model

The recognition model $R(\cdot)$ is trained to predict a phoneme label sequence $\boldsymbol{l}$ from an input acoustic feature sequence $\boldsymbol{x}$ such as mel-frequency cepstral coefficients (MFCCs). A pair of the phoneme label and input acoustic feature is sampled from a multi-speaker corpus $\mathcal{D}^{(\mathrm{M})} = \{(\boldsymbol{x}_n^{(\mathrm{M})}, \boldsymbol{l}_n^{(\mathrm{M})})\}_{n=1}^{N^{(\mathrm{M})}}$, where $N^{(\mathrm{M})}$ denotes the amount of training data for the recognition model. A PPG sequence $\hat{\boldsymbol{p}}^{(\mathrm{M})} = R(\boldsymbol{x}^{(\mathrm{M})})$ that represents a sequence of posterior probabilities of the phoneme label $\boldsymbol{l}^{(\mathrm{M})}$ given input acoustic feature $\boldsymbol{x}^{(\mathrm{M})}$ is predicted by the recognition model. $R(\cdot)$ is trained to minimize the phoneme prediction loss defined as the softmax cross-entropy (SCE) between the phoneme label and PPG, i.e., $L_{\mathrm{SCE}}(\boldsymbol{l}^{(\mathrm{M})}, \hat{\boldsymbol{p}}^{(\mathrm{M})})$.

2.2 Training of the Speech Synthesis Model

Assuming that $R(\cdot)$ is the speaker-independent speech recognition model, the speaker-dependent speech synthesis

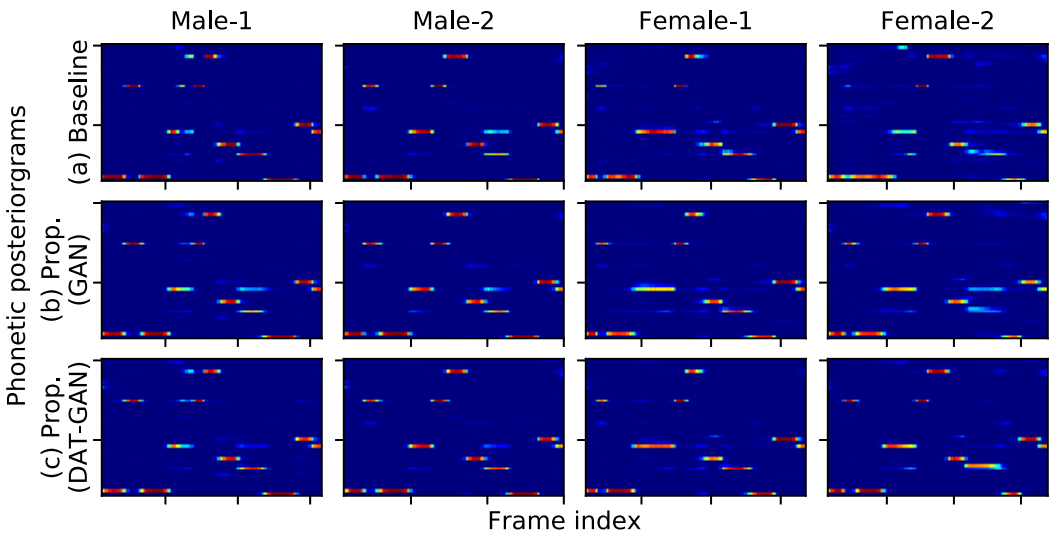

Fig. 1 Examples of PPGs predicted by input acoustic features of four different speakers who uttered the same sentences used in subjective evaluation described in Sect. 4.2. The horizontal axes represent the temporal axis, and the vertical axes represents the phoneme index. Brighter values denote high posterior probabilities. We modified the ranges of the temporal axes in these figures for clear illustration.

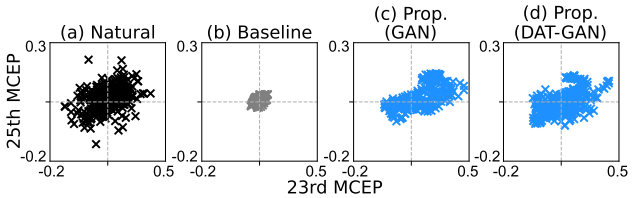

Fig. 2 Scatter plots of (a) natural MCEPs and (b)–(d) generated MCEPs which are extracted from one utterance of the female target speaker not used for training synthesis model.

model $G(\cdot)$ is trained to generate a target acoustic feature sequence $\boldsymbol{y}$ such as mel-cepstral coefficients (MCEPs) from a PPG sequence. A pair of the input and target acoustic features is sampled from a target speaker corpus $\mathcal{D}^{(\mathrm{O})} = \{(\boldsymbol{x}_n^{(\mathrm{O})}, \boldsymbol{y}_n^{(\mathrm{O})})\}_{n=1}^{N^{(\mathrm{O})}}$, where $N^{(\mathrm{O})}$ denotes the amount of training data for the synthesis model. A generated acoustic feature sequence $\hat{\boldsymbol{y}}^{(\mathrm{O})}$ is obtained through the recognition and synthesis models, i.e., $\hat{\boldsymbol{y}}^{(\mathrm{O})} = G(R(\boldsymbol{x}^{(\mathrm{O})}))$. $G(\cdot)$ is trained to minimize the acoustic feature generation loss defined as the mean squared error (MSE) between the target and generated acoustic feature sequences, i.e., $L_{\mathrm{MSE}}(\boldsymbol{y}^{(\mathrm{O})}, \hat{\boldsymbol{y}}^{(\mathrm{O})})$. Note that model parameters of $R(\cdot)$ (e.g., weights and bias parameters) are not updated by this training.

2.3 Problems

The conventional method can train the many-to-one VC model without using parallel speech corpora. However, as shown in Fig. 1 (a), actual PPGs fed into the synthesis model in the VC process can be different among input speakers, since training of the recognition model based on the phoneme prediction loss does not guarantee to learn speaker-invariant PPGs. The PPG differences can degrade the converted speech quality because the synthesis model trained on only PPGs of the target speaker does not necessarily generalize to input PPGs of the source speakers. Moreover, as shown in Fig. 2 (b), generated acoustic features of the algorithm tend to be over-smoothed, which considerably deteriorates the quality.

