Improving the Speaker Identity of Non-Parallel Many-to-Many Voice Conversion with Adversarial Speaker Recognition

Shaojin Ding, Guanlong Zhao, Ricardo Gutierrez-Osuna

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

{shjd, gzhao, rgutier}@tamu.edu

Abstract

Phonetic Posteriorgrams (PPGs) have received much attention for non-parallel many-to-many Voice Conversion (VC), and have been shown to achieve state-of-the-art performance. These methods implicitly assume that PPGs are speaker-independent and contain only linguistic information in an utterance. In practice, however, PPGs carry speaker individuality cues, such as accent, intonation, and speaking rate. As a result, these cues can leak into the voice conversion, making it sound similar to the source speaker. To address this issue, we propose an adversarial learning approach that can remove speaker-dependent information in VC models based on a PPG2speech synthesizer. During training, the encoder output of a PPG2speech synthesizer is fed to a classifier trained to identify the corresponding speaker, while the encoder is trained to fool the classifier. As a result, a more speaker-independent representation is learned. The proposed method is advantageous as it does not require pre-training the speaker classifier, and the adversarial speaker classifier is jointly trained with the PPG2speech synthesizer end-to-end. We conduct objective and subjective experiments on the CSTR VCTK Corpus under standard and one-shot VC conditions. Results show that the proposed method significantly improves the speaker identity of VC syntheses when compared with a baseline system trained without adversarial learning.

Index Terms: voice conversion, phonetic posteriorgram, speaker recognition, adversarial training

1. Introduction

Voice conversion (VC) aims to convert utterances from a source speaker to make it sound as if a target speaker had produced it. Conventional VC approaches [1, 2, 3, 4, 5] usually require training a model for each speaker pair using parallel corpora. Alternative approaches have emerged in recent years that do not require parallel corpora and can build a universal model for all pairs of speakers [6, 7, 8, 9, 10, 11, 12, 13]. Among these, the Phonetic-PosteriorGram-to-speech (PPG2speech) synthesizer [8, 9, 10, 13] has been shown to be effective for non-parallel many-to-many VC. The PPG2speech synthesizer is a sequence-to-sequence (seq2seq) model that transforms PPGs to speech features (e.g., Mel-spectrogram). The PPG2speech synthesizer has an encoder-decoder structure. During training, the encoder learns a speaker-independent hidden representation from input PPGs, and the decoder learns to generate the speech features given the hidden representation and the corresponding speaker embedding (e.g., i-vector [14], d-vector [15]). During inference, the PPG of a source speaker and the speaker embedding of a target speaker is used to produce VC syntheses.

The PPG2speech synthesizer assumes that the input PPGs represent the pronunciation of speech sounds in a speaker normalized space, which is speaker-independent and contains only linguistic information. In practice, however, PPGs still carry speaker identity information such as accent, intonation, and speaking rate [16] that can leak into the voice conversions.

In this work, we address this problem using adversarial learning. Namely, we propose a new training procedure that includes an *adversarial speaker classifier* jointly trained with the PPG2speech synthesizer. During training, the encoder output is fed into the adversarial speaker classifier, and the classifier is optimized to identify the corresponding speaker. At the same time, the encoder is optimized to fool the adversarial speaker classifier. As a result, the encoder outputs become more speaker-independent. The adversarial speaker classifier does not need to be pre-trained. Instead, it is jointly trained with the synthesizer end-to-end, and the minimax optimization in adversarial learning is achieved by back-propagation.

To evaluate the proposed adversarial learning system, we applied it to a state-of-the-art non-parallel many-to-many PPG2speech synthesizer based on Tacotron2 [17]. Then, we tested its effectiveness against the same PPG2speech synthesizer trained without adversarial learning. Using the CSTR VCTK Corpus [18], we conducted both objective and subjective experiments under two testing conditions: *standard* (test speakers were known during training) and *one-shot* (test speakers were unseen during training, and only look at a few of his/her utterances during inference). Results show that the proposed method can significantly improve the perceived speaker identity of the VC syntheses in both testing conditions.

