NON-PARALLEL VOICE CONVERSION USING VARIATIONAL AUTOENCODERS CONDITIONED BY PHONETIC POSTERIORGRAMS AND D-VECTORS

Yuki Saito[†,‡]*, Yusuke Ijima*[‡]*, Kyosuke Nishida*[‡]*, and Shinnosuke Takamichi*[†]

[†] Graduate School of Information Science and Technology, The University of Tokyo
[‡] NTT Media Intelligence Laboratories, NTT Corporation, Japan

ABSTRACT

This paper proposes novel frameworks for non-parallel voice conversion (VC) using variational autoencoders (VAEs). Although conventional VAE-based VC models can be trained using non-parallel speech corpora with given speaker representations, phonetic contents of the converted speech tend to vanish because of an over-regularization issue often observed in latent variables of the VAEs. To overcome the issue, this paper proposes a VAE-based non-parallel VC conditioned by not only the speaker representations but also phonetic contents of speech represented as phonetic posteriorgrams (PPGs). Since the phonetic contents are given during the training, we can expect that the VC models effectively learn speaker-independent latent features of speech. Focusing on the point, this paper also extends the conventional VAE-based non-parallel VC to many-to-many VC that can convert arbitrary speakers' characteristics into another arbitrary speakers' ones. We investigate two methods to estimate speaker representations for speakers not included in speech corpora used for training VC models: 1) adapting conventional speaker codes, and 2) using d-vectors for the speaker representations. Experimental results demonstrate that 1) PPGs successfully improve both naturalness and speaker similarity of the converted speech, and 2) both speaker codes and d-vectors can be adopted to the VAE-based many-to-many non-parallel VC.

Index Terms— VAE-based non-parallel VC, phonetic posteriorgrams, d-vectors, many-to-many VC

1. INTRODUCTION

Voice conversion (VC) [1] is a technique to convert characteristics of source speech into those of target speech while keeping its linguistic information. Recently, deep neural networks (DNNs) [2] have been adopted to VC models which convert source speech parameters into target speech parameters because they can accurately convert characteristics of speech compared to conventional Gaussian mixture models (GMMs) [3]. The DNNs are typically trained by using parallel speech corpora that include the same utterances recorded by source and target speakers. Although these models can learn frame-wise mapping from source speech parameters into target speech parameters and significantly improve quality of the converted speech, recording parallel speech corpora for training the VC models is often difficult in practice.

To overcome the difficulty caused by using parallel speech corpora, researchers have been investigated non-parallel VC, which does not require any parallel speech corpora to construct VC models. Recently, variational autoencoders (VAEs) [4] have been adopted to the VC models for non-parallel VC because their training criterion is more tractable than restricted Boltzmann machines [5]. In the conventional VAE-based non-parallel VC [6], encoder networks extract speaker-independent latent variables from input speech parameters, and decoder networks reconstruct the parameters from the latent variables, and given speaker representations. Thus, we can suppose that the latent variables represent phonetic contents of speech, and VC is done by modifying the speaker representations fed into the decoder networks. However, quality of the converted speech is lower than that converted by DNNs trained with parallel speech corpora. One of the primal issues that causes the quality degradation is an over-regularization effect often observed in the latent variables of the VAEs [7], which makes the distribution of the latent variables be too simplistic. One can address the issue by using more complex prior distribution of the latent variables such as GMMs [8], but adopting this idea to the VAE-based non-parallel VC seems to be difficult because variation in the phonetic contents is typically large and thus the number of the cluster of the GMMs should not be readily decided.

To improve the quality of speech converted by conventional VAE-based non-parallel VC, this paper proposes an effective framework for training the VC models. In the framework, VAEs are trained on the condition of not only the speaker representations but also phonetic contents of the input speech are given during the training. Assuming that large speech corpora for constructing speaker-independent automatic speech recognition (ASR) models are available, we introduce output of the ASR models (phonetic posteriorgrams: PPGs) [9] to the VAE-based VC because they can be regarded as the latent variables of the phonetic contents. Moreover, focusing on an ability of the VAEs to extract speaker-independent latent variables of speech taken from many and unspecified speakers, we also extend the conventional VAE-based non-parallel VC to many-to-many VC, which can convert arbitrary speakers' characteristics into another arbitrary speakers' ones. To this end, we investigate effective speaker representations for many-to-many non-parallel VC. In addition to conventional speaker codes [10], we introduce d-vectors [11], which are obtained by output of pre-trained automatic speaker verification (ASV) models, to the speaker representations. Since the effectiveness of the d-vectors are well known in ASV, we can regard them as latent variables of the speaker representations for arbitrary speakers. Experimental results demonstrate that 1) PPGs successfully improve both naturalness and speaker similarity of the converted speech compared to the conventional VAE-based non-parallel VC, and 2) both speaker codes and d-vectors can be adopted to the VAE-based many-to-many non-parallel VC.