3. Proposed Many-to-One VC Method

3.1 Joint Adversarial Training of the Speech Recognition and Synthesis Models

3.1.1 DAT of the Speech Recognition Model for Many-to-One VC

The DAT [19] is a general framework to train DNN-based recognition models that are more robust towards variation in their input features by learning domain-invariant latent variables, and has been applied to accented speech recognition [22] and speaker recognition [23]. Although the DAT was originally invented for improving recognition accuracy, Chou et al. [24] demonstrated its efficacy in autoencoder-based VC that learns speaker-independent latent variables. Note that their method does not guarantee that the latent variables represent phonetic content of input speech unlike many-to-one VC using PPGs. For improving the converted speech quality of many-to-one VC, our goal is to obtain PPGs that are invariant to variation in input speakers. In this paper, we regard the two speech corpora used for the training of the recognition and synthesis models as the domains, i.e., there are 1) the multi-speaker domain $\mathcal{D}^{(\mathrm{M})}$ and 2) the target speaker domain $\mathcal{D}^{(\mathrm{O})}$, and minimize the difference between the two domains by the DAT.

For clear formulation, we split the recognition model $R(\cdot)$ into two sub-models, $R(\cdot) = R_{\mathrm{p}}(R_{\mathrm{f}}(\cdot))$, where $R_{\mathrm{f}}(\cdot)$ is a feature extraction model that extracts latent variables $\hat{\boldsymbol{f}}$ representing phonetic content of input speech as $\hat{\boldsymbol{f}} = R_{\mathrm{f}}(\boldsymbol{x})$, and $R_{\mathrm{p}}(\cdot)$ is a phoneme prediction model that predicts a PPG sequence from the latent variables, i.e., $\hat{\boldsymbol{p}} = R_{\mathrm{p}}(\hat{\boldsymbol{f}}) = R_{\mathrm{p}}(R_{\mathrm{f}}(\boldsymbol{x}))$. To capture the domain difference, we introduce a domain classification model $D_{\mathrm{dc}}(\cdot)$ that uses the latent variables to identify the domains as the training of $R(\cdot)$. $D_{\mathrm{dc}}(\cdot)$ is trained to minimize the loss function defined as:

$$L_{\mathrm{dc}}\left(\hat{\boldsymbol{f}}^{(\mathrm{M})}, \hat{\boldsymbol{f}}^{(\mathrm{O})}\right) = -\log D_{\mathrm{dc}}\left(\hat{\boldsymbol{f}}^{(\mathrm{O})}\right) - \log\left(1 - D_{\mathrm{dc}}\left(\hat{\boldsymbol{f}}^{(\mathrm{M})}\right)\right), \tag{1}$$

where $\hat{\boldsymbol{f}}^{(\mathrm{M})}$ and $\hat{\boldsymbol{f}}^{(\mathrm{O})}$ are extracted from $\boldsymbol{x}^{(\mathrm{M})}$ and $\boldsymbol{x}^{(\mathrm{O})}$, respectively. On the other hand, the recognition model $R(\cdot)$ is trained to minimize the loss defined as follows:

$$L_{\mathrm{R}}\left(\boldsymbol{l}^{(\mathrm{M})}, \hat{\boldsymbol{p}}^{(\mathrm{M})}, \hat{\boldsymbol{f}}^{(\mathrm{M})}, \hat{\boldsymbol{f}}^{(\mathrm{O})}\right) = L_{\mathrm{SCE}}\left(\boldsymbol{l}^{(\mathrm{M})}, \hat{\boldsymbol{p}}^{(\mathrm{M})}\right) - \omega_{\mathrm{R}} L_{\mathrm{dc}}\left(\hat{\boldsymbol{f}}^{(\mathrm{M})}, \hat{\boldsymbol{f}}^{(\mathrm{O})}\right), \tag{2}$$

where ω_{R} is a hyperparameter that controls the effect of the second term. The loss function can be regarded as the weighted sum of the phoneme prediction loss and the loss to make $D_{\mathrm{dc}}(\cdot)$ misclassify the domains by modifying the latent variables. Therefore, the minimization of Eq. (2) can be expected to reduce the differences among input speakers observed in their PPGs.

3.1.2 GAN-Based Training of the Speech Synthesis Models

The GAN-based training algorithm for statistical parametric speech synthesis [20] was proposed for dealing with the over-smoothing effect, which uses a speaker verification model $D_{\mathrm{sv}}(\cdot)$ that distinguishes natural acoustic features $\boldsymbol{y}$ from synthetic ones $\hat{\boldsymbol{y}}$. Since an objective of the GAN is matching distributions of natural and synthetic data, the algorithm makes the distribution of $\hat{\boldsymbol{y}}$ close to that of $\boldsymbol{y}$. We introduce this algorithm to the training of the speech synthesis model in many-to-one VC for improving the converted speech quality.

Referring to [20], we adopt a Wasserstein GAN [25]-based discriminator for the speaker verification model

$D_{\mathrm{sv}}(\cdot)$, which is trained to minimize the loss function defined as follows:

$$L_{\mathrm{sv}}\left(\boldsymbol{y}^{(\mathrm{O})}, \hat{\boldsymbol{y}}^{(\mathrm{O})}\right) = -D_{\mathrm{sv}}\left(\boldsymbol{y}^{(\mathrm{O})}\right) + D_{\mathrm{sv}}\left(\hat{\boldsymbol{y}}^{(\mathrm{O})}\right). \quad (3)$$

After updating $D_{\mathrm{sv}}(\cdot)$, its model parameters are clamped to a fixed interval such as $[-0.01, 0.01]$ for satisfying the K-Lipschitz constraint of the discriminative model. Minimizing Eq. (3) approximates the earth mover's distance between the distributions of $\boldsymbol{y}^{(\mathrm{O})}$ and $\hat{\boldsymbol{y}}^{(\mathrm{O})}$, and the synthesis model $G(\cdot)$ tries to reduce the distance by minimizing the loss function defined as follows:

$$L_{\mathrm{G}}\left(\boldsymbol{y}^{(\mathrm{O})}, \hat{\boldsymbol{y}}^{(\mathrm{O})}\right) = L_{\mathrm{MSE}}\left(\boldsymbol{y}^{(\mathrm{O})}, \hat{\boldsymbol{y}}^{(\mathrm{O})}\right) + \omega_{\mathrm{G}} L_{\mathrm{adv}}\left(\hat{\boldsymbol{y}}^{(\mathrm{O})}\right), \quad (4)$$

where $L_{\mathrm{adv}}(\hat{\boldsymbol{y}}^{(\mathrm{O})}) = -D_{\mathrm{sv}}(\hat{\boldsymbol{y}}^{(\mathrm{O})})$ is the adversarial loss to fool $D_{\mathrm{sv}}(\cdot)$ and ω_{G} is a hyperparameter that controls its weight. The minimization of Eq. (4) can be expected to overcome the over-smoothing effect of generated acoustic features in many-to-one VC.