2. Literature review

Conventional VC frameworks (e.g., based on GMMs [1], sparse representations [2, 3], and DNNs [4, 5]) require time-aligned parallel corpora in training. However, the size of parallel corpora is usually limited (e.g., 1 hour per speaker in the widely used CMU ARCTIC corpus [19]), and collecting parallel corpora can be laborious and expensive. To overcome this limitation, several non-parallel VC approaches have been proposed, such as the INCA algorithm [20], and various DNN architectures [16, 21, 22, 23, 24]. These methods avoid the use of parallel corpora, but they still require training a separate model for each pair of source-target speakers. To address this problem, serveral studies have proposed non-parallel many-to-many VC approaches based on Variational Autoencoders (VAE) [6, 7, 11, 25] and the PPG2speech synthesizer [8, 9, 10]. One-hot vectors are typically used as speaker embedding, due to its simplicity; several studies [8, 9, 10, 11] also explored the use of learned speaker embeddings (e.g., i-vector [14], d-vector [15]) to generalize to unseen speakers (i.e., one-shot VC).

PPGs have gained much recent attention for VC. Sun *et al*. [13] first proposed to use PPGs for one-to-one VC. In this work, they extracted PPGs from source speech using an acoustic model, and then trained a DNN to produce the converted speech from source PPGs. Miyoshi *et al*. [16] extended the previ-

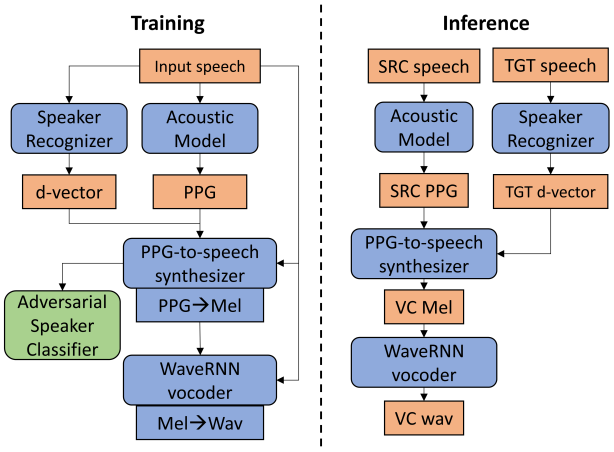

Figure 1: *Overall workflow of the proposed non-parallel many-to-many VC system.*

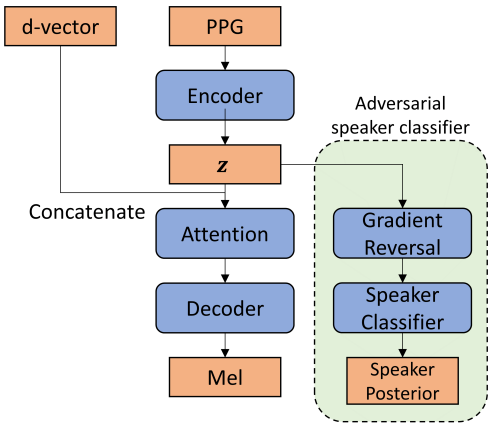

Figure 2: *PPG2speech synthesizer with adversarial speaker classifier. $\boldsymbol{z}$ denotes the hidden representation produced by the encoder. The adversarial speaker classifier is only used during training.*

ous PPG-based method with a sequence-to-sequence model that converted the context posterior probabilities, which improved the speaker identity of the converted speech. Zhou *et al*. [26] adopted bilingual PPGs for cross-lingual voice conversion. Liu *et al*. [8], Lu *et al*. [9], and Mohammadi *et al*. [10] extended the one-to-one PPG-based VC framework for many-to-many VC by conditioning on a speaker embedding.

Two previous studies [27, 12] explored the use of adversarial learning to disentangle linguistic and speaker representations in VC. Huang *et al*. [27] used a pre-trained speaker classifier in a VAE to reduce speaker information from the linguistic representations. Zhang *et al*. [12] achieved the same purpose using AEs by explicitly enforcing the distribution of the hidden representation from each speaker to be identical. **Our proposed method differs from these prior approaches in several aspects.** First, our adversarial learning algorithm has two advantages. Huang *et al*. [27] pre-trained the classifier and froze its weights during the training of the VC model. In contrast, our proposed method does not require the pre-training of the adversarial speaker classifier. Zhang *et al*. [12] used an explicit loss function for adversarial learning. In contrast, the speaker-independent hidden representation in our proposed method is implicitly learned through the minimax optimization. Second, these previous approaches have only been evaluated for *standard* conditions. In contrast, our study considers both *standard* and *one-shot* conditions, the latter being appealing for real-world applications since it requires little data from the target speaker.