2. CONVENTIONAL VAE-BASED NON-PARALLEL VC

2.1. Non-parallel VC using VAEs conditioned by speaker codes [6]

VAE-based VC models represent probabilistic generative models of speech that speech parameters $\boldsymbol{x}$ are generated from their latent variables $\boldsymbol{z}$ and speaker representations $\boldsymbol{y}_{\mathrm{s}}$. In [6], speaker codes [10]

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

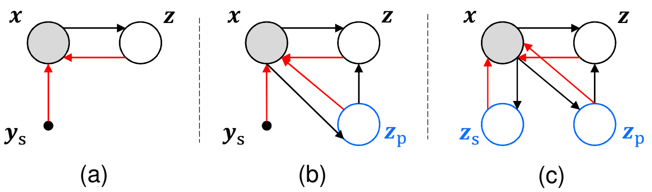
$\boldsymbol{x}$
$\boldsymbol{z}$
$\boldsymbol{y}_{\mathrm{s}}$
(a)
$\boldsymbol{x}$
$\boldsymbol{z}$
$\boldsymbol{y}_{\mathrm{s}}$
$\boldsymbol{z}_{\mathrm{p}}$
(b)
$\boldsymbol{x}$
$\boldsymbol{z}$
$\boldsymbol{z}_{\mathrm{s}}$
$\boldsymbol{z}_{\mathrm{p}}$
(c)

Fig. 1. Directed graphical models of VAE-based VC models; (a) VAEs conditioned by one-hot speaker codes $\boldsymbol{y}_{\mathrm{s}}$, (b) VAEs conditioned by one-hot speaker codes $\boldsymbol{y}_{\mathrm{s}}$ and PPGs $\boldsymbol{z}_{\mathrm{p}}$, and (c) VAEs conditioned by d-vectors $\boldsymbol{z}_{\mathrm{s}}$ and PPGs $\boldsymbol{z}_{\mathrm{p}}$. Black and red arrows denote inferring latent variables and generating speech parameters $\boldsymbol{x}$, respectively.

are adopted to the speaker representations, which use 1-of-S representation to identify the one of S-speakers. The speaker codes for the i-th speaker are defined as follows:

$$y_{\mathrm{s}}^{(i)}(k) = \begin{cases} 1 & \text{if } \ k = i \\ 0 & \text{otherwise} \end{cases} \qquad (1 \le k \le S). \tag{1}$$

Assuming that $\boldsymbol{z}$ is independent of $\boldsymbol{y}_{\mathrm{s}}$, our objective is to estimate model parameters $\boldsymbol{\theta}$ that maximize the marginal likelihood of the speech parameters conditioned by given speaker representations, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{y}_{\mathrm{s}}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y}_{\mathrm{s}}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z}$, where $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is a prior of the latent variables. Since the integral in the likelihood is intractable, we introduce two networks; speaker-independent encoder networks $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, which approximate true posterior of the latent variables $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, and speaker-dependent decoder networks, which approximate true posterior of the speech parameters $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y}_{\mathrm{s}})$. $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are sets of model parameters of the encoder and decoder, respectively. The objective for training the VAEs is maximizing the variational lower bound of the log likelihood defined as follows:

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{y}_{\mathrm{s}}\right) = &-D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}\right) || \, p_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)\right) \\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y}_{\mathrm{s}}\right)\right], \end{aligned} \tag{2}$$

where $D_{\mathrm{KL}}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence between two distributions. We assume that both encoder and decoder networks represent diagonal Gaussian distributions, of which the mean and covariance are estimated by the networks. The isotropic Gaussian distribution $\mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$ is typically adopted to the prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ in order to obtain closed form of the KL term in Eq. (2). The reparameterization trick [4] is used for the backpropagation algorithm. Figure 1(a) illustrates the directed graphical model of the VAEs.

