Interspeech 2018
2-6 September 2018, Hyderabad

Voice Conversion Across Arbitrary Speakers based on a Single Target-Speaker Utterance

Songxiang Liu[1], *Jinghua Zhong*[1], *Lifa Sun*[1,2], *Xixin Wu*[1], *Xunying Liu*[1] *and Helen Meng*[1]

[1]Human-Computer Communications Laboratory
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China
[2]SpeechX Limited, Shenzhen, China

{sxliu, jhzhong, lfsun, wuxx, xyliu, hmmeng}@se.cuhk.edu.hk

Abstract

Developing a voice conversion (VC) system for a particular speaker typically requires considerable data from both the source and target speakers. This paper aims to effectuate VC across arbitrary speakers, which we call any-to-any VC, with only a single target-speaker utterance. Two systems are studied: (1) the i-vector-based VC (IVC) system and (2) the speaker-encoder-based VC (SEVC) system. Phonetic PosteriorGrams are adopted as speaker-independent linguistic features extracted from speech samples. Both systems train a multi-speaker deep bidirectional long-short term memory (DBLSTM) VC model, taking in additional inputs that encode speaker identities, in order to generate the outputs. In the IVC system, the speaker identity of a new target speaker is represented by i-vectors. In the SEVC system, the speaker identity is represented by speaker embedding predicted from a separately trained model. Experiments verify the effectiveness of both systems in achieving VC based only on a single target-speaker utterance. Furthermore, the IVC approach is superior to SEVC, in terms of the quality of the converted speech and its similarity to the utterance produced by the genuine target speaker.

Index Terms: voice conversion, i-vector, speaker encoder, low-resource deployment

1. Introduction

The goal of voice conversion (VC) is to modify a speech signal uttered by a source speaker to sound as if it was uttered by a target speaker, without changing the linguistic content. Various methods have been proposed for VC. Gaussian Mixture Models (GMMs) have been used for VC to develop weighted linear conversion functions mapping the source-target feature vectors [1, 2]. Other methods, including kernel partial least squares regression [3], frequency warping [4, 5] and neural networks [6, 7, 8, 9, 10, 11] have also been studied. Developing a VC system for a particular speaker typically requires considerable parallel data between the source and target speakers. Many techniques have been studied to perform VC when only non-parallel data is available. In [12, 13], the INCA-based algorithms were proposed to iteratively seek frame-wise alignment between non-parallel source and target utterances, where the VC performance may degrade due to the inaccurate alignment. Another approach is to train VC models with available speech data from other speakers and then adapt to the desired target speaker. In [14, 15], a maximum a posterior (MAP) method was used to adapt a source GMM with target data, while in [16], the eigenvoice approach was introduced into VC. These methods still require parallel data from other speakers. Recently, adaptive restricted boltzmann machine [17], variational autoencoder [18], and phonetic posteriorgram (PPG)-based methods [19, 20] have been proposed to achieve parallel free VC.

Furthermore, VC systems are often deployed for a specific target speaker. When a new target speaker comes along, a VC model needs to be either adapted from a pre-trained model or built entirely from scratch with speech data from the new target speaker. Adaptation and new model development is hard to achieve when only limited speech data (e.g., one utterance) is available from that target speaker. One way to tackle this low-resource deployment problem is to train a multi-speaker VC (MSVC) model with available multi-speaker speech corpus, where speaker identities (speakerIDs) are encoded as additional model inputs. For a new source-target speaker pair, the trained MSVC model takes in linguistic features extracted from the source utterance and additional input encoding the target speakerID to get the target spectral features. Different ways to encode speakerIDs have been studied. In [21], i-vectors are employed, which are low-dimensional speaker specific vectors, as additional inputs to encode speakerIDs when training an average voice model (AVM) with a multi-speaker speech corpus. In text-to-speech synthesis, learnable speaker embeddings have been adopted as additional inputs to encode speakerIDs [22, 23].

