INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Group Latent Embedding for Vector Quantized Variational Autoencoder in Non-Parallel Voice Conversion

Shaojin Ding and Ricardo Gutierrez-Osuna

Department of Computer Science and Engineering, Texas A&M University, USA

{shjd, rgutier}@tamu.edu

Abstract

This paper proposes a Group Latent Embedding for Vector Quantized Variational Autoencoders (VQ-VAE) used in non-parallel Voice Conversion (VC). Previous studies have shown that VQ-VAE can generate high-quality VC syntheses when it is paired with a powerful decoder. However, in a conventional VQ-VAE, adjacent atoms in the embedding dictionary can represent entirely different phonetic content. Therefore, the VC syntheses can have mispronunciations and distortions whenever the output of the encoder is quantized to an atom representing entirely different phonetic content. To address this issue, we propose an approach that divides the embedding dictionary into groups and uses the weighted average of atoms in the nearest group as the latent embedding. We conducted both objective and subjective experiments on the non-parallel CSTR VCTK corpus. Results show that the proposed approach significantly improves the acoustic quality of the VC syntheses compared to the traditional VQ-VAE (13.7% relative improvement) while retaining the voice identity of the target speaker.

Index Terms: non-parallel voice conversion, variational autoencoder, group latent embedding

1. Introduction

Voice conversion (VC) aims to transform an utterance from a source speaker as if a target speaker had produced it. VC finds use in many applications, such as personalized text-to-speech synthesis [1], speaker spoofing [2] and pronunciation training [3]. Various approaches have been proposed to perform VC. Gaussian Mixture Models (GMM) [4, 5], Deep Neural Networks (DNN) [6-9], and sparse representations [10, 11] are widely used and can achieve convincing results. Within DNN based methods, various network architectures [6-9] have been explored, and more recently Variational Autoencoders (VAE) [12-14]. A VAE consists of an encoder network and a decoder network. In training, the encoder network learns a speaker-independent latent embedding from input speech signals, and the decoder reconstructs the input speech signals given the latent embedding and the corresponding speaker embedding. At runtime, VC is achieved by replacing the speaker embedding with that of a target speaker.

VC approaches based on VAE have several advantages: they do not require parallel corpora and time alignment during training [12], and they can be generalized to unseen speakers [15] and even cross-lingual scenarios [16]. However, speech generated by VAE has lower quality than that produced by GMM and DNN systems trained on parallel corpora [15]. The main reason is that the assumed prior distribution of the latent variables is too simple — often a single Gaussian. The resulting latent embedding is often overly simplified and poorly represents the underlying structure of the data (i.e., the over-regularization problem [17-19]). Several studies have addressed this problem. In [20, 21], the authors extended the prior distribution of latent variables from a single Gaussian to a Gaussian Mixture. Nonetheless, the over-regularization effect still exists [20], and the latent embedding tends to be ignored when it is paired with a powerful autoregressive decoder [13]. A VQ-VAE [13] solves the above two problems by learning the prior through an embedding dictionary rather than using a pre-defined static distribution. However, there is no constraint on the embedding dictionary, and therefore adjacent atoms can represent entirely different phonetic contents. As a result, the VC speech can have mispronunciations and distortions whenever the output of the encoder is quantized to an atom representing entirely different phonetic content.

To address this problem, we propose a Group Latent Embedding for a VQ-VAE that enforces atoms with similar phonetic content to be close to each other but those with different content to be away from each other. Instead of using the nearest atom from the embedding dictionary as the latent embedding, we further divide the embedding dictionary into groups. During training, given the output of the encoder, we first select its nearest group in the embedding dictionary, and then we use the weighted average of the atoms within the group as the latent embedding. The weights are reversely proportional to the distance between the output of the encoder and the atoms. During the back-propagation step, instead of updating only one atom in the embedding dictionary at a time, we update all the atoms in the group in proportion to their weights. Finally, we pass the latent embedding to the decoder and condition the decoder on a corresponding speaker embedding, which outputs the reconstruction of the input. At runtime, VC is performed by replacing the speaker embedding with that of the target speaker. We conduct both objective and subjective experiments on the CSTR VCTK Corpus [22] to evaluate the proposed approach. Results show that the proposed method can improve both Mel-Cepstral Distortion and the acoustic quality of the VC syntheses without sacrificing the voice identity of the target speaker. In a final analysis, we demonstrate the group structure of the atoms in the embedding dictionary.