After training the VAEs, VC can be easily performed by feeding the speaker codes of the target speaker into the decoder. For instance, when we convert source speech parameters into those of the j-th speaker included in speech corpora used for the training, we feed $\boldsymbol{y}_{\mathrm{s}}^{(j)}$, i.e., $y_{\mathrm{s}}^{(j)}(k) = 1$ if $k = j$, into the decoder frame-by-frame.

2.2. Problems

Since we assume that the latent variables are independent of the speaker representations, they can be expected to represent phonetic contents of speech. However, in the conventional VAE-based non-parallel VC [6], the phonetic contents tend to vanish because of too strong influence of the prior used in the KL term of Eq. (2). This issue is known as an over-regularization of the latent variables [7], which often makes the obtained latent variables be overly simplified and poorly represent the underlying structure of the phonetic contents. Moreover, although the VAEs have a potential to extract speaker-independent latent variables from many and unspecified speakers, currently they can only convert characteristics of speakers into those of target speakers included in speech corpora used for training the VAEs.

3. PROPOSED VAE-BASED NON-PARALLEL VC

Here, we propose a novel framework for the VAE-based non-parallel VC to improve the converted speech quality. Moreover, we extend the conventional VAE-based non-parallel VC to many-to-many VC, which can convert arbitrary speakers' characteristics into another arbitrary speakers' ones.

3.1. Non-parallel VC using VAEs conditioned by PPGs and speaker codes

Instead of estimating phonetic contents of source speech as the latent variables of the VAEs, we directly utilize the phonetic contents for training the VAEs. Although a straightforward way to realize this is to use phoneme sequences, we adopt PPGs [9], which are obtained by output of pre-trained ASR models $R(\cdot)$, as the phonetic contents because we can expect them to represent speaker-independent latent variables of the phonetic contents. Let $\boldsymbol{z}_{\mathrm{p}} = R(\boldsymbol{x})$ be PPGs predicted from speech parameters $\boldsymbol{x}$. In the proposed framework, the objective function shown in Eq. (2) is rewritten as:

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{y}_{\mathrm{s}}, \boldsymbol{z}_{\mathrm{p}}\right) = &-D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{z}_{\mathrm{p}}\right) || \, p_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)\right) \\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{z}_{\mathrm{p}})}\left[\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{z}_{\mathrm{p}}, \boldsymbol{y}_{\mathrm{s}}\right)\right], \end{aligned} \tag{3}$$

that is, the PPGs $\boldsymbol{z}_{\mathrm{p}}$ are fed into both of the encoder and decoder networks, which guarantees that the phonetic contents of source speech are kept in the training and conversion stage. Figure 1(b) shows the directed graphical model of the proposed VAE-based VC using PPGs and speaker codes.

3.2. Effective speaker representations for many-to-many VC

When target speaker of many-to-many VC is not included in speech corpora used for training VAEs, we have to estimate speaker representations for the new speaker using small amount of utterances. Here, we investigate two methods for estimating the speaker representations for new speakers; 1) adapting speaker codes to the new speaker, and 2) using d-vectors which are effective in ASV.