3.1.3 Joint Training of the Speech Recognition and Synthesis Models

To optimize both the recognition and synthesis models for many-to-one VC, we *jointly* train the two models with a unified framework. First, we update the two discriminative models $D_{\mathrm{dc}}(\cdot)$ and $D_{\mathrm{sv}}(\cdot)$ by minimizing Eqs. (1) and (3), respectively. Then, we jointly update both $R(\cdot)$ and $G(\cdot)$ by minimizing the sum of Eqs. (2) and (4). Since the loss for training $G(\cdot)$ is also used for training $R(\cdot)$, the recognition model can be expected to predict PPGs that are speaker-invariant and can accurately generate target acoustic features. Figure 3 illustrates a schematic diagram of the proposed joint adversarial training.

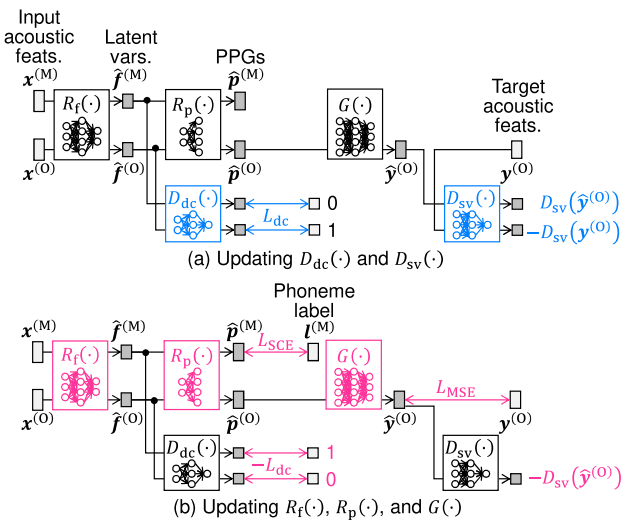

Fig. 3 Schematic diagram of proposed training algorithm. We firstly update the two discriminative models $D_{\mathrm{dc}}(\cdot)$ and $D_{\mathrm{sv}}(\cdot)$. Then, we update the recognition model $R(\cdot) = R_{\mathrm{p}}(R_{\mathrm{f}}(\cdot))$ and synthesis model $G(\cdot)$ by utilizing the updated discriminative models. These updates are iterated during the training, and the final VC model is constructed by concatenating the recognition and synthesis models.

3.2 Discussion

As shown in Figs. 2 (c) and 2 (d), our GAN-based training algorithm alleviates the over-smoothing effect of generated acoustic features. However, only using the GAN cannot necessarily reduce the PPG differences as shown in Fig. 1 (b). On the other hand, our algorithm using both the DAT and GAN successfully reduces the PPG differences as shown in Fig. 1 (c), which can be expected to improve the converted speech quality.

In the GAN-based training, we can also reduce the distance between the distributions of the target acoustic features $\boldsymbol{y}^{(\mathrm{O})}$ and ones predicted by other speakers, i.e., $\hat{\boldsymbol{y}}^{(\mathrm{M})} = G(R(\boldsymbol{x}^{(\mathrm{M})}))$. This can be done by approximating the distance in the training of $D_{\mathrm{sv}}(\cdot)$ and minimizing the distance in the training of $G(\cdot)$. However, we found that the formulation considerably degraded the converted speech quality from our preliminary experiment. It is assumed that the differences among the target and other speakers in the acoustic feature domain might be larger than those in the latent variable domain, and minimizing the former by using the GAN becomes more difficult.

Regarding related work, there are several methods that incorporate the GAN into training of a VC model. CycleGAN-VC [26], [27] trains a VC model using non-parallel speech corpora based on the adversarial training considering cyclic-consistency [28]. StarGAN-VC [29], [30] extends this VC to many-to-many VC by introducing the StarGAN [31] into training of a VC model. Although the GAN-based VC techniques can train the non-parallel VC models without using any text transcriptions, they cannot guarantee the quality of converted speech when the input speaker is not included in the training data. On the other hand, our method can take any arbitrary speakers as an input speaker of the VC, although it requires a large speech corpus with text transcriptions and limits the target speaker to a specific one. We believe that semi-supervised training of the recognition and synthesis models (e.g., machine speech chain [32]) and conditional GAN [33] using one-hot speaker codes [34] can alleviate these limitations. Also, we can apply the proposed training algorithm to more realistic and practical VC frameworks based on PPGs such as cross-lingual VC [35], one-shot VC [36], and WaveNet [37]-based VC [38], [39].

4. Experimental Evaluation

4.1 Experimental Conditions

We considered two many-to-one VC tasks in this evaluation and compared the proposed method with conventional ones. We used two professional speakers, i.e., one female voice actress taken from the NICT Voice Actress Dialogue Corpus [40] and one male voice actor included in an internal dataset of DeNA, as the target speakers for the VC

tasks. For training the speech recognition model $R(\cdot)$, we used the Spontaneous Speech Corpus of Japanese (CSJ) [41] that included 1,417 amateur speakers (470 females and 947 males) with various speaking styles such as a monologue, dialogue, and reading aloud. We used about 99% of the corpus as the multi-speaker corpus $\mathcal{D}^{(\mathrm{M})}$. The remainder of the CSJ corpus was utilized for objective evaluations described in Sects. 4.2.2 and 4.2.3. For training the speech synthesis model $G(\cdot)$, we used 5,174 utterances of the female target speaker (FT) or 2,211 utterances of the male target speaker (MT) as the target speaker corpus $\mathcal{D}^{(\mathrm{O})}$. The other 50 utterances of the target speakers were used for objective evaluations described in Sects. 4.2.1 and 4.2.3. Note that $\mathcal{D}^{(\mathrm{M})}$ and $\mathcal{D}^{(\mathrm{O})}$ were significantly different in many aspects such as recording environments (somewhat noisy or definitely clean), speaking styles, and speaking skills (amateur or professional). All speech samples were downsampled at 16 kHz. The WORLD vocoder [42] (D4C edition [43]) was utilized to extract log F0, 40-dimensional MCEPs, and band aperiodicity. In many-to-one VC, the 1st-through-39th MCEPs of the target speaker were predicted by DNNs. The F0 values were extracted by integrating results of multiple F0 extractors [42], [44], [45]. The log F0 was linearly converted. Band aperiodicity and the 0th MCEP were not converted.