2. Literature review

Most conventional VC systems, such as those based on GMMs, DNNs, and sparse representations, require time-aligned parallel corpora. GMM-based methods [4, 5] learn the joint distribution of source and target spectral features and then estimate the target spectral features through least-squares regression. DNN-based methods map the source spectral features directly into the target space through various network structures such as restricted Boltzmann machines [6], auto-encoders [7], feed-forward neural networks [8], and recurrent neural networks [9]. Sparse representation methods [10, 11] first build exemplar dic-

http://dx.doi.org/10.21437/Interspeech.2019-1198

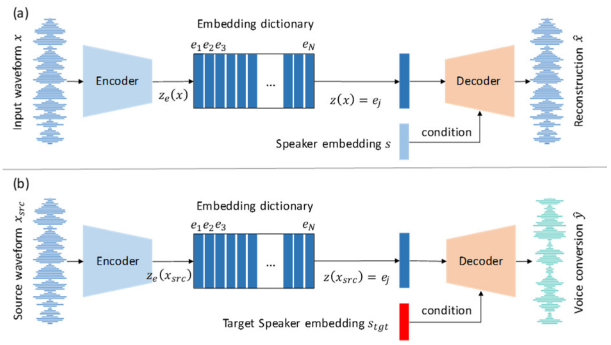

Figure 1: *The framework of using VQ-VAE in voice conversion. (a) Training (b) Testing.*

tionaries for a source and a target speaker. At runtime, they use sparse coding to extract a speaker-independent code from the source speech and then combine it with the target dictionary to generate VC speech.

To avoid the laborious process of collecting parallel corpora, several non-parallel VC techniques have been proposed in recent years. These include the INCA algorithm [23], DNNs [24, 25], sparse representations [26], and phonetic posteriorgrams [27]. More recently, Hsu *et al.* [12, 28] proposed to use a VAE for VC. Following this, Kameoka et al. [29] and Huang *et al.* [30] improved the quality of the VC syntheses by using auxiliary classifier and WaveNet vocoder [31] adaption, respectively. Additionally, Hsu *et al.* [14] proposed a Factorized Hierarchical VAE (FHVAE) to separate the linguistic and speaker representations, and Mohammadi [16] applied FHVAE to cross-lingual VC. Saito *et al.* [15] alleviated the VC quality degradation caused by the over-regularization problem of VAE through adding a phonetic posteriorgram to the input of the decoder. They also generalized it to unseen speakers by using the d-vector [32] of these speakers. Oord *et al.* [13] proposed VQ-VAE to address the over-regularization and paired with WaveNet [33] decoder to achieve waveform-wise high-quality voice conversion.

3. Method

In this section, we first introduce the general framework of using VQ-VAE [13] for VC. Following this, we describe the proposed Group Latent Embedding.

3.1. VC using VQ-VAE

Figure 1 illustrates the overall process of using VQ-VAE in VC. A VQ-VAE has an encoder-decoder network structure and an embedding dictionary $e \in \mathbb{R}^{N \times D}$, where N is the number of atoms and D is the dimensionality of each atom. Given an non-parallel corpus of multiple speakers, the inputs to the network are pairs of audio segment x and the corresponding speaker embedding s. During training, x is passed to an encoder network $\boldsymbol{E}$, which produces the output $z_e(x) = \boldsymbol{E}(x)$. Then, the latent embedding $z(x)$ is computed by finding the nearest (Euclidean distance) atom in the embedding dictionary as,

$$z(x) = e_j, \qquad \text{where } j = \underset{j}{\operatorname{argmin}} \|z_e(x) - e_j\|_2 \tag{1}$$

where e_j is the j-th atom in the embedding dictionary, and $j \in \{1, 2, \ldots, N\}$. Finally, the latent embedding $z(x)$ and the speaker embedding s are passed into a decoder network $\boldsymbol{D}$ to produce the reconstruction of the input $\hat{x} = \boldsymbol{D}(z(x), s)$.