3.2.1. Adapting speaker codes using backpropagation algorithm

One way to achieve many-to-many VC with conventional speaker codes is adapting them to the new speaker, which was firstly investigated in DNN-based multi-speaker text-to-synthesis [12]. First, we set initial values of the estimated speaker codes as $\hat{y}_{\mathrm{s}}^{(\mathrm{tar})}(k) = 1/S \ (1 \le k \le S)$. Then, we calculate the mean squared error (MSE) between input speech parameters $\boldsymbol{x}^{(\mathrm{tar})}$ and reconstructed speech parameters $\hat{\boldsymbol{x}}^{(\mathrm{tar})}$ defined as $L_{\mathrm{MSE}}(\boldsymbol{x}^{(\mathrm{tar})}, \hat{\boldsymbol{x}}^{(\mathrm{tar})}) = (\boldsymbol{x}^{(\mathrm{tar})} - \hat{\boldsymbol{x}}^{(\mathrm{tar})})^{\top}(\boldsymbol{x}^{(\mathrm{tar})} - \hat{\boldsymbol{x}}^{(\mathrm{tar})})$, where $\hat{\boldsymbol{x}}^{(\mathrm{tar})} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(\mathrm{tar})}|\hat{\boldsymbol{z}}, \hat{\boldsymbol{y}}_{\mathrm{s}}^{(\mathrm{tar})})$ and $\hat{\boldsymbol{z}} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(\mathrm{tar})})$. Finally, we calculate the gradient of the MSE by the estimated speaker codes $\partial L_{\mathrm{MSE}}/\partial \hat{\boldsymbol{y}}_{\mathrm{s}}^{(\mathrm{tar})}$ with the backpropagation algorithm, and update the speaker codes as $\hat{\boldsymbol{y}}_{\mathrm{s}}^{(\mathrm{tar})} - \eta \partial L_{\mathrm{MSE}}/\partial \hat{\boldsymbol{y}}_{\mathrm{s}}^{(\mathrm{tar})}$ with small coefficient η. We iterate the procedure to obtain better speaker codes for the new speaker.

3.2.2. Using d-vectors for speaker representations

d-vectors [11], which are obtained by bottleneck features of pre-trained ASV models $V(\cdot)$, are adopted to the speaker representations for many-to-many VC. Because the role of the ASV models is to extract features used for identifying specific speaker, We can regard the d-vectors as latent variables of the speaker representations. Let $\boldsymbol{z}_{\mathrm{s}} = V(\boldsymbol{x})$ be d-vectors extracted from speech parameters $\boldsymbol{x}$. Here, the objective function shown in Eq. (3) is rewritten as:

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{z}_{\mathrm{s}}, \boldsymbol{z}_{\mathrm{p}}\right) = &-D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{z}_{\mathrm{p}}\right) || \, p_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)\right) \\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{z}_{\mathrm{p}})}\left[\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{z}_{\mathrm{p}}, \boldsymbol{z}_{\mathrm{s}}\right)\right], \end{aligned} \tag{4}$$

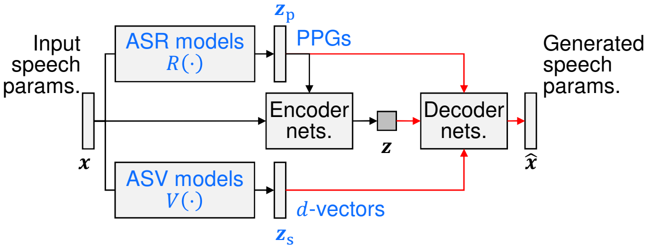

Fig. 2. Proposed VAE-based VC conditioned with PPGs and d-vectors corresponding to directed graphical model shown in Fig. 1(c). ASR and ASV models are not updated in training for encoder and decoder networks.

that is, the conventional *discrete* speaker codes are replaced with the *continuous* d-vectors. In training the VAEs, the d-vectors are fed into the decoder networks frame-by-frame in the same manner as the speaker codes. In conversion stage, the speaker representations for new target speaker are estimated as the averaged values of the d-vectors in *voiced* regions. Note that, this differs from the traditional use of the d-vectors in literatures of ASV, which represents the characteristics of speakers as the averaged values of the d-vectors in *all* regions. Figures 1(c) and 2 show the directed graphical model and overview of the proposed VAE-based VC using PPGs and d-vectors, respectively.

3.3. Discussion

In the proposed VAE-based VC using PPGs and d-vectors, we have to construct the ASR and ASV models by using relatively large amount of speech corpora. Although labeling data for training these models somewhat costs, we can incorporate the models into the proposed framework for training the VAEs in the same manner as semi-supervised learning of a conditional VAEs [13]. Moreover, techniques for end-to-end speech processing [14, 15] can be also applied to the proposed VAE-based VC.