3. Methods

Illustrated in Figure 1, our proposed VC system consists of four modules (highlighted in blue): a speaker-independent acoustic model to extract PPGs, a speaker recognition model to extract d-vectors as the speaker embeddings, a PPG2speech synthesizer to convert PPGs to Mel-spectrograms, and a final neural vocoder to generate a speech waveform from the Mel-spectrogram. First, we introduce a state-of-the-art PPG2speech synthesizer based on Tacotron2 [17] as a baseline system. Then, we describe the proposed adversarial learning approach.

3.1. Baseline method: PPG2speech synthesizer

Our system is based on the text-to-speech Tacotron2 model, which uses a seq2seq model to convert a text embedding sequence to a Mel-spectrogram. Tacotron2 has an encoder-decoder architecture. The encoder network contains three convolution layers and one Bidirectional Long Short Term Memory (LSTM) layer, which takes a text embedding as the input and produces a hidden representation. The output of the encoder is then fed to an autoregressive decoder LSTM with a location-sensitive attention mechanism [28], which produces the Mel-spectrogram. Finally, the Mel-spectrogram is input to a post-net with five convolution layers, which predicts the residual and improves the synthesis by adding the residual.

In our case (voice conversion), the inputs of the PPG2speech synthesizer are PPGs instead of text embeddings. The PPG sequences are usually significantly longer than text embedding sequences. To capture the high-level phonetic and contextual information in an input PPG sequence, we replace the LSTM layer in the encoder with two pyramidal-LSTM (pLSTM) layers [29]. Each pLSTM reduces the time resolution by a factor of two, and therefore our encoder produces four times shorter hidden representation sequences compared with the input sequences. To generalize the Tacotron2 model to perform many-to-many VC, we condition the decoder with a speaker embedding. In this work, we use a d-vector [30] as the speaker embedding, and concatenate it with the encoder output, following [31].

The overall framework of our PPG2speech synthesizer is shown in Figure 2. Given a non-parallel corpus containing multiple speakers, the inputs to the network are pairs of PPGs ($\boldsymbol{x} \in \mathbb{R}^{T \times D}$) and the corresponding speaker embeddings ($\boldsymbol{s} \in \mathbb{R}^{M}$), where T is the length of the sequence, D is the dimensionality of the PPGs, and M is the dimensionality of speaker embedding. During training, a PPG sequence $\boldsymbol{x}$ is first fed to the encoder $\boldsymbol{E}$,

$$\boldsymbol{z} = \boldsymbol{E}(\boldsymbol{x}; \boldsymbol{\theta_e}) \tag{1}$$

where $\boldsymbol{z}$ is the resulting hidden representation and $\boldsymbol{\theta_e}$ are the encoder parameters. Then, the hidden representation $\boldsymbol{z}$ and the speaker embedding $\boldsymbol{s}$ are concatenated and fed to an autoregressive attention-decoder network (along with the post-net) $\boldsymbol{D}$ with parameters $\boldsymbol{\theta_d}$, to produce the Mel-spectrogram $\boldsymbol{o_{mel}}$,

$$\boldsymbol{o_{mel}} = \boldsymbol{D}([\boldsymbol{z}, \boldsymbol{s}]; \boldsymbol{\theta_d}) \tag{2}$$

At the same time, the network also predicts if the generating

process should stop, i.e., a stop token $\boldsymbol{o_{stop}}$. The model is optimized by minimizing the loss:

$$L_{VC}(\boldsymbol{\theta_e}, \boldsymbol{\theta_d}) = \alpha \left\| \boldsymbol{o_{mel}} - \boldsymbol{y_{mel}} \right\|_2^2 + \beta CE(\boldsymbol{o_{stop}}, \boldsymbol{y_{stop}}) \quad (3)$$

where $\boldsymbol{y_{mel}}$ is the ground-truth Mel-spectrogram; $\boldsymbol{y_{stop}}$ is the ground truth stop token values; $CE(\cdot)$ is the cross-entropy loss; α, β are the weights for each term to control the relative importance.