This paper focuses on performing VC across arbitrary speakers, which we call any-to-any VC, using only a single target speaker's utterance. We adopt the PPG-based VC method [19], where PPGs are speaker-independent linguistic features and can be extracted from the same utterance with the spectral features. Hence, we can train the model using completely non-parallel data. We train deep bidirectional long-short term memory (DBLSTM)-based MSVC models, which take in additional inputs encoding speakerIDs, with a multi-speaker speech corpus. We compare the use of i-vectors and learnable speaker embeddings to encode speakerIDs. While i-vectors for a new target speaker are extracted from a pre-trained offline i-vector extractor from the target speaker's utterances, the embedding of that speaker cannot be obtained directly from the learned speaker embedding table. Inspired by [24], we train a separate model, called speaker encoder, with learned speaker embeddings and then estimate the speaker embedding of a new target speaker using her/his utterances. The any-to-any VC systems explored in this paper have several advatages:

- They can achieve VC across a new source-target speaker pair using only one target-speaker utterance. The converted speech has acceptable quality and similarity compared with the ground-truth target utterances.
- They have no adaptation procedure, meaning that the

10.21437/Interspeech.2018-1504

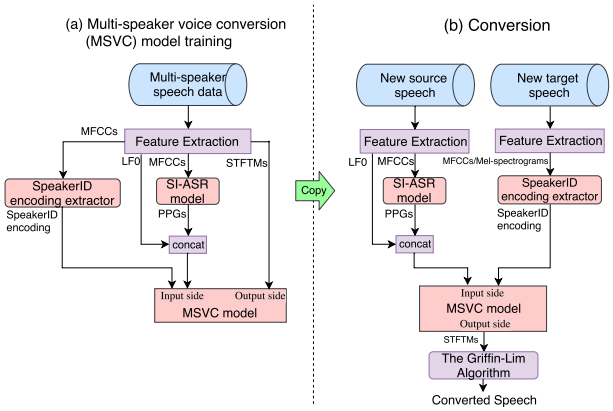

Figure 1: *Schematic diagram of an any-to-any voice conversion system.*

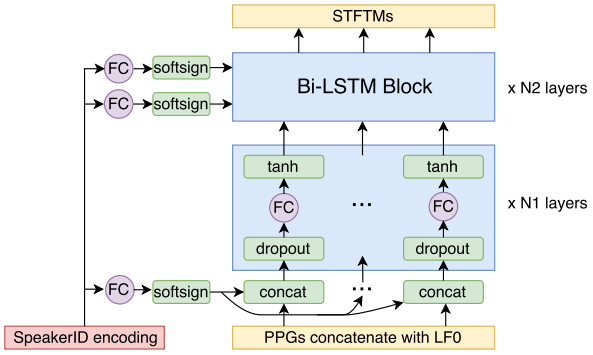

Figure 2: *Schematic diagram of the multi-speaker voice conversion (MSVC) model in Figure 1.*

trained systems can be used for new source-target speakers directly.

- They have no requirement for parallel data during training, meaning that any available speech corpus can be used for model training.

The rest of the paper is organized as follows: Section 2 introduces the any-to-any VC systems. Section 3 describes the experimental setup. Section 4 presents the evaluation results and Section 5 concludes this paper.

2. Any-to-Any Voice Conversion Systems

We propose two different systems to achieve any-to-any VC: (1) The i-vector-based VC (IVC) system and (2) the speaker-encoder-based VC (SEVC) system. Both systems train a DBLSTM-based MSVC model. The IVC system uses i-vectors to encode speakerIDs, while the SEVC system uses learnable speaker embeddings to encode speakerIDs.

2.1. The I-vector-based VC System

2.1.1. The I-vector Extractor

The i-vector, a low-dimensional vector, has proven to be the most successful speaker representation for speaker recognition. In this work, we employ the classical GMM i-vector approach [25] as i-vector extractor. It compresses both channel and speaker information into a low-dimensional space called total variability space, and accordingly projects each GMM supervector to a total factor feature vector called the i-vector. Given features of N utterances and N_u frames for the u-th utterance, $\{x_i^{(u)}\}_{\{i=1,\ldots,N_u;u=1,\ldots,N\}}$, F is the dimension of each frame, the i-th speech frame $x_i^{(u)}$ from the the u-th utterance is assumed to be generated by the following Gaussian distribution:

$$x_i^{(u)} \sim \sum_k \pi_k^{(u)} \mathcal{N}(m_k + T_k \omega^{(u)}, \Sigma_k) \quad (1)$$

where m_k and Σ_k is the mean and covariance of the k-th Gaussian in the universal background model (UBM), the T_k matrices describe a low-rank space (named total variability space) and $\omega^{(i)}$ is a low-dimensional total variability factor (named i-vector) with standard normal distribution. There are K Gaussian components in the UBM which is used as the class k in Eq. 1.