4. EXPERIMENTAL EVALUATION

4.1. Experimental conditions

We used two speech corpora for the evaluation. The one for training ASR and ASV models for the proposed VAE-based VC using PPGs and d-vectors included Japanese 260 speakers (130 male and 130 female speakers). Each speaker uttered about 100 utterances, and total time of the speech corpus was about 31 hours. The other for training/evaluating VC models included Japanese three speakers (two male and one female speakers). We constructed two VC models for male-to-male and male-to-female conversion. Since 425 fully-parallel utterances were recorded by the three speakers, we used 400 utterances for the training and 25 utterances for the evaluation. In order to make non-parallel setting in the VAE-based VC, we divided the 400 utterances into two subsets; i.e., the 1st-through-200th utterances were taken from source speaker, and the remainders were taken from target speaker. The sampling rate of the all speech corpora was 22.05 kHz. The STRAIGHT vocoder [16] was employed to extract 40 dimensional mel-cepstral coefficients, 10 band aperiodicities, log F0, and U/V at 5 ms steps. The mel-cepstral coefficients were normalized to have zero-mean unit-variance. In the conversion stage, the 1st-through-39th mel-cepstral coefficients and their dynamic features were converted by VC models. The input 0th mel-cepstral coefficients were directly used as those of target speech. The MLPG [17] was performed to generate static mel-cepstral coefficients. Input F0 was linearly transformed, and band-aperiodicity was not transformed.

In the evaluation, we compared the performances of the following four VC models.

FFNN: Feed-Forward DNNs trained by using parallel speech corpora

VAE-SC: VAEs using speaker codes [6]

VAE-SC-PPG: VAEs using speaker codes and PPGs

VAE-DV-PPG: VAEs using d-vectors and PPGs

These models were firstly evaluated in one-to-one VC, which trained VC models by using speech corpora including only source and target speakers. In one-to-one VC, the VAEs were trained with *completely non-parallel* speech corpora, while the DNNs used in "FFNN" were trained with *fully-parallel* speech corpora aligned by using the dynamic time warping (DTW) algorithm, which were referred to the ideal baseline of the VC models. Besides, "VAE-SC-PPG" and "VAE-DV-PPG" were evaluated in many-to-many VC, which trained VC models by using speech corpora including 260 speakers used for constructing the ASR and ASV models, In the conversion stage, speaker representations for the target speaker were estimated using the methods described in Sections 3.3.1 and 3.3.2.

All architectures for DNNs and VAEs used in the evaluation were Feed-Forward. The ASR models predicted 56-dimensional PPGs frame-by-frame. The hidden layers of the ASR models had 4 × 1024 units with sigmoid non-linearity. The ASV models predicted posterior probabilities of the speaker identity. Here, in addition to the 260 speakers, the one value which denotes unvoiced region was attached to the speaker identity. The hidden layers of the ASV models had 4 × 256 units with sigmoid non-linearity. The 16-dimensional d-vectors were extracted from the bottleneck layer of the ASV models. In the VAEs, the encoder networks had two hidden layers with rectified linear unit (ReLU) [18] non-linearity. The number of hidden units for the first and second hidden layers were 256 and 128, respectively. The architecture for the decoder networks was symmetric about that for the encoder. The dimensionality of the latent variables was 64. Feed-Forward DNNs were constructed by using fully-parallel speech corpora including only source and target speakers. The hidden layers of the DNNs had 4 × 128 units with ReLU non-linearity. The optimization algorithm was AdaGrad [19], whose learning rate was set to 0.01. All of the VC models were trained with 25 epochs.

4.2. Objective evaluation

We calculated mel-cepstral distortions (MCDs) between target and converted mel-cepstral coefficients. We aligned frame length of target and converted mel-cepstral coefficients by using DTW algorithm to calculate MCDs. The effects of the number of utterances used for training VC models or estimating new speaker representations were also investigated. The four VC models to be compared in one-to-one VC were trained by using 5, 10, 25, 50, 100, and 200 utterances. In many-to-many VC, speaker representations for the target speaker were estimated by using the same numbers of the Utterances as used in the training for one-to-one VC.

Figure 3 shows the results in the evaluation of one-to-one VC. Nevertheless the VC models were trained *completely non-parallel* speech corpora, MCDs of the proposed "VAE-SC-PPG" and "VAE-DV-PPG" were significantly improved compared with the conventional "VAE-SC," and became closer to those of "FFNN" trained with *fully-parallel* speech corpora. Moreover, the MCDs of "VAE-DV-PPG" were slightly lower than those of "VAE-SC-PPG," which suggested that the continuous speaker representations worked better in VAE-based non-parallel VC.