3.2. Proposed method: Adversarial speaker classifier

As we have noted, the PPG2speech synthesizer ignores the fact that PPGs carry speaker individualities such as accent, intonation, and speaking rate. As a result, the converted speech can still resemble the source speaker. The proposed adversarial speaker classifier, shown in Figure 2, is designed to address this issue. The classifier $\boldsymbol{C}$ takes the encoder output $\boldsymbol{z}$ as input and passes it through three fully-connected layers. The last layer produces a probability for each speaker:

$$\boldsymbol{p} = \boldsymbol{C}(\boldsymbol{z}; \boldsymbol{\theta_c}) = \boldsymbol{C}(\boldsymbol{E}(\boldsymbol{x}; \boldsymbol{\theta_e}); \boldsymbol{\theta_c}) \quad (4)$$

where $\boldsymbol{\theta_c}$ denote the parameters of the classifier. The encoder $\boldsymbol{E}$ and adversarial speaker classifier $\boldsymbol{C}$ are jointly trained with the speaker classification loss:

$$L_{CLS}(\boldsymbol{\theta_e}, \boldsymbol{\theta_c}) = -\sum_{k=1}^{K} \mathbb{I}(y_{speaker} == k) \log p_k \quad (5)$$

where $\mathbb{I}(\cdot)$ is the indicator function, K is the number of speakers, $y_{speaker}$ is the speaker who produced $\boldsymbol{x}$, and p_k is the probability of speaker k. During training, parameters $\boldsymbol{\theta_c}$ are optimized to minimize the classification loss to better identify the corresponding speaker, whereas parameters $\boldsymbol{\theta_e}$ are optimized to maximize the classification loss (i.e., to fool the classifier.) This minimax competition will finally converge when the output of the encoder is sufficiently speaker-independent such that the classifier is not able to identify the speaker.

The VC model is trained jointly with the adversarial speaker classifier in a multi-task learning fashion,

$$L(\boldsymbol{\theta_e}, \boldsymbol{\theta_d}, \boldsymbol{\theta_c}) = L_{VC}(\boldsymbol{\theta_e}, \boldsymbol{\theta_d}) - \lambda L_{CLS}(\boldsymbol{\theta_e}, \boldsymbol{\theta_c}) \quad (6)$$

where λ control the relative importance of L_{CLS}. Parameters $\boldsymbol{\theta_e}, \boldsymbol{\theta_d}, \boldsymbol{\theta_c}$ are optimized such that,

$$\boldsymbol{\theta_e}, \boldsymbol{\theta_d} = \text{argmin} L(\boldsymbol{\theta_e}, \boldsymbol{\theta_d}, \boldsymbol{\theta_c}) \quad (7)$$

$$\boldsymbol{\theta_c} = \text{argmax} L(\boldsymbol{\theta_e}, \boldsymbol{\theta_d}, \boldsymbol{\theta_c}) \quad (8)$$

and they can be updated though back-propagation using stochastic gradient descent (SGD) as,

$$\boldsymbol{\theta_e} \leftarrow \boldsymbol{\theta_e} - \mu(\frac{\partial L_{VC}}{\partial \boldsymbol{\theta_e}} - \lambda \frac{\partial L_{CLS}}{\partial \boldsymbol{\theta_e}}) \quad (9)$$

$$\boldsymbol{\theta_d} \leftarrow \boldsymbol{\theta_d} - \mu \frac{\partial L_{VC}}{\partial \boldsymbol{\theta_e}} \quad (10)$$

$$\boldsymbol{\theta_c} \leftarrow \boldsymbol{\theta_c} - \mu \frac{\partial L_{CLS}}{\partial \boldsymbol{\theta_c}} \quad (11)$$

where μ is the learning rate. The negative coefficient $-\lambda$ in eq. 9 reversed the gradient back-propagated from the adversarial speaker classifier. The gradient reversal maximizes L_{CLS} for $\boldsymbol{\theta_e}$ and makes the encoder fool the classifier, which is key to the optimization. In practice, we use the *gradient reversal layer* introduced in [32, 33]. During forward-propagation, it operates as an identity transform, and during back-propagation it multiplies the gradient by $-\lambda$.