Since the i-vector contains both speaker and channel information, Linear Discriminant Analysis and Probabilistic LDA (PLDA) [26], good inter-session compensation methods, are applied to the i-vectors for dimension reduction.

2.1.2. Training and Conversion

The IVC system consists of two parts: MSVC model training part and conversion part.

Figure 1(a) shows the MSVC training process. The i-vector extractor introduced in Section 2.1.1 is used as the speakerID encoding extractor. The speaker-independent automatic speech recognition (SI-ASR) model has a DNN architecture, which is pre-trained by a standard ASR corpus. The MSVC model architecture is shown in Figure 2, which comprises of two parts: N_1 stacked fully connected (FC) layers and N_2 stacked bidirectional long-short term memory (Bi-LSTM) [27] layers. To fully condition on the speakerIDs, we incorporate the speakerIDs into multiple portions of the MSVC model. We map the speakerID encodings to higher-level representation with one FC layer before concatenating with PPGs and log-scale F0 (LF0). We produce the initial states of the stacked Bi-LSTM layers by distinct FC layers. Given speech samples from the multi-speaker training corpus, their Mel-frequency cepstral coefficients (MFCCs), LF0 and short-time Fourier transform magnitudes (STFTMs) are first extracted. The MFCCs are then fed into the pre-trained i-vector extractor and SI-ASR model to obtain the i-vectors and PPGs, respectively. We use the i-vectors obtained as speakerID encodings. Since PPGs are speaker-independent linguistic features which are free from prosodic information, we add LF0 to the input side of the MSVC model to compensate. Therefore, the speakerID encodings, PPGs and LF0 are fed into the MSVC model to drive out the STFTMs outputs, which are used to to compute a regression loss with the ground truth STFTMs. Then the model parameters are updated using the back-propagation through time (BPTT) algorithm.

The conversion process is shown in Figure 1(b). Given a pair of new source and target speakers, the source speech samples are used to extract MFCCs and LF0, while the target speech samples are used to obtain only MFCCs. The source MFCCs are then fed into the SI-ASR model to get PPGs, while the target MFCCs are applied to compute i-vector from the speakerID encoding extractor. Then the MSVC model is driven by the obtained PPGs, LF0 and the target speaker i-vector to get the predicted STFTMs. Finally, the converted speech is computed using the Griffin-Lim algorithm[28].

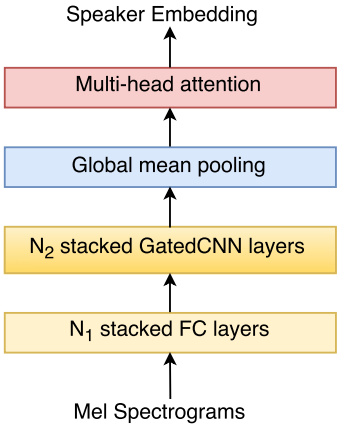

Figure 3: *Schematic diagram of the speaker encoder.*

2.2. The Speaker-Encoder-based VC System

2.2.1. Speaker Encoder

The speaker encoder is a regression model which is used to predict embedding of a speaker unseen during the training stage. We use a similar speaker encoder architecture to that proposed in [24]. As shown in Figure 3, the speaker encoder consists with four parts: N_1 stacked FC layers, N_2 stacked convolutional layers with Gated Linear Units (GatedCNN) [29], one global mean pooling layer and one multi-head attention layer [30]. The speaker encoder is trained using data from only the observed training speakers, where the Mel-spectrograms of audio samples are used as inputs, while the learned speaker embeddings are regarded as outputs.