All DNN architectures were 1D convolutional neural networks along time axis [46] with a fixed sequence length of 128 frames. The feature extraction model $R_{\mathrm{f}}(\cdot)$ extracted 256-dimensional latent variables from 13-dimensional MFCCs and their dynamic features. The phoneme prediction model $R_{\mathrm{p}}(\cdot)$ predicted 43-dimensional Japanese PPGs by using the latent variables. The synthesis model $G(\cdot)$ generated the 1st-through-39th MCEPs of the target speaker from the PPGs. MFCCs and MCEPs were normalized to have zero mean and unit variance. The domain classification model $D_{\mathrm{dc}}(\cdot)$ distinguished $\mathcal{D}^{(\mathrm{O})}$ from $\mathcal{D}^{(\mathrm{M})}$ by using the latent variables. The speaker verification model $D_{\mathrm{sv}}(\cdot)$ discriminated natural MCEPs from generated ones. The activation function for hidden layers was a leaky rectified linear unit [47]. To avoid overfitting, dropout [48] was applied to all hidden layers. To accelerate the DNN training, batch normalization [49] was applied to some hidden layers in the synthesis model. Table 1 shows details of the DNN architectures. In this table, "Conv1D(C_{in}, C_{out}, k, s)" and "Deconv1D(C_{in}, C_{out}, k, s)" denote 1D convolutional and deconvolutional layers, respectively. C_{in} and C_{out} mean the number of channels of input and output, respectively. k and s denote the convolution window size and stride width, respectively.

As an initial setting, we constructed the recognition model $R(\cdot) = R_{\mathrm{p}}(R_{\mathrm{f}}(\cdot))$ with the multi-speaker corpus $\mathcal{D}^{(\mathrm{M})}$. The initialization was performed with 1 epoch by using all utterances in $\mathcal{D}^{(\mathrm{M})}$. The optimizer used for the initialization was AdaGrad [50], with its learning rate set to 0.01. The frame-wise phoneme prediction accuracy of the initialized recognition model calculated with the evaluation data of the CSJ corpus was 80.4%. By using the model, we constructed

Table 1 DNN architectures used in experimental evaluation. In this table, "Conv1D(C_{in}, C_{out}, k, s)" and "Deconv1D(C_{in}, C_{out}, k, s)" denote 1D convolution and deconvolution layers, respectively. C_{in} and C_{out} mean the number of channels of input and output, respectively. k and s denote the convolution window size and stride width, respectively

Recognition $R(\cdot) = R_{\mathrm{p}}(R_{\mathrm{f}}(\cdot))$	**Synthesis** $G(\cdot)$
Feature extraction $R_{\mathrm{f}}(\cdot)$	Conv1D(43, 256, 15, 1)
Conv1D(26, 256, 15, 1)	Conv1D(256, 512, 5, 2)
Conv1D(256, 512, 5, 2)	Conv1D(512, 1024, 5, 2)
Conv1D(512, 1024, 5, 2)	Deconv1D(1024, 512, 5, 2)
Deconv1D(1024, 512, 5, 2)	Deconv1D(512, 256, 5, 2)
Deconv1D(512, 256, 5, 2)	Conv1D(256, 39, 15, 1)
Phoneme prediction $R_{\mathrm{p}}(\cdot)$	
Conv1D(256, 43, 15, 1)	
Domain classification $D_{\mathrm{dc}}(\cdot)$	**Speaker verification** $D_{\mathrm{sv}}(\cdot)$
Conv1D(256, 512, 1, 1)	Conv1D(39, 512, 1, 1)
Conv1D(512, 512, 5, 1)	Conv1D(512, 512, 5, 1)
Conv1D(512, 512, 5, 1)	Conv1D(512, 512, 5, 1)
Conv1D(512, 1, 1, 1)	Conv1D(512, 1, 1, 1)

five many-to-one VC models with the following algorithms:

Baseline: Training $G(\cdot)$ with the fixed $R(\cdot)$ [16]
Prop. (Joint): Jointly training $R(\cdot)$ and $G(\cdot)$ with the hyperparameter settings $\omega_{\mathrm{R}} = 0$ and $\omega_{\mathrm{G}} = 0$
Prop. (DAT): Jointly training $R(\cdot)$ and $G(\cdot)$ with the hyperparameter settings $\omega_{\mathrm{R}} = 0.25$ and $\omega_{\mathrm{G}} = 0.0$
Prop. (GAN): Jointly training $R(\cdot)$ and $G(\cdot)$ with the hyperparameter settings $\omega_{\mathrm{R}} = 0$ and $\omega_{\mathrm{G}} = 0.5$
Prop. (DAT-GAN): Jointly training $R(\cdot)$ and $G(\cdot)$ with the hyperparameter settings $\omega_{\mathrm{R}} = 0.25$ and $\omega_{\mathrm{G}} = 0.5$

Here, the hyperparameters $(\omega_{\mathrm{R}}, \omega_{\mathrm{G}})$ were empirically chosen. All of the five algorithms were performed with 5 epochs by using all utterances in the corpus $\mathcal{D}^{(\mathrm{O})}$. In the training of "Prop. (*)," a pair of labeled training data $(\boldsymbol{x}^{(\mathrm{M})}, \boldsymbol{l}^{(\mathrm{M})})$ was randomly sampled from $\mathcal{D}^{(\mathrm{M})}$. The optimizers used for training all DNNs, i.e., $R(\cdot)$, $G(\cdot)$, $D_{\mathrm{dc}}(\cdot)$, and $D_{\mathrm{sv}}(\cdot)$, were AdaGrad, with their learning rates set to 0.01.

4.2 Objective Evaluations

4.2.1 Logarithmic Global Variance Distance

Since we focused on *non-parallel* many-to-one VC in this paper, we could not calculate any objective evaluation metrics that require parallel speech utterances of source and target speakers. Instead, we used global variances (GVs) [5] of natural and generated speech of the target speakers (MT or FT), which is defined as the second moment of the speech parameter sequence and quantifies the degree of the oversmoothing effect of MCEPs predicted by the VC model. Here, we calculated logarithmic GV distance (LogGVD) defined as follows:

$$\mathrm{LogGVD} = \frac{1}{M} \| \log \hat{\boldsymbol{v}} - \log \boldsymbol{v} \|_2^2, \tag{5}$$

where $\boldsymbol{v}$ and $\hat{\boldsymbol{v}}$ denote GV vectors of natural and generated MCEPs, respectively. M corresponds to the order of MCEPs, and we set it to 39 in this evaluation. This evaluation result would correspond to the ideal performance of

Table 2 LogGVDs between natural and generated speech with their standard deviations

	Target speaker	
	MT	FT
Baseline	1.96 ± 0.34	5.05 ± 0.71
Prop. (Joint)	1.98 ± 0.38	3.93 ± 0.78
Prop. (DAT)	1.44 ± 0.26	3.80 ± 0.67
Prop. (GAN)	0.44 ± 0.18	0.21 ± 0.11
Prop. (DAT-GAN)	**0.23 ± 0.10**	**0.11 ± 0.06**

Table 3 Frame-wise phoneme recognition accuracy of speech recognition models [%]

	Target speaker	
	MT	FT
Baseline	**80.4**	**80.4**
Prop. (Joint)	63.5	77.6
Prop. (DAT)	62.3	77.5
Prop. (GAN)	62.8	77.5
Prop. (DAT-GAN)	62.4	77.0

the PPG-based many-to-one VC methods because there was no domain mismatch between input features fed into the VC model in the training and inference. We used the evaluation data (50 utterances of MT or FT) to calculate the LogGVDs.