4. Experimental setup

4.1. Acoustic model, speaker recognition model, and neural vocoder

We used a fully-connected DNN [34] as the acoustic model, which outputs 5,816 senones. We used the implementation in Kaldi [35] and trained the acoustic model on the Librispeech corpus [36]. We implemented the speaker recognition model proposed in [30] to produce a 256 dimensional d-vectors and trained it on the VoxCeleb2 dataset [37]. We used a universal WaveRNN [38] as the neural vocoder for all the testing speakers. The vocoder was trained on the VCTK training set (see below). Both the speaker recognition model and the neural vocoder were implemented in PyTorch [39].

4.2. PPG2speech synthesizer

We trained and evaluated the proposed VC system on the CSTR VCTK Corpus [18], which contains utterances from 109 English speakers with several accents (e.g., British, American, Scottish, Irish, Indian). For each speaker, there are on average 300 utterances, a subset of which have the same linguistic contents across all speakers. In our experiments, we divided the corpus into three subsets: a training set, a *standard* (test speakers were seen in training) test set, and a *one-shot* set (test speakers were unseen in training) test set. The training set consists of 105 speakers. Among these speakers, we selected four speakers for standard testing (p227, p228, p240, p256). We used the first 20 utterances of these speakers as the standard test set, and excluded them from the training set. The one-shot test set consists of the first 20 utterances of 4 speakers (p225, p226, p229, p232) that did not appear during training. All the test speakers had a British accent. For the standard test set, we considered four VC directions: p227 to p228 (M-F), p228 to p240 (F-F), p240 to p256 (F-M), and p256 to p227 (M-M). For the one-shot test set, we also 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).

For each utterance, we down-sampled the waveform from 48kHz to 16kHz to match the sampling rate of other modules, and then extracted an 80-dim Mel-spectrogram with a 50ms window and 12.5ms shift. Following the same frame shift, we extracted the PPG (collapsed into a 40-dim mono-phone PPG from the 5,816-dim senone PPG) and the d-vector (256-dim) for each utterance using the acoustic model and speaker recognition model, respectively. The fully-connected layers of adversarial speaker classifier have 512 nodes. We set other model hyperparameters following [17].

We implemented the VC models using TensorFlow[1] [40] and trained on a single NVIDIA V100 GPU. Hyperparameters α, β were set to 1.0, 0.005 empirically. Following [32], we gradually changed λ in adversarial speaker classifier from 0 to

[1] Audio samples and source code are available at https://github.com/shaojinding/Adversarial-Many-to-Many-VC.

1 during the training process as:

$$\lambda_p = \frac{2}{1 + \exp(-10 \cdot p)} - 1 \quad (12)$$

where p is the percentage of the training process. We used a batch size of 64 and an Adam Optimizer with a learning rate of 10^{-4}. The model converged after 60,000 steps, and the entire training time was around 30 hours.

5. Experiments

We conducted both objective and subjective experiments under *standard* and *one-shot* conditions. For objective evaluation, we used the Mel-Cepstral Distortion (MCD) [41] between VC and the ground-truth target utterances. Since computing MCD requires the ground-truth target speech, we selected a subset of 19 utterances that have the same linguistic content. For subjective evaluation, we conducted two listening tests on Amazon Mechanical Turk. In the first test, we asked listeners to rate the similarity between pairs of utterances using a Voice Similarity Score (VSS) [42]. In the second test, we asked listeners to rate the acoustic quality using a Mean Opinion Score (MOS). All participants were required to pass a pre-test that asked them to identify different regional accents in the United States. Additionally, in each listening test, we used 12 calibration utterances to detect if participants were cheating. We excluded ratings of the calibration utterances from the data analysis.

5.1. Standard testing

For standard testing, we compared the proposed adversarial-learning approach (denoted as **Proposed**) against the baseline PPG2speech system in Section 3.1 (**PPG2speech**). We did not compare it to other non-parallel many-to-many VC methods, as our PPG2speech baseline shares the same spirit as previous methods. To ensure a fair comparison, we kept the encoder and decoder architectures identical to the proposed approach.