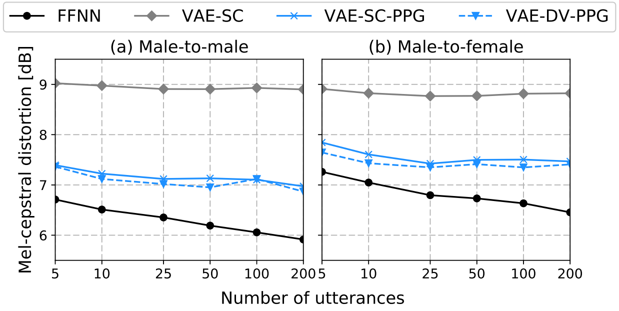

Fig. 3. MCDs of converted speech in one-to-one VC. Only "FFNN" was trained by using fully-parallel speech corpora.

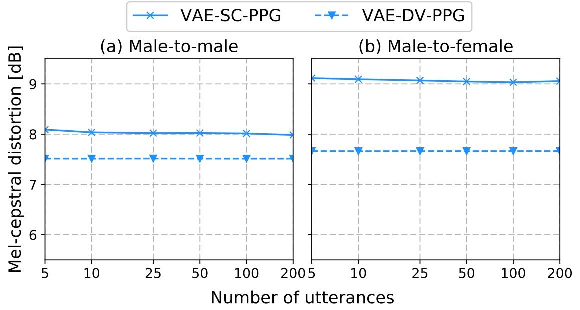

Fig. 4. MCDs of converted speech in many-to-many VC.

Figure 4 shows the results in the evaluation of many-to-many VC. Focusing on the number of utterances used to estimate the speaker representations, the MCDs of "VAE-SC-PPG" had a tendency to decrease by using more utterances for the estimation, although the improvements were very limited. Meanwhile, the MCDs of "VAE-DV-PPG" were almost constant and always lower than those of the "VAE-SC-PPG," regardless of the number of utterances and gender of the target speaker. These results indicated that using d-vectors was more effective for estimating the speaker representations than adapting speaker codes in many-to-many non-parallel VC.

4.3. Subjective evaluation

We conducted subjective evaluations in terms of naturalness and speaker similarity of the converted speech. Here, six VC models named as "FFNN," "VAE-SC," "VAE-SC-PPG (one-to-one)," "VAE-DV-PPG (one-to-one)," "VAE-SC-PPG (many-to-many)," and "VAE-DV-PPG (many-to-many)" were compared at the same time. The fully-parallel speech corpora including 400 utterances of source and target speakers were only used to train "FFNN." The non-parallel speech corpora including 200 utterances of source and target speakers were used to train "VAE-SC," "VAE-SC-PPG (one-to-one)," and "VAE-DV-PPG (one-to-one)." 100 utterances of target speakers were used for estimating their speaker representations for "VAE-SC-PPG (many-to-many)" and "VAE-DV-PPG (many-to-many)." A five-point scaled mean opinion scores (MOS) test was conducted to evaluate naturalness of the converted speech. Speech samples generated by each model were presented listeners in random order. Similarly, a five-point scaled differential MOS (DMOS) test was conducted to evaluate speaker similarity of the converted speech. Re-synthesized speech samples from their speech parameters were presented with corresponding converted speech in random order. 8 listeners participated in each evaluation.

Figure 5 shows the results. Here, only "FFNN" was trained with aligned fully-parallel speech corpora and the scores were referred to the ideal baseline of the VC models. Focusing on the resultant scores of one-to-one VC, the proposed "VAE-SC-PPG (one-to-one)" and "VAE-DV-PPG (one-to-one)" achieved significantly higher scores than those of the conventional "VAE-SC" in terms of both naturalness and speaker similarity, which demonstrated that the PPGs successfully improved quality of the converted speech in the VAE-based non-parallel VC. On the other hand, focusing on the scores of many-to-many VC, the proposed "VAE-SC-PPG (many-to-many)" and "VAE-DV-PPG (many-to-many)" achieved almost the same performances as those in one-to-one VC, even though the source and target speakers were not included in speech corpora used for constructing the VC models. These results demonstrates that the conventional VAE-based non-parallel VC was extended to many-to-many VC by using effectively estimated speaker representations. We also found that the d-vectors were effective to improve naturalness of the converted speech in inner-gender VC.