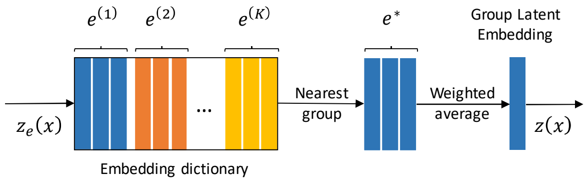

Figure 2: *The diagram of Group Latent Embedding.*

To learn the parameters of the model, we minimize the training objective:

$$\begin{aligned} L = &-\log(x|z(x), s) + \|sg[z_e(x)] - e_j\|_2^2 \\ &+\beta\|z_e(x) - sg[e_j]\|_2^2 \end{aligned} \tag{2}$$

where $-\log(x|z(x), s)$ is the negative log likelihood term optimizing the encoder and decoder, $\|sg[z_e(x)] - e_j\|_2^2$ is used to update the embedding dictionary, and $\beta\|z_e(x) - sg[e_j]\|_2^2$ is a commitment loss to guarantee that the encoder also updates according to the selected atom. $sg[\bullet]$ denotes the stop-gradient operator, which is defined to be identity during forward-propagation but zero partial derivatives during back-propagation. e_j is the nearest atom of $z_e(x)$, and β is the coefficient of the commitment loss.

At runtime, an audio waveform from the source speaker x_{src} and the speaker embedding of the target speaker s_{tgt} are passed to the network, and the VC waveform is generated as,

$$\hat{y} = \boldsymbol{D}(z(x_{src}), s_{tgt}) \tag{3}$$

3.2. Proposed method: Group Latent Embedding

In a conventional VQ-VAE, there is no constraint on the distribution of the atoms in the embedding dictionary, so adjacent atoms can represent entirely different phonetic content. To alleviate the mispronunciations and distortions caused by this incorrect quantization, we propose a Group Latent Embedding (GLE) inspired by group sparse coding algorithms [34, 35]. With GLE, we wish to promote that similar atoms (i.e., those with similar phonetic content) be close to each other, and that different atoms be away from each other. Thus, we can reduce the chances of quantizing the output of the encoder to a mismatched atom.

The diagram of the GLE is shown in Figure 2. We divide the embedding dictionary $e \in \mathbb{R}^{N \times D}$ into K sub-dictionaries, and we denote the k-th sub-dictionary as $e^{(k)} \in \mathbb{R}^{M \times D}$, where $N = M \times K$. During forward-propagation, we first find the nearest sub-dictionary e^* for $z_e(x)$:

$$e^* = e^{(k)}, \qquad \textit{where } k = \underset{k}{\operatorname{argmin}}\, d(z_e(x), e^{(k)}) \tag{4}$$

where $d(z_e(x), e^{(k)})$ is the distance from $z_e(x)$ to the k-th sub-dictionary, defined as the average distance over all the atoms in it:

$$d(z_e(x), e^{(k)}) = \frac{1}{M}\sum_{j=1}^{M}\left\|z_e(x) - e_j^{(k)}\right\|_2 \tag{5}$$

Within e^*, the latent embedding is computed as the weighted average of the atoms in the sub-dictionary:

$$z(x) = \frac{\sum_{j=1}^{M} w_j e_j^*}{\sum_{j=1}^{M} w_j} \tag{6}$$

where w_j is the weight for the j-th atom. w_j is inversely proportional to the distance between $z_e(x)$ and e_j^*, as in eq. (7).