Results from the objective and subjective evaluations are summarized in Table 1. The proposed method achieved a statistically significant lower MCD (8.37) than the baseline (8.47, $p = 0.01$). For the VSS test, 17 participants rated 108 utterance pairs: 32 pairs (16 VC-SRC pairs, 16 VC-TGT pairs) for each of the three systems, and 12 calibration utterances[2]. 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. The VSS was computed by collapsing the above two fields into a 14-point scale: -7 (definitely different speakers) to +7 (definitely the same speaker). As shown in Table 1, the proposed approach received a VSS rating of -6.20 on VC-SRC pairs, and 5.02 on VC-TGT pairs, which indicated that listeners were confident that VC syntheses and source speech were produced by different speakers, and that syntheses and target speech were produced by the same speaker, respectively. These scores were significantly better than those for the baseline: -5.62 VC-SRC, 4.25 VC-TGT; $p \ll 0.001$ in both cases.

For the MOS test, 19 participants rated 72 utterances from the three VC systems: 20 utterances per system, and 12 calibration utterances. For each utterance, participants were required to rate its acoustic quality from 1-bad to 5-excellent. As shown in Table 1, participants rate the proposed approach to have a 3.86 MOS, which is higher than the baseline (3.77, $p = 0.03$).

Table 1: *Objective (MCD, lower the better) and subjective (MOS and VSS, higher the better) results under standard condition. All the results are shown with 95% confidence interval.*

Method	MCD	VSS		MOS
		VC-SRC	VC-TGT	
PPG2speech	8.47±0.07	-5.62±0.09	4.25±0.12	3.77±0.06
Proposed	**8.37±0.07**	**-6.20±0.06**	**5.02±0.10**	**3.86±0.05**

5.2. One-shot testing

For one-shot testing, we also compared the proposed approach against the PPG2speech baseline. Results from the objective and subjective evaluation tests are shown in Table 2. In the MCD test, the proposed method (9.31) marginally outperforms the PPG2speech baseline (9.38, $p = 0.04$). In the VSS test, 18 participants rated 76 utterance pairs: 32 pairs (16 VC-SRC pairs and 16 VC-TGT pairs) for each of the two systems, and 12 calibration utterances. As shown in Table 2, participants were quite confident that the syntheses from the proposed method and the source speech were produced by different speakers (-6.12 VC-SRC), and that the syntheses and the target speech were produced by the same speaker (4.80 VC-TGT). This result also surpasses the PPG2speech baseline (-5.53 VC-SRC, 4.17 VC-TGT; $p \ll 0.001$ in all cases) with statistical significance.

In the MOS test, 19 participants rated 52 utterances from the two VC systems: 20 utterances per system, and 12 calibration utterances. As shown in Table 2, participants rated the proposed approach to have a 3.77 MOS, which is significantly higher than the ratings of the baseline (3.61, $p \ll 0.001$).

Table 2: *Objective (MCD, lower the better) and subjective (MOS and VSS, higher the better) results under one-shot condition. All the results are shown with 95% confidence interval.*

Method	MCD	VSS		MOS
		VC-SRC	VC-TGT	
PPG2speech	9.38±0.09	-5.53±0.11	4.17±0.21	3.61±0.06
Proposed	**9.31±0.08**	**-6.12±0.10**	**4.80±0.20**	**3.77±0.06**

6. Conclusions

We have proposed an adversarial learning approach to improve speaker identity in non-parallel many-to-many voice conversion. During training, the encoder output is consumed by an adversarial speaker classifier, which is optimized to identify the corresponding speaker. At the same time, the encoder is optimized to fool the adversarial speaker classifier, and therefore, it can produce more speaker-independent linguistic representations. We conducted both objective and subjective experiments under standard and one-shot conditions. Results indicate that the proposed method consistently improves the speaker identity and acoustic quality of VC syntheses over the baseline under both conditions.

7. Acknowledgements

This work was supported by NSF awards 1619212 and 1623750.

[2] A VC-SRC pair consists of a VC utterance and an utterance randomly selected from the source speaker (SRC), whereas a VC-TGT pair consists of a VC utterance and an utterance randomly selected from the target speaker (TGT).