2.2.2. Training and Conversion

The MSVC model, shown in Figure 2, adopts speaker embeddings as additional inputs encoding speakerIDs. The training process is as follows: (1) The MSVC model is first trained with a multi-speaker speech corpus, where the embedding parameters are appropriately initialized and then jointly trained with other model parameters by a regression loss. (2) We then use speech samples from the training speakers as inputs and the learned training speaker embeddings as outputs to train the speaker encoder.

As shown in Figure 1(b), where the trained speaker encoder is adopted as the speakerID encoding extractor, the conversion process is similar to that introduced in Section 2.1.2. The only difference is that the target speakerID is represented by an embedding estimated from the speaker encoder using Mel-spectrograms computed from speech signals of that target speaker.

3. Experiments

We compare the two approaches for any-to-any VC introduced in Section 2: (1) the IVC system and (2) the SEVC system. To explore whether more target-speaker utterances can improve the quality of the converted speech and its similarity to the speech uttered by the target speaker, we train three speaker encoders for use of 1, 5 and 10 target-speaker utterance(s) respectively. At the conversion stage, we feed 1, 5 or 10 target-speaker utterance(s) into the corresponding trained speaker encoder to estimate the target-speaker embedddings.

3.1. The SI-ASR Model and the I-vector Extractor Training

The SI-ASR model has a DNN architecture with 4 hidden layers containing 1024 hidden units. Senones are treated as the phonetic class of PPGs. The number of senone classes is 131, which are obtained by clustering at the SI-ASR training stage. Speech data of 462 speakers in the TIMIT corpus [31] is used to train the SI-ASR model. 13-dimensional MFCCs extracted using a 25-ms Hamming window with 5-ms shift are used as features. We implemented the SI-ASR model using the Kaldi speech recognition toolkit [32]. All speech samples used in this paper are resampled to 16kHz.

We train the i-vector extractor using the Wall Street Journal corpora (WSJ0+WSJ1) [33] and the TIMIT corpus, which contains 847 speakers in total. For the acoustic features in speaker modeling, the first 19 MFCCs and log energy are calculated, together with their first and second derivatives. The frame length is 25ms. Then energy-based voice-activity detection (VAD) and sliding-window cepstral mean and variance normalization (CMVN) are applied to remove non-speech frames and for feature normalization. The gender-independent 2,048 Gaussian UBMs, i-vector extractor, LDA and PLDA with whitening and length normalization are trained on all the model training data. The dimension of i-vector is set to 400. The rank of LDA and PLDA projection matrix is set to 200 and 32. So the final dimension of i-vector is 32.

3.2. MSVC Model Training

As shown in Figure 2, the IVC system and the SEVC system have the same MSVC model configuration. We use 2 FC layers with dropout (0.2) containing 512 hidden units. 4 Bi-LSTM layers with 512 hidden units are deployed, the outputs of which are then mapped to STFTMs by another FC layer. The FC layer before the concatenation operations has 132 hidden units, while the FC layers producing the initial states of the Bi-LSTM layers have 512 hidden units.

The VCTK corpus [34] is used for training the MSVC models. The VCTK consists of 108 English native speakers with various accents. There are parallel utterances between different speakers in VCTK. To leverage the benefit that the proposed systems have no requirement for parallel data, we only use non-parallel utterances during training. 96 speakers (40 males and 56 females) with 90 utterances each are used as training speakers and the remaining 12 speakers are used as evaluation speakers. STFTMs are computed with Hanning windowing, 25-ms window size, 5-ms window shift and 1024-point Fourier transform. Waveforms are pre-emphasized (0.97) before Fourier transform. We normalize PPGs, LF0, i-vectors and STFTMs to have zero mean and unit variance. We raise the predicted STFTMs by a power of 1.35 before feeding to the Griffin-Lim algorithm to reduce artifacts. The speaker embeddings have dimension of 32. We use L1 loss and Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [35] with learning rate 0.002.