$$w_j = \frac{1}{\left\|z_e(x) - e_j^*\right\|_2} \tag{7}$$

Since we used the weighted average of the atoms in a group as the latent embedding during the forward-propagation, we will need to update all these atoms in the back-propagation. As a result, the training objective becomes eq. (8).

$$L = -\log(x|z(x), s) + \left\|sg[z_e(x)] - \frac{\sum_{j=1}^{M} w_j e_j^*}{\sum_{j=1}^{M} w_j}\right\|_2^2 + \beta\left\|z_e(x) - sg\left[\frac{\sum_{j=1}^{M} w_j e_j^*}{\sum_{j=1}^{M} w_j}\right]\right\|_2^2 \tag{8}$$

3.3. Network architecture

The proposed Group Latent Embedding can be used with any encoder and decoder structure. To enable a fair comparison, we use an encoder-decoder architecture similar to that in [13]. The encoder contains 10 one-dimensional convolutional layers. Six of them are with kernel size of 4 and stride of 2, and the other 4 are with kernel size of 4 and stride 1. The decoder has 3 one-dimensional convolutional layers with kernel size of 4 and stride of 4, 3 RNN layers with GRU [36] cells, followed by a WaveRNN [37] module. We do not use WaveNet [33] (as in [13]), since WaveRNN is much more efficient (~10 times faster in generating an audio waveform, as shown in [37]) without a loss of acoustic quality. The number of channels of all the convolutional layers and RNN layers is 128, and that of WaveRNN is 896. Additionally, we use one-hot vectors for speaker embedding for simplicity, but the model can be generalized to unseen speakers using other speaker embeddings such as i-vectors [38], x-vectors [39], and d-vectors [32].

4. Experimental setup

4.1. Corpus

We conducted experiments on a non-parallel corpus, CSTR VCTK Corpus [22], which consists of 109 English speakers with several accents (e.g., English, American, Scottish, Irish, Indian, etc.). For each speaker, there are ~300 utterances, and part of them have the same linguistic contents across all the speakers. In our experiments, we used a subset of the corpus containing all the utterances for the first 30 speakers (p225-p256, with p235 and p242 missing). We selected four speakers for evaluation: p225 (Female), p226 (Male), p229 (Female), and p232 (Male). All four speakers have an English accent. For each of them, we used the first 30 utterances as the testing set, and we considered four VC directions: p225 to p226 (F-M), p226 to p232 (M-M), p232 to p229 (M-F), and p229 to p225 (F-F). All the results are averaged over these four VC directions. We use the other utterances from these 4 speakers and all the utterances from the other 26 speakers for training, or 11,533 utterances in total.

4.2. Implementation details

We implemented the proposed approach in PyTorch[1] [40]. First, we down-sampled the waveforms from 48kHz to 22.05kHz. Then, we randomly selected segments of 1,024 samples from the original waveform and used the segments as inputs to the network. We also did zero-padding of the inputs to guarantee them to be compatible with the encoder and decoder. Furthermore, we augmented the data by adding random noise and randomly shifted inputs to the left or right to enhance translational invariance.

We set the number of groups in the latent embedding to 41, corresponding to the number of phonemes in the CMU ARCTIC corpus [41]. For the sub-dictionary of each group, we set the number of atoms to 10, for a total of 410 atoms in the embedding dictionary. At each iteration, we normalized the 128-dimensional atoms to have unit norm after updating them. We set $\beta = 0.25$ following [13]. As we have 30 speakers in the training set, the speaker embedding was a 30-dimensional one-hot vector.

Our model was trained on an Nvidia GTX 1080Ti GPU with a batch size of 48. We used Adam optimizer with a base learning rate of 10^{-4}. The model converged after 3,000 epochs, and the entire training time was around 120 hours.

5. Results

We evaluated the VC performance of the proposed method (**GLE**) through a set of objective and subjective evaluations. We also compared the proposed method against two baselines:

VQ-VAE: the original Vector Quantized Variational Autoencoder [13]. To ensure a fair comparison, we used the same number of atoms in the embedding dictionary (410 atoms) as well as the same encoder and decoder network structure.