Table 2 lists the averaged LogGVDs and their standard deviations. From the results, we found that "Prop. (GAN)" significantly reduced the LogGVDs better than "Baseline," which demonstrated that the GAN-based algorithm for many-to-one VC using PPGs was effective in minimizing the distributional differences between natural and generated MCEPs. Moreover, we observed that "Prop. (DAT)" also decreased the LogGVDs, and the combination of the DAT and GAN, i.e., "Prop. (DAT-GAN)," achieved the lowest value among the five methods. These results indicated that learning speaker-invariant features in the recognition model increased the accuracy in modeling MCEPs of the target speakers. On the other hand, "Prop. (Joint)" showed the different tendencies in accordance with the different target speakers, i.e., its LogGVDs were almost the same as "Baseline" when we used MT and similar to "Prop. (DAT)" in the other case. One of the reasons might be the data imbalance of the CSJ corpus that included more male speakers than female speakers.

4.2.2 Speech Recognition Accuracy

Although improving accuracy of speech recognition is not the final goal of the PPG-based many-to-one VC, whether the proposed algorithms affected the accuracy or not deserves to be reported. Here, by using the evaluation data of the CSJ corpus, we calculated the frame-wise phoneme recognition accuracy of the recognition models after training the many-to-one VC models.

Table 3 lists the evaluation results. From the results, we observed that "Baseline" achieved the highest recognition accuracy, which was a natural result since we fixed the recognition model trained to minimize the recognition error during the synthesis model training. Meanwhile, all the proposed algorithms decreased the accuracy, which suggested that the loss functions for training the speech synthesis did not necessarily improve the speech recognition accuracy.

Table 4 MCCs of domain classification models

	Target speaker	
	MT	FT
Baseline	0.36	0.33
Prop. (Joint)	0.22	0.16
Prop. (DAT)	**0.02**	**0.04**
Prop. (GAN)	0.18	0.18
Prop. (DAT-GAN)	0.04	**0.04**

4.2.3 Speaker Invariance of the Speech Recognition Model

To evaluate the robustness of the speech recognition model against the variation in input speakers, we calculated the Matthews correlation coefficients (MCCs) [51] of the domain classification model. The MCC quantifies the performance of a binary classification model and is defined as:

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}, \tag{6}$$

where TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives of a binary classification, respectively. The MCC takes a value between -1 (complete misclassification) and $+1$ (perfect classification), and their mid-value (i.e., 0) corresponds to no better than random classification. Therefore, our goal in the PPG-based many-to-one VC is to cause the speech recognition model to learn features that make the MCC of the domain classification model become close to 0. To calculate the MCCs, we used 50 utterances of the target speaker MT or FT as positive examples and 500 utterances (50 utterances $\times$ 10 speakers) taken from the evaluation data of the CSJ corpus as negative examples.

Table 4 lists the evaluation results. From the results, we found that all the proposed algorithm decreased the MCCs compared with "Baseline." In particular, the use of DAT made the MCCs almost zero, while the other methods did not. Therefore, we concluded that the proposed algorithm using the GAN and DAT not only reduced the distributional differences between natural and generated MCEPs, but also made the speech recognition model more invariant to the domain mismatch between the target and other speakers.

4.3 Subjective Evaluations

We conducted subjective evaluations on the naturalness and speaker similarity of the converted speech for a comparison with conventional methods and ours. Since the speech corpora for building many-to-one VC systems were completely non-parallel, we used the ATR Japanese Speech Database (set C) [52] that included 291 amateur speakers (143

Table 5 MOS scores on naturalness of converted speech with their 95% confidence intervals. Here, we compared "Baseline" with proposed algorithms using GAN

(a) Results of FSs/MSs-to-FT VC

	FSs-to-FT	MSs-to-FT
Baseline	2.70 ± 0.12	2.51 ± 0.11
Prop. (GAN)	**3.00** ± **0.13**	2.55 ± 0.12
Prop. (DAT-GAN)	**2.95** ± **0.13**	**2.75** ± **0.12**

(b) Results of FSs/MSs-to-MT VC

	FSs-to-MT	MSs-to-MT
Baseline	2.63 ± 0.10	2.55 ± 0.11
Prop. (GAN)	**2.94** ± 0.11	**3.01** ± **0.12**
Prop. (DAT-GAN)	**2.96** ± 0.12	**2.96** ± **0.11**

females and 148 males) with a reading-aloud style for choosing source speakers in many-to-one VC. To investigate the effects of variation in source speakers, we selected one parallel speech utterance (phonetically balanced sentence A01) of randomly selected 10 male speakers (MSs) and 10 female speakers (FSs) for evaluating many-to-one VC. Here, we evaluated the performances of conventional and proposed VC algorithms in FSs-to-FT, MSs-to-FT, FSs-to-MT, and MSs-to-MT VC tasks.

4.3.1 Evaluation of Naturalness of Converted Speech Generated by Proposed Algorithms with the GAN

Firstly, we compared "Baseline" with the proposed algorithms using the GAN, i.e., "Prop. (GAN)" and "Prop. (DAT-GAN)," in terms of the naturalness of their converted speech. We conducted five-point scaled mean opinion score (MOS) tests on the naturalness. The converted speech generated by the three many-to-one VC systems was presented to listeners in random order. In the evaluation of FSs-to-FT and MSs-to-FT VC, thirty listeners participated in the assessment by using our crowdsourced subjective evaluation systems. Each listener evaluated 60 converted speech samples (20 source speakers × the three algorithms). Similarly, we conducted the evaluation of FSs-to-MT and MSs-to-MT VC with 25 listeners.