3.3. Speaker Encoder

The speaker encoder has 2 FC layers with 64 hidden units, 2 layers of GatedCNN layers with 64 kernels and kernel width 12. The multi-head attention layer is applied with 2 heads and a unit size of 64 for keys, queries and values. Log-Mel spectrograms with 80 frequency bands are extracted from the training speech samples using Hanning windowing, 25-ms window size, 5-ms window shift and 1024-point Fourier transform. Log-Mel spectrograms are normalized to zero mean and unit variance be-

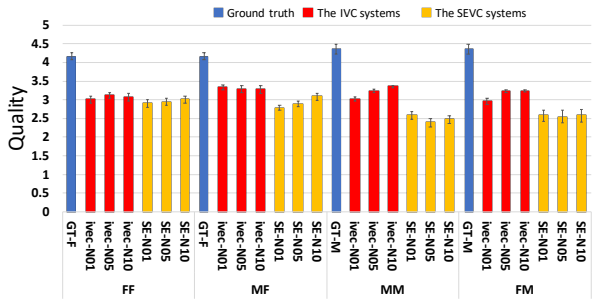

Figure 4: *Comparison of the MOS scores of the i-vector-based VC (IVC) systems and the speaker-encoder-based VC (SEVC) systems. FF denotes conversion from female to female.*

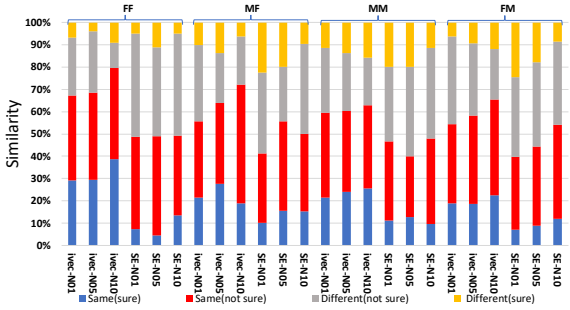

Figure 5: *Comparison of the similarity scores of the i-vector-based VC (IVC) systems and the speaker-encoder-based VC (SEVC) systems. FF denotes conversion from female to female.*

fore being fed into the speaker encoder, while the speaker embeddings are not normalized. 86 speakers (35 males and 51 females) from the training speakers are used to train the speaker encoder and the remaining 10 training speakers are used as validation set. Three speaker encoders are trained, which corresponds to 1, 5 and 10 target-speaker utterance(s). L1 loss is optimized by Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with minibatch size of 32 and initial learning rate of 0.001 with annealing rate of 0.5 applied every 2000 update steps.

4. Experimental Results

4.1. Subjective Evaluation Setup

We chose 4 speakers (2 females and 2 males) from the remaining 12 evaluation speakers, which are totally unseen during training. 1 female and 1 male speakers from the picked are used as target speakers, with the remaining 2 as source speakers. Utterances from each source speaker with text prompts different from the training samples are randomly chosen to perform the evaluation.

We conducted the standard 5-scale mean opinion score (MOS) test and 4-scale similarity test on Amazon Mechanical Turk platform.[1] In the MOS test, each group of stimuli contains the ground truth speech samples from the target speakers, which are randomly shuffled before being displayed to listeners. In the similarity test, converted speech samples are directly compared with the ground truth speech samples.

[1]Some audio samples can be found in "https://vcdemo.github.io"

4.2. Results and Analysis

The results of the MOS and similarity tests are shown in Figure 4 and Figure 5. In Figure 4, GT-F and GT-M stand for female and male ground truth speech samples, respectively. SE-N01 denotes the SEVC system using 1 target-speaker utterance to estimate speaker embedding, while ivec-N01 denotes the IVC system using 1 target-speaker utterance to extract i-vector and so on. The key observations from the results are as follows:

- Both the proposed IVC and SEVC systems can achieve VC across a new source-target speaker pair using only one target-speaker utterance. The converted speech has desirable quality and similarity.
- The IVC system is superior to the SEVC system in terms of the converted speech's quality and similarity.