PPG-GMM: a state-of-the-art GMM-based non-parallel VC framework [27]. The approach extracts a phonetic posteriorgram (PPG) for each frame in the source and target speaker corpus. Then, it generates pairs of source-target frames by computing the similarity of their respective PPGs. Finally, the paired speech frames are used to train a GMM. We set the number of mixture components to 41, the same as the number of groups in our proposed method. We also applied Maximum Likelihood Parameter Generation (MLPG) and Global Variance (GV) compensation [5] based on static+delta features to maximize VC performance.

5.1. Objective evaluation

We evaluated the proposed method objectively by computing the Mel-Cepstral Distortion (MCD) between the converted speech and the time-aligned target speech. Since computing MCD requires the ground-truth target speech, we selected 29 utterances that have the same linguistic content from the testing set and used dynamic time warping [42] to align the converted speech and the target ground-truth[2]. Following this, we used the WORLD vocoder [43] (D4C edition [44]) to extract a 513-dimensional spectrogram and 25-dimensional Mel-cepstrum. We finally used the Mel-cepstrum to compute the MCD. Results are shown in Figure 3 (a). The proposed GLE achieved the lowest MCD (7.56) and outperformed both baseline systems (VQ-VAE: 8.43, 10.3% relative improvement; PPG-GMM: 8.57, 11.8% relative improvement).

[1] Our implementation and audio samples are available at https://github.com/shaojinding/GroupLatentEmbedding

[2] Note that this is done for evaluation purposes. Our proposed technique does not require parallel corpora.

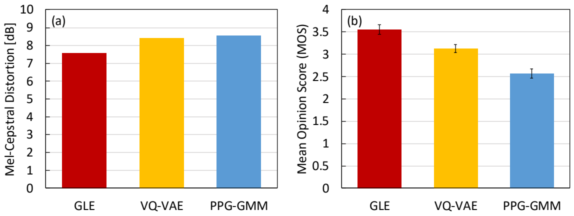

Figure 3: *(a) Average MCD of the proposed method (GLE) and baselines (VQ-VAE, PPG-GMM). (b) Acoustic quality results with 95% confidence intervals.*

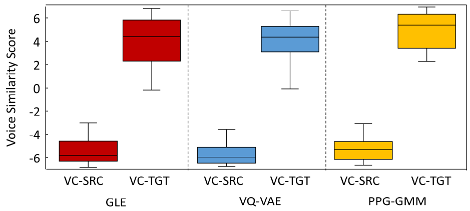

Figure 4: *Voice identity results of the proposed method (GLE) and baselines (VQ-VAE, PPG-GMM). VC-SRC: VSS between VC and the source speaker; VC-TGT: VSS between VC and the target speaker.*

5.2. Subjective evaluation

To provide a subjective evaluation of the proposed method, we conducted two listening tests on Amazon Mechanical Turk. First, we measured acoustic quality with a 5-point Mean Opinion Score (MOS) test. Second, we measured speaker identity with a Voice Similarity Score (VSS) test ranging from -7 (definitely different speakers) to +7 (definitely the same speaker) [45]. Participants were required to reside in the U.S. and pass a pre-test that asked them to identify different regional accents in the United States.

Mean opinion Score. Sixteen participants rated 72 utterances from the three VC systems: 20 utterances per system, plus 12 calibration utterances to detect if participants were cheating [46]. We excluded ratings of the calibration utterances from the data analysis. Figure 3 (b) shows the MOS results with 95% confidence intervals. The proposed method (GLE) was rated to have a 3.56 MOS, which is higher than the two baselines with statistical significance: VQ-VAE (3.13 MOS; 13.7% relative improvement; single-tail t-test, $p \ll 0.001$) and PPG-GMM (2.57 MOS; 38.5% relative improvement; single-tail t-test, $p \ll 0.001$).