Table 5 shows the experimental results. From the results, we found that "Prop. (DAT-GAN)" outperformed "Baseline" in the all VC tasks, which demonstrated that our algorithm combined the DAT and GAN was effective in improving the naturalness of the converted speech. A noteworthy fact was that the proposed algorithm using only the GAN did not always yield the significant improvement, as shown in Table 5 (a). This result suggested that just using the GAN-based training was insufficient to deal with the differences among speakers observed in PPGs.

4.3.2 Evaluation of Speaker Similarity of Converted Speech Generated by Proposed Algorithms with GAN

Secondly, we compared "Baseline" or "Prop. (GAN)" with "Prop. (DAT-GAN)" in terms of the speaker similarity of their converted speech. We conducted preference XAB tests on the speaker similarity. Three speech utterances of MT or FT not included in the training data were used as reference "X" for evaluating the similarity. The converted speech pairs of the method "A" ("Baseline" or "Prop. (GAN)") and the method "B" ("Prop. (DAT-GAN)") were presented to listeners in random order. In the evaluation of FSs-to-FT and MSs-to-FT VC, thirty listeners participated in the assessment by using our crowdsourced subjective evaluation systems. Each listener evaluated 40 converted speech samples (20 source speakers × the two comparisons). Similarly, we conducted the evaluation of FSs-to-MT and MSs-to-MT VC with 25 listeners.

Table 6 Preference scores of speaker similarity of converted speech with their p-values. Here, we compared "Prop. (DAT-GAN)" with "Baseline" or "Prop. (GAN)"

(a) Results of FSs-to-FT VC

Method A	Score	p-value	Method B
Baseline	0.317 vs. **0.683**	$< 10^{-9}$	Prop. (DAT-GAN)
Prop. (GAN)	0.387 vs. **0.613**	$< 10^{-6}$	Prop. (DAT-GAN)

(b) Results of MSs-to-FT VC

Method A	Score	p-value	Method B
Baseline	0.283 vs. **0.717**	$< 10^{-9}$	Prop. (DAT-GAN)
Prop. (GAN)	0.373 vs. **0.627**	$< 10^{-9}$	Prop. (DAT-GAN)

(c) Results of FSs-to-MT VC

Method A	Score	p-value	Method B
Baseline	0.328 vs. **0.672**	$< 10^{-9}$	Prop. (DAT-GAN)
Prop. (GAN)	0.348 vs. **0.652**	$< 10^{-9}$	Prop. (DAT-GAN)

(d) Results of MSs-to-MT VC

Method A	Score	p-value	Method B
Baseline	0.308 vs. **0.692**	$< 10^{-9}$	Prop. (DAT-GAN)
Prop. (GAN)	0.276 vs. **0.724**	$< 10^{-9}$	Prop. (DAT-GAN)

Table 6 shows the experimental results. From this table, we found that "Prop. (DAT-GAN)" achieved significantly higher preference scores than not only "Baseline" but also "Prop. (GAN)" in the all VC tasks. These results demonstrated that the algorithm was effective in improving the speaker similarity of the converted speech.

4.3.3 Effects of Joint Training and the DAT without the GAN

Thirdly, we further investigated the effects of the other proposed algorithms ("Prop. (Joint)" and "Prop. (DAT)"). As shown in Table 2, these proposed algorithms were unable to reduce LogGVDs to the extent that GAN-based proposed algorithms were able to do so. However, other objective evaluations revealed that there were clear differences between "Baseline" and the two proposed algorithms. Therefore, we compared 1) "Baseline" with "Prop. (Joint)" and 2) "Baseline" with "Prop. (DAT)" by preference AB tests on the naturalness and preference XAB tests on the speaker similarity to clarify the effects caused by the joint training or the DAT-based algorithms. The converted speech pairs of the method

Table 7 Preference scores of naturalness of converted speech with their p-values. Here, we compared "Baseline" with "Prop. (Joint)" or "Prop. (DAT)"

(a) Results of FSs-to-FT VC

Method A	Score	p-value	Method B
Baseline	0.468 vs. 0.532	0.153	Prop. (Joint)
Baseline	0.448 vs. **0.552**	0.002	Prop. (DAT)

(b) Results of MSs-to-FT VC

Method A	Score	p-value	Method B
Baseline	0.512 vs. 0.488	0.592	Prop. (Joint)
Baseline	0.492 vs. 0.508	0.721	Prop. (DAT)

(c) Results of FSs-to-MT VC

Method A	Score	p-value	Method B
Baseline	0.420 vs. **0.580**	$< 10^{-3}$	Prop. (Joint)
Baseline	0.408 vs **0.592**	$< 10^{-3}$	Prop. (DAT)

(d) Results of MSs-to-MT VC

Method A	Score	p-value	Method B
Baseline	0.388 vs. **0.612**	$< 10^{-6}$	Prop. (Joint)
Baseline	0.400 vs. **0.600**	$< 10^{-3}$	Prop. (DAT)

Table 8 Preference scores of speaker similarity of converted speech with their p-values. Here, we compared "Baseline" with "Prop. (Joint)" or "Prop. (DAT)"

(a) Results of FSs-to-FT VC

Method A	Score	p-value	Method B
Baseline	0.500 vs. 0.500	1.000	Prop. (Joint)
Baseline	0.484 vs. 0.516	0.475	Prop. (DAT)

(b) Results of MSs-to-FT VC

Method A	Score	p-value	Method B
Baseline	0.456 vs. 0.544	0.049	Prop. (Joint)
Baseline	0.432 vs. **0.568**	0.002	Prop. (DAT)

(c) Results of FSs-to-MT VC

Method A	Score	p-value	Method B
Baseline	0.424 vs. **0.576**	0.001	Prop. (Joint)
Baseline	0.404 vs. **0.596**	$< 10^{-3}$	Prop. (DAT)

(d) Results of MSs-to-MT VC

Method A	Score	p-value	Method B
Baseline	0.408 vs. **0.592**	$< 10^{-3}$	Prop. (Joint)
Baseline	0.396 vs. **0.604**	$< 10^{-3}$	Prop. (DAT)

"A" ("Baseline") and the method "B" ("Prop. (Joint)" or "Prop. (DAT)") were presented to listeners in random order. Twenty five listeners participated in the evaluations and each listener evaluated 40 converted speech samples (20 source speakers × the two comparisons).