Figure 4 shows that the IVC system outperforms the SEVC model consistently across all gender combinations in terms of converted speech's quality. Some gender combinations see improvement in converted speech quality when using more target-speaker utterances to extract i-vectors and to estimate speaker embeddings. As for the similarity performance of the converted speech, Figure 5 shows that the IVC system achieves higher similarity score than the SEVC system for all gender combinations. For both systems, the similarity score is higher when the target speaker is female. One possible reason is that more female speakers than the male speakers are used during training. It is expected that the gap will diminish when a more balanced training corpus is used. We see consistent improvement in similarity score when more speech samples are used to compute target speaker i-vectors and speaker embeddings. The rationality behind this is that the target speaker characteristics can be represented more appropriately when more target speech samples can be obtained. However, when trained with such small multi-speaker speech corpus, the IVC system is able to achieve acceptable conversion similarity with only one target-speaker utterance.

5. Conclusions

In this paper, we have proposed the IVC system and the SEVC system, which aim to effectuate VC across arbitrary speakers (refered as any-to-any VC). Experiments have verified the effectiveness of both systems in achieving any-to-any VC using only one target-speaker utterance. The IVC system is superior to the SEVC system in terms of the quality of the converted speech and its similarity to the utterance produced by the genuine target speaker. The two proposed systems have no adaptation procedure, which means that the trained systems can be used directly for new source-target speaker pairs. Moreover, the two proposed systems have no requirement for parallel data and are able to readily make use of available speech corpora for model training. The proposed systems are expected to achieve higher quality and similarity when larger speech corpora are used for training. Future work will include improving the proposed system by substituting the Griffin-Lim algorithm with better vocoders.

6. Acknowledgements

This project is partially supported by the General Research Fund from the Research Grants Council of Hong Kong SAR Government (Project No. 14208817).

7. 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, 1998.

[2] 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, 2007.

[3] E. Helander, H. Silén, T. Virtanen, and M. Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 3, pp. 806–817, 2012.

[4] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 922–931, 2010.

[5] X. Tian, Z. Wu, S. W. Lee, and E. S. Chng, "Correlation-based frequency warping for voice conversion," in *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*. IEEE, 2014, pp. 211–215.

[6] 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, no. 5, pp. 954–964, 2010.

[7] E. Azarov, M. Vashkevich, D. Likhachov, and A. A. Petrovsky, "Real-time voice conversion using artificial neural networks with rectified linear units." in *INTERSPEECH*, 2013, pp. 1032–1036.

[8] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 19–23.

[9] J. Nirmal, M. Zaveri, S. Patnaik, and P. Kachare, "Voice conversion using general regression neural network," *Applied Soft Computing*, vol. 24, pp. 1–12, 2014.

[10] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using rnn pre-trained by recurrent temporal restricted boltzmann machines," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 3, pp. 580–587, 2015.

[11] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4869–4873.

[12] 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, no. 5, pp. 944–953, 2010.

[13] Y. Agiomyrgiannakis, "The matching-minimization algorithm, the inca algorithm and a mathematical framework for voice conversion with unaligned corpora." in *ICASSP*, 2016.

[14] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu, "Voice conversion with smoothed gmm and map adaptation," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[15] C.-H. Lee and C.-H. Wu, "Map-based adaptation for speech conversion using adaptation data selection and non-parallel training," in *Ninth International Conference on Spoken Language Processing*, 2006.

[16] T. Toda, Y. Ohtani, and K. Shikano, "Eigenvoice conversion based on gaussian mixture model," 2006.

[17] 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, 2016.

[18] 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.

[19] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[20] L. Sun, H. Wang, S. Kang, K. Li, and H. M. Meng, "Personalized, cross-lingual tts using phonetic posteriorgrams." in *INTERSPEECH*, 2016, pp. 322–326.

[21] J. Wu, Z. Wu, and L. Xie, "On the use of i-vectors and average voice model for voice conversion without parallel data," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific*. IEEE, 2016, pp. 1–6.

[22] S. O. Arık, G. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," *arXiv preprint arXiv:1705.08947*, 2017.

[23] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," 2017.

[24] S. O. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," *arXiv preprint arXiv:1802.06006*, 2018.

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

[26] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proceedings Interspeech*, 2011, pp. 249–252.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[29] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[31] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus, 1993," *Linguistic Data Consortium, Philadelphia*.

[32] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[33] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[34] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.