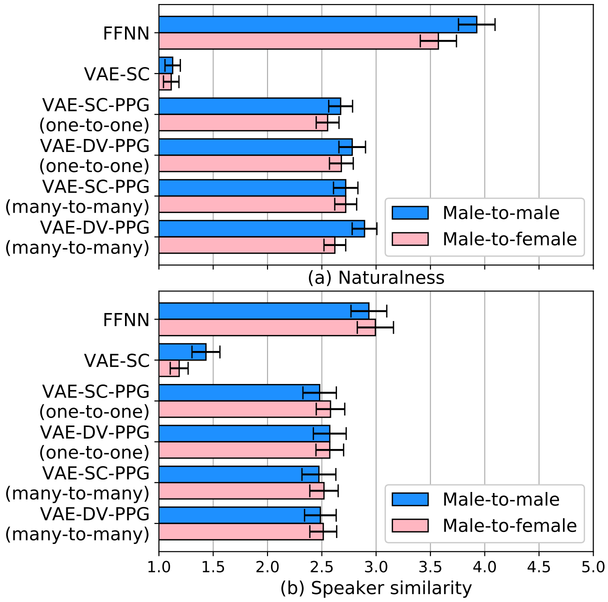

Fig. 5. Results of subjective evaluations in terms of (a) naturalness and (b) speaker similarity with 95% confidence intervals.

5. CONCLUSION

This paper proposed a novel framework for non-parallel voice conversion (VC) using variational autoencoders (VAEs). In the proposed framework, phonetic posteriorgrams (PPGs), which represent latent variables of phonetic contents, were introduced to the conventional VAE-based VC in order to improve the converted speech quality. The conventional VC using VAEs was also extended to many-to-many VC, which can convert arbitrary speakers' characteristics into another arbitrary speakers' ones. d-vectors, which represent characteristics of speakers as continuous vectors, were adopted to speaker representations for many-to-many VC, in addition to conventional speaker codes. Experimental results demonstrated that 1) PPGs successfully improved both naturalness and speaker similarity of the converted speech, and 2) both speaker codes and d-vectors were adopted to the VAE-based many-to-many non-parallel VC. In the future, we will further investigate the effect of the dimensionality of the d-vectors.

Acknowledgements: Part of this work was supported by JSPS KAKENHI Grant Number 16H06681.

6. REFERENCES

[1] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, Mar. 1988.

[2] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *Proc. ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 3893–3896.

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

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv*, vol. abs/1312.6114, 2013.

[5] T. Nakashika, T. Takiguchi, and Y. Minami, "Non-parallel training in voice conversion using an adaptive restricted boltzmann machine," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2032–2045, Nov. 2016.

[6] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Proc. APSIPA ASC*, Jeju, South Korea, Dec. 2016.

[7] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv*, vol. abs/1511.06349, 2016.

[8] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv*, vol. abs/1611.02648, 2016.

[9] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Proc. ICME*, Seattle, U.S.A., Jul. 2016.

[10] N. Hojo, Y. Ijima, and H. Mizuno, "An investigation of dnn-based speech synthesis using speaker codes," in *Proc. INTERSPEECH*, California, U.S.A., Sep. 2016, pp. 2278–2282.

[11] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 4080–4084.

[12] H.-T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, "Adapting and controlling dnn-based speech synthesis using input codes," in *Proc. ICASSP*, New Orleans, U.S.A., Mar. 2017, pp. 1905–1909.

[13] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. NIPS*, Montreal, Canada, Dec. 2014, pp. 3581–3589.

[14] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *arXiv*, vol. abs/1701.02720, 2017.

[15] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 5115–5119.

[16] H. Kawahara, I. Masuda-Katsuse, and A. D. Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3–4, pp. 187–207, Apr. 1999.

[17] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, Istanbul, Turkey, June 2000, pp. 1315–1318.

[18] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, Lauderdale, U.S.A., Apr. 2011, pp. 315–323.

[19] 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, Jul. 2011.