Tables 7 and 8 shows the evaluation results on the naturalness and speaker similarity, respectively. From the results, we found that the preference scores of the two proposed algorithms were comparable or superior to those of "Baseline." In particular, we observed two remarkable points: 1) "Prop. (DAT)" improved the speaker similarity in the inter-gender VC (i.e., MSs-to-FT VC and FSs-to-MT VC) and 2) the two proposed algorithms outperformed "Baseline" in FSs-/MSs-to-MT VC with regard to both the naturalness and speaker similarity in spite of decreasing the speech recognition accuracy as shown in Table 3. These results suggested that 1) without using the GAN, the DAT had the effect of improving the speaker similarity by learning speaker-invariant features in the recognition model and 2) the high speech recognition accuracy does not guarantee the quality of converted speech and the proposed joint training had potential to optimize the recognition model for the many-to-one VC task.

Table 9 Preference scores of naturalness of converted speech with their p-values. Here, we compared "StarGAN-VC" with "Prop. (DAT-GAN)"

(a) Results of FSs-to-FT VC

Method A	Score	p-value	Method B
StarGAN-VC	0.168 vs. **0.832**	$< 10^{-9}$	Prop. (DAT-GAN)

(b) Results of MSs-to-FT VC

Method A	Score	p-value	Method B
StarGAN-VC	0.152 vs. **0.848**	$< 10^{-9}$	Prop. (DAT-GAN)

(c) Results of FSs-to-MT VC

Method A	Score	p-value	Method B
StarGAN-VC	0.300 vs. **0.700**	$< 10^{-9}$	Prop. (DAT-GAN)

(d) Results of MSs-to-MT VC

Method A	Score	p-value	Method B
StarGAN-VC	0.400 vs. **0.600**	$< 10^{-3}$	Prop. (DAT-GAN)

Table 10 Preference scores of speaker similarity of converted speech with their p-values. Here, we compared "StarGAN-VC" with "Prop. (DAT-GAN)"

(a) Results of FSs-to-FT VC

Method A	Score	p-value	Method B
StarGAN-VC	0.140 vs. **0.860**	$< 10^{-9}$	Prop. (DAT-GAN)

(b) Results of MSs-to-FT VC

Method A	Score	p-value	Method B
StarGAN-VC	0.096 vs. **0.904**	$< 10^{-9}$	Prop. (DAT-GAN)

(c) Results of FSs-to-MT VC

Method A	Score	p-value	Method B
StarGAN-VC	0.132 vs. **0.868**	$< 10^{-9}$	Prop. (DAT-GAN)

(d) Results of MSs-to-MT VC

Method A	Score	p-value	Method B
StarGAN-VC	0.176 vs. **0.824**	$< 10^{-9}$	Prop. (DAT-GAN)

4.3.4 Comparison with Another Non-Parallel VC Method

Finally, we compared the best proposed algorithm, i.e., "Prop. (DAT-GAN)," with another state-of-the-art non-parallel VC method. Here, we used StarGAN-VC [29] as the competitive VC method that can achieve high-quality many-to-many VC without using any parallel speech corpora. We built the StarGAN-VC model by using an open-source implementation[†] and 100 utterances of 22 speakers (MSs, FSs, MT, and FT) for the training. Similar to Sect. 4.3.3, we

[†]https://github.com/hujinsen/pytorch-StarGAN-VC

conducted preference (X)AB tests with 25 listeners.

Tables 9 and 10 show the evaluation results on the naturalness and speaker similarity, respectively. From the results, we concluded that "Prop. (DAT-GAN)" outperformed not only the conventional PPG-based many-to-one VC but also state-of-the-art non-parallel VC method.

5. Conclusion

We proposed a joint adversarial training algorithm for speech recognition and synthesis models used in deep neural network (DNN)-based many-to-one voice conversion (VC). For making the recognition model more robust towards the differences among input speakers, we introduced a domain-adversarial training of the recognition model. We also incorporated a generative adversarial network-based training of the synthesis model for overcoming the over-smoothing effect of generated acoustic features. We formulated a unified objective function for jointly training the recognition and synthesis models so that they are optimized for many-to-one VC. Experimental evaluation demonstrated that our algorithm significantly improved the converted speech quality compared with a conventional algorithm for DNN-based many-to-one VC and StarGAN-based non-parallel VC. In the future, we will investigate the effect of the hyperparameters of the proposed algorithm and introduce sequence-to-sequence modeling [53] into our algorithm.

Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Number 18J22090.

References

[1] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," IEEE Trans. Speech and Audio Processing, vol.6, no.2, pp.131–142, March 1988.
[2] A.B. Kain, J.-P. Hosom, X. Niu, J.P.H. van Santen, M. Fried-Oken, and J. Staehely, "Improving the intelligibility of dysarthric speech," Speech Communication, vol.49, no.9, pp.743–756, 2007.
[3] H. Doi, T. Toda, T. Nakano, M. Goto, and S. Nakamura, "Singing voice conversion method based on many-to-many Eigenvoice conversion and training data generation using a singing-to-singing synthesis system," Proc. APSIPA ASC, pp.1–6, Nov. 2012.
[4] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z.H. Ling, "The Voice Conversion Challenge 2018: Promoting development of parallel and nonparallel methods," Odyssey Workshop, Les Sables d'Olonne, France, pp.195–202, June 2018.
[5] T. Toda, A.W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," IEEE Trans. Audio, Speech, Language Process., vol.15, no.8, pp.2222–2235, Nov. 2007.
[6] T. Toda, O. Ohtani, and K. Shikano, "One-to-many and many-to-one voice conversion based on Eigenvoices," Proc. ICASSP, Hawaii, U.S.A., pp.1249–1252, April 2007.
[7] S. Desai, E.V. Raghavendra, B. Yegnanarayana, A.W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," Proc. ICASSP, Taipei, Taiwan, pp.3893–3896, April 2009.
[8] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kunio, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks," Proc. INTERSPEECH, Stockholm, Sweden, pp.1283–1287, Aug. 2017.
[9] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "AttS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," Proc. ICASSP, Brighton, U.K., pp.6805–6809, May 2019.
[10] T. Nakashika, T. Takiguchi, and Y. Minami, "Non-parallel training in voice conversion using an adaptive restricted Boltzmann machine," IEEE/ACM Trans. Audio, Speech, Language Process., vol.24, no.11, pp.2032–2045, Nov. 2016.
[11] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," Proc. APSIPA ASC, Jeju, South Korea, Dec. 2016.
[12] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," Proc. ICASSP, Alberta, Canada, pp.5274–5278, April 2018.
[13] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," Proc. ICASSP, Vancouver, Canada, pp.7962–7966, May 2013.
[14] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol.313, no.5786, pp.504–507, 2006.
[15] Z. Wu, P.L.D. Leon, C. Demiroglu, A. Khodabakhsh, S. King, Z. Ling, D. Saito, B. Stewart, T. Toda, M. Wester, and J. Yamagishi, "Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance," IEEE/ACM Trans. Audio, Speech, Language Process., vol.24, no.4, pp.768–783, 2016.
[16] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," Proc. ICME, Seattle, U.S.A., July 2016.
[17] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," Proc. INTERSPEECH, Stockholm, Sweden, pp.1268–1272, Aug. 2017.
[18] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," Speech Communication, vol.51, no.11, pp.1039–1064, 2009.
[19] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," Journal of Machine Learning Research, vol.17, no.59, pp.1–35, April 2016.
[20] Y. Saito, S. Takamichi, and H. Saruwatari, "Statistical parametric speech synthesis incorporating generative adversarial networks," IEEE/ACM Trans. Audio, Speech, Language Process., vol.26, no.1, pp.84–96, Jan. 2018.
[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Proc. NIPS, Montreal, Canada, pp.2672–2680, 2014.
[22] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, "Domain adversarial training for accented speech recognition," Proc. ICASSP, Alberta, Canada, pp.4854–4858, April 2018.
[23] Q. Wang, W. Rao, S. Sun, L. Xie, E.S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," Proc. ICASSP, Alberta, Canada, pp.4889–4893, April 2018.
[24] J.-C. Chou, C.-C. Yeh, H.-Y. Lee, and L.-S. Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," Proc. INTERSPEECH, Hyderabad, India, pp.501–505, Sept. 2018.
[25] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv, vol.abs/1701.07875, 2017.
[26] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks," Proc. EUSIPCO, Rome, Italy, pp.2114–2118, Sept. 2018.