8. References

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

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

[3] S. Ding, G. Zhao, C. Liberatore, and R. Gutierrez-Osuna, "Learning structured sparse representations for voice conversion," *IEEE TASLP*, vol. 28, pp. 343–354, 2019.

[4] 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 TASLP*, vol. 22, no. 12, pp. 1859–1872, 2014.

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

[6] S. Ding and R. Gutierrez-Osuna, "Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion," *Interspeech*, pp. 724–728, 2019.

[7] 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*. IEEE, 2016, pp. 1–6.

[8] S. Liu, J. Zhong, L. Sun, X. Wu, X. Liu, and H. Meng, "Voice conversion across arbitrary speakers based on a single target-speaker utterance." in *Interspeech*, 2018, pp. 496–500.

[9] H. Lu, Z. Wu, D. Dai, R. Li, S. Kang, J. Jia, and H. Meng, "One-shot voice conversion with global speaker embeddings," *Interspeech*, pp. 669–673, 2019.

[10] S. H. Mohammadi and T. Kim, "One-shot voice conversion with disentangled representations by leveraging phonetic posteriorgrams," *Interspeech*, pp. 704–708, 2019.

[11] 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*. IEEE, 2018, pp. 5274–5278.

[12] J. Zhang, Z. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE TASLP*, 2019.

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

[14] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, p. 14.

[15] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification."

[16] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," *Interspeech*, pp. 1268–1272, 2017.

[17] J. Shen, R. Pang, R. J. Weiss *et al.*, "Natural TTS synthesis by conditioning Wavenet on Mel spectrogram predictions," in *ICASSP*. IEEE, 2018, pp. 4779–4783.

[18] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-CSTR VCTK corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[19] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA workshop on speech synthesis*, 2004.

[20] D. Erro, A. Moreno, and A. Bonafonte, "INCA algorithm for training voice conversion systems from nonparallel corpora," *IEEE TASLP*, vol. 18, no. 5, pp. 944–953, 2009.

[21] T. Nakashika, T. Takiguchi, and Y. Minami, "Non-parallel training in voice conversion using an adaptive restricted boltzmann machine," *IEEE TASLP*, vol. 24, no. 11, pp. 2032–2045, 2016.

[22] 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.

[23] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent conversion using phonetic posteriorgrams," in *ICASSP*. IEEE, 2018, pp. 5314–5318.

[24] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-VC2: Improved cyclegan-based non-parallel voice conversion," in *ICASSP*. IEEE, 2019, pp. 6820–6824.

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

[26] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *ICASSP*. IEEE, 2019, pp. 6790–6794.

[27] W.-C. Huang, H. Luo, H.-T. Hwang *et al.*, "Unsupervised representation disentanglement using cross domain features and adversarial learning in variational autoencoder based voice conversion," *arXiv preprint arXiv:2001.07849*, 2020.

[28] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NeurIPS*, 2015, pp. 577–585.

[29] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*. IEEE, 2016, pp. 4960–4964.

[30] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *ICASSP*. IEEE, 2018, pp. 4879–4883.

[31] Y. Jia, Y. Zhang, R. Weiss *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *NeurIPS*, 2018, pp. 4480–4490.

[32] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, 2015, pp. 1180–1189.

[33] Z. Meng, J. Li, Y. Gong, and B.-H. Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *ICASSP*. IEEE, 2018, pp. 5949–5953.

[34] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*. IEEE, 2014, pp. 215–219.

[35] D. Povey, A. Ghoshal, G. Boulianne *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[36] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[37] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *Interspeech*, pp. 1086–1090, 2018.

[38] N. Kalchbrenner, E. Elsen, K. Simonyan *et al.*, "Efficient neural audio synthesis," in *ICML*, 2018, pp. 2410–2419.

[39] A. Paszke, S. Gross, S. Chintala *et al.*, "Automatic differentiation in pytorch," 2017.

[40] M. Abadi, A. Agarwal *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[41] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1. IEEE, 1993, pp. 125–128.

[42] D. Felps and R. Gutierrez-Osuna, "Developing objective measures of foreign-accent conversion," *IEEE TASLP*, vol. 18, no. 5, pp. 1030–1040, 2010.