Voice Similarity Score. Seventeen participants rated 108 utterance pairs: 32 pairs (16 VC-SRC pairs and 16 VC-TGT pairs) for each of the three systems, and 12 calibration utterances. For each utterance pair, participants were required to decide whether the two utterances were from the same speaker and then rate their confidence in the decision on a 7-point scale. Following [45], VSS was computed by collapsing the above two fields into a 14-point scale. Figure 4 shows the results of the VSS test. Participants were quite confident that the GLE syntheses and the source speech are produced by different speakers (VSS: -5.51), and that GLE syntheses and the target speech are produced by the same speaker (VSS: 4.44). We found no statistical significance between GLE and the two baselines (VC-SRC VSS, $p \gg 0.05$; VC-TGT VSS, $p \gg 0.05$). Additionally, we noticed that the confidence level for VC-TGT pairs is slightly lower than that for VC-SRC for the three systems. A possible explanation is that the VC syntheses still have part of the source speaker's segmental characteristics, which is an important cue of voice identity. As a result, it discourages participants from rating VC and the ground-truth target speech as being from the same speaker.

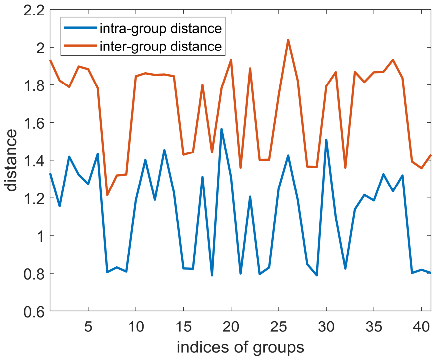

Figure 5: *Intra-group distances and inter-group distances of all 41 groups in Group Latent Embedding.*

6. Discussion

Our experiments show that the proposed system can improve the MCD and MOS of VCs without sacrificing the voice identity of the target speaker. This is achieved by adding a Group Latent Embedding to a conventional VQ-VAE. We analyzed the time complexity of the proposed method (based on a single GTX 1080Ti GPU and batch size of 48). During training, GLE has slightly higher time complexity (1.45 iterations per sec) than the original VQ-VAE (1.60 iterations per sec). At runtime, GLE took around 20 sec to synthesize one sec of 22.05 kHz speech (~1000 samples per second). We did not find significant differences between GLE and VQ-VAE in terms of synthesis time.

Finally, we analyzed the effect of GLE on the distribution of atoms in the embedding dictionary. For this purpose, we computed two measures per group: 1) the average distance between the group's mean and all the atoms within the group (intra-group distance); and 2) the average distance between the group's mean and the means of the remaining groups (inter-group distance). Figure 5 shows these two distances for all 41 groups. In all cases, the intra-group distance is significantly lower than the inter-group distance (an average of 1.12 vs. 1.68). This observation matches our goal that GLE can enforce similar atoms to be close to each other and different atoms to be away from each other.

7. Conclusions

We proposed a Group Latent Embedding for a VQ-VAE in non-parallel VC. Namely, we first divided the embedding dictionary into groups and then used the weighted average of the atoms in a group as the latent embedding. Both objective and subjective experimental results indicate that the proposed method improves the MCD and MOS of VC syntheses (compared to two state-of-the-art baselines) while retaining the target speakers' voice identity.

8. Acknowledgement

This work was supported by NSF awards 1619212 and 1623750.

9. References

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

[2] Z. Wu and H. Li, "Voice conversion and spoofing attack on speaker verification systems," in *APSIPA*, 2013, pp. 1-9.

[3] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech communication,* vol. 51, pp. 920-932, 2009.

[4] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing,* vol. 6, pp. 131-142, 1998.

[5] 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, pp. 2222-2235, 2007.

[6] 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 Transactions on Audio, Speech and Language Processing (TASLP),* vol. 22, pp. 1859-1872, 2014.