[27] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC2: Improved CycleGAN-based non-parallel voice conversion," Proc. ICASSP, Brighton, U.K., pp.6820–6824, May 2019.
[28] J.-Y. Zhu, T. Park, P. Isola, and A.A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," Proc. ICCV, Venice, Italy, pp.2223–2232, Oct. 2017.
[29] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," Proc. SLT, Greece, Athens, pp.266–273, Dec. 2018.
[30] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "StarGAN-VC2: Rethinking conditional methods for stargan-based voice conversion," Proc. INTERSPEECH, Graz, Austria, pp.679–683, Sept. 2019.
[31] 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," Prop. CVPR, Salt Lake City, U.S.A., pp.8789–8797, June 2018.
[32] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," Proc. ASRU, Okinawa, Japan, pp.301–308, Dec. 2017.
[33] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv, vol.abs/1411.1784, 2014.
[34] N. Hojo, Y. Ijima, and H. Mizuno, "DNN-based speech synthesis using speaker codes," IEICE Trans. Inf. & Syst., vol.E101-D, no.2, pp.462–472, Feb. 2018.
[35] Y. Zhou, X. Tian, H. Xu, R.K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," Proc. ICASSP, Brighton, U.K., pp.6790–6794, May 2019.
[36] S.H. Mohammadi and T. Kim, "One-shot voice conversion with disentangled representations by leveraging phonetic posteriorgrams," Proc. INTERSPEECH, Graz, Austria, pp.704–708, Sept. 2019.
[37] A. 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, vol.abs/1609.03499, 2016.
[38] H. Lu, Z. Wu, R. Li, S. Kang, J. Jia, and H. Meng, "A compact framework for voice conversion using wavenet conditioned on phonetic posteriorgrams," Proc. ICASSP, Brighton, U.K., pp.6810–6814, May 2019.
[39] S. Liu, Y. Cao, X. Wu, L. Sun, X. Liu, and H. Meng, "Jointly trained conversion model and WaveNet vocoder for non-parallel voice conversion using mel-spectrograms and phonetic posteriorgrams," Proc. INTERSPEECH, Graz, Austria, pp.704–708, Sept. 2019.
[40] K. Sugiura, Y. Shiga, H. Kawai, T. Misu, and C. Hori, "A cloud robotics approach towards dialogue-oriented robot speech," Advanced Robotics, vol.29, no.7, pp.449–456, March 2015.
[41] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," Proc. LREC, pp.947–952, May 2000.
[42] 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, July 2016.
[43] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," Speech Communication, vol.84, pp.57–65, Nov. 2016.
[44] A. Camacho, "Swipe: A sawtooth waveform inspired pitch estimator for speech and music," Ph.D. dissertation, University of Florida, 2007.
[45] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," Speech Coding and Synthesis, pp.495–518, 1995.
[46] O. Abdel-Hamid, A.r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," IEEE/ACM Trans. Audio, Speech, Language Process., vol.22, no.10, pp.1533–1545, Oct. 2014.
[47] A.L. Maas, A.Y. Hannun, and A.Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," Proc. ICML, 2013.
[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol.15, no.1, pp.1929–1958, April 2014.
[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Proc. ICML, 2015.
[50] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," Journal of Machine Learning Research, vol.12, pp.2121–2159, July 2011.
[51] B.W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," Biochimica et Biophysica Acta (BBA) - Protein Structure, vol.405, no.2, pp.442–451, Oct. 1975.
[52] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "ATR Japanese speech database as a tool of speech recognition and synthesis," Speech Communication, vol.9, no.4, pp.357–363, Aug. 1990.
[53] 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. Audio, Speech, Language Process., vol.27, no.3, pp.631–644, Jan. 2019.

Yuki Saito received his M.S. degree from the Graduate School of Information Science and Technology, The University of Tokyo, Japan, in 2018. He is currently a Ph.D. student at The University of Tokyo. His research interests include speech synthesis, voice conversion, and machine learning. He has received eight paper awards including the 2017 IEICE ISS Young Researcher's Award in Speech Field. He is a Student Member of ASJ and a Student Member of IEEE SPS.

Kei Akuzawa received his M.S. degree from the Graduate School of Engineering, The University of Tokyo, Japan, in 2019. He is currently a Ph.D. student at The University of Tokyo. He has received the Japanese Society for Artificial Intelligence (JSAI) Annual Conference Student Incentive Award in 2018. His research interests include speech synthesis and sequential generative models.

Kentaro Tachibana received his M.S. degree from the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan, in 2008. From 2008–2017, he worked at Toshiba Corporate Research and Development Center, Japan. From 2014–2017, he was employed at National Institute of Information and Communications Technology (NICT), Japan, for temporary assignment. From 2017, he has been working at DeNA Co. Ltd., Japan, where he continues to work today. His research interests include statistical speech processing, especially voice conversion and text-to-speech synthesis.