[7] S. H. Mohammadi and A. Kain, "A Voice Conversion Mapping Function Based on a Stacked Joint-Autoencoder," in *INTERSPEECH*, 2016, pp. 1647-1651.

[8] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 18, pp. 954-964, 2010.

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

[10] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion in noisy environment," in *SLT*, 2012, pp. 313-317.

[11] Z. Wu, E. S. Chng, and H. Li, "Exemplar-based voice conversion using joint nonnegative matrix factorization," *Multimedia Tools and Applications,* vol. 74, pp. 9943-9958, 2015.

[12] 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 *APSIPA*, 2016, pp. 1-6.

[13] A. van den Oord and O. Vinyals, "Neural discrete representation learning," in *NIPS*, 2017, pp. 6306-6315.

[14] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *NIPS*, 2017, pp. 1878-1889.

[15] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *ICASSP*, 2018, pp. 5274-5278.

[16] S. H. Mohammadi and T. Kim, "Investigation of using disentangled and interpretable representations for one-shot cross-lingual voice conversion," in *Interspeech*, 2018, pp. 2833–2837.

[17] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349,* 2015.

[18] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "How to train deep variational autoencoders and probabilistic ladder networks," in *ICML*, 2016.

[19] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improving variational inference with inverse autoregressive flow," in *NIPS*, pp. 4743–4751.

[20] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran*, et al.*, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv preprint arXiv:1611.02648,* 2016.

[21] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," *arXiv preprint arXiv:1611.05148,* 2016.

[22] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit," 2017.

[23] D. Erro, A. Moreno, and A. Bonafonte, "INCA algorithm for training voice conversion systems from nonparallel corpora," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 18, pp. 944-953, 2010.

[24] T. Nakashika, T. Takiguchi, Y. Minami, 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 (TASLP),* vol. 24, pp. 2032-2045, 2016.

[25] F.-L. Xie, F. K. Soong, and H. Li, "A KL Divergence and DNN-Based Approach to Voice Conversion without Parallel Training Sentences," in *Interspeech*, 2016, pp. 287-291.

[26] C. Liberatore, S. Aryal, Z. Wang, S. Polsley, and R. Gutierrez-Osuna, "SABR: Sparse, Anchor-Based Representation of the Speech Signal," in *INTERSPEECH*, 2015.

[27] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent Conversion Using Phonetic Posteriorgrams," *ICASSP,* 2018.

[28] 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," *arXiv preprint arXiv:1704.00849,* 2017.

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

[30] W.-C. Huang, Y.-C. Wu, H.-T. Hwang, P. L. Tobing, T. Hayashi, K. Kobayashi*, et al.*, "Refined WaveNet Vocoder for Variational Autoencoder Based Voice Conversion," *arXiv preprint arXiv:1811.11078,* 2018.

[31] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-Dependent WaveNet Vocoder," in *INTERSPEECH*, 2017, pp. 1118-1122.

[32] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *ICASSP*, 2014, pp. 4052-4056.

[33] A. V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves*, et al.*, "WaveNet: A Generative Model for Raw Audio," 2016.

[34] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 68, pp. 49-67, 2006.

[35] Z. Szabó, B. Póczos, and A. Lőrincz, "Online group-structured dictionary learning," in *CVPR*, 2011, pp. 2865-2872.

[36] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk*, et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078,* 2014.

[37] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart*, et al.*, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435,* 2018.

[38] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey*, 2010.

[39] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329-5333.

[40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito*, et al.*, "Automatic differentiation in pytorch," 2017.

[41] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[42] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*, 1994, pp. 359-370.

[43] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems,* vol. 99, pp. 1877-1884, 2016.

[44] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication,* vol. 84, 2016.

[45] D. Felps and R. Gutierrez-Osuna, "Developing objective measures of foreign-accent conversion," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 18, 2010.

[46] S. Buchholz and J. Latorre, "Crowdsourcing preference tests, and how to detect cheating," in *INTERSPEECH*, 2011.