CONDITIONAL RESTRICTED BOLTZMANN MACHINE FOR VOICE CONVERSION

Zhizheng Wu[1,2], Eng Siong Chng[1,2], Haizhou Li[1,2,3]

[1]School of Computer Engineering, Nanyang Technological University (NTU), Singapore
[2]Temasek Laboratories@NTU, Nanyang Technological University (NTU), Singapore
[3]Human Language Technology Department, Institute for Infocomm Research (I^{2}R), Singapore
{wuzz, aseschng}@ntu.edu.sg, hli@i2r.a-star.edu.sg

ABSTRACT

The conventional statistical-based transformation functions for voice conversion have been shown to suffer over-smoothing and over-fitting problems. The over-smoothing problem arises because of the statistical average during estimating the model parameters for the transformation function. In addition, the large number of parameters in the statistical model cannot be well estimated from the limited parallel training data, which will result in the over-fitting problem. In this work, we investigate a robust transformation function for voice conversion using conditional restricted Boltzmann machine. Conditional restricted Boltzmann machine, which performs linear and non-linear transformations simultaneously, is proposed to learn the relationship between source and target speech. CMU ARCTIC corpus is adopted in the experimental validations. The number of parallel training utterances is varied from 2 to 40. For these different training situations, two objective evaluation measures, mel-cepstral distortion and correlation coefficient, both show that the proposed method outperforms the main stream joint density Gaussian mixture model method consistently.

Index Terms— Speech synthesis, voice conversion, conditional restricted Boltzmann machine

1. INTRODUCTION

The purpose of *voice conversion* is to modify one speaker's voice (source) to sound like another speaker (target) without changing the linguistic information. The voice conversion techniques usually consist of training and real-time conversion processes. During the training process, a relationship is learned from the paired source and target feature vectors. The feature vectors can be any parameters which represent the speaker identity. In the run-time conversion process, the relationship is applied to the input source feature vectors and pass the converted feature vectors to synthesis filter to reconstruct speech signal. Obviously, to learn a robust conversion function from the limited amount of parallel training data is the key technique.

To implement a robust conversion function and generate natural sounding speech signal, a number of statistical methods have been proposed. For example, joint density Gaussian mixture model (JD-GMM) [1], partial least squares regression [2], mixture of factor analyzers [3] and local linear transformation [4] methods try to build a local linear transformation function. In addition, methods, such as neural network [5, 6] and dynamic kernel partial least squares regression [7] have also been proposed to learn the non-linear relationship between source and target speech. Because of the probabilistic treatment and flexibility, JD-GMM based methods have become one of the most effective and popular methods [1, 8, 9].

Although JD-GMM based methods can effective transform the source speech feature vectors into target speech feature space and generate converted speech with acceptable quality, the *over-smoothing* and *over-fitting* problems have been reported in [10, 2, 3, 4]. Over-smoothing is due to the statistical average during training the mean vectors and covariance matrices of the Gaussian components [10] (e. g., each mean vector is a weighted summation of all the training vectors). When the parallel training data is insufficient to estimated many parameters of the full covariance in each Gaussian components, the JD-GMM method will be over-fitting and cannot make good prediction for unseen testing data. Motivated by the success of restricted Boltzmann machine and conditional restricted Boltzmann machine in speech recognition [11], we proposed to utilize *conditional restricted Boltzmann machine* to estimate a robust conversion function for voice conversion. Different from conventional non-linear or linear transformation approaches, the conditional restricted Boltzmann machine (CRBM) learns both linear and non-linear relationship between source and target speech vectors. Thus, CRBM can learn the more detail relationship between the source and target speech.

2. CONVENTIONAL JOINT DENSITY GAUSSIAN MIXTURE MODEL METHOD

In this study, we adopt the mainstream joint density Gaussian mixture model (JD-GMM) conversion method, which is orig-

inal proposed in [12], as our baseline method for comparison.

In the off-line training process, given parallel training utterances from source X and target Y speakers, dynamic time warping (DTW) can be adopted to align the source and target speech vector sequences, and the aligned feature vector pairs are written as: $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, \ldots, \mathbf{z}_T]$, where $\mathbf{z}_t = [\mathbf{x}_t^\top, \mathbf{y}_t^\top]^\top \in \mathcal{R}^{2d}$ and $\mathbf{x}_t \in \mathcal{R}^d$, $\mathbf{y}_t \in \mathcal{R}^d$. The joint probability density of $\mathbf{X}$ and $\mathbf{Y}$ is modelled by Gaussian mixture model as below:

$$P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Z}) = \sum_{k=1}^{K} \pi_k^{(z)} \mathcal{N}(\mathbf{z}|\mu_k^{(z)}, \mathbf{\Sigma}_k^{(z)}), \quad (1)$$

where $\mu_k^{(z)} = \begin{bmatrix} \mu_k^{(x)} \\ \mu_k^{(y)} \end{bmatrix}$ and $\Sigma_k^{(z)} = \begin{bmatrix} \Sigma_k^{(xx)} & \Sigma_k^{(xy)} \\ \Sigma_k^{(yx)} & \Sigma_k^{(yy)} \end{bmatrix}$ are the mean vector and covariance matrix of the k^{th} Gaussian compoents, respectively. Given a component k, $\pi_k^{(z)}$ is its prior probability with $\sum_{k=1}^{K} \pi_k^{(z)} = 1$. In the training process, the GMM parameters $\lambda^{(z)} = \{\pi_k^{(z)}, \mu_k^{(z)}, \mathbf{\Sigma}_k^{(z)} | k = 1, 2, \ldots, K\}$ are estimated using the expectation maximization (EM) algorithm.

During the run-time conversion process, we employ the estimated JD-GMM to build a conversion function, and then apply the conversion function to a input source speech feature vector $\mathbf{x}$, where the conversion function $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{x})$ is given as follows:

$$\mathcal{F}(\mathbf{x}) = \sum_{k=1}^{K} p_k(\mathbf{x})(\mu_k^{(y)} + \Sigma_k^{(yx)}(\Sigma_k^{(xx)})^{-1}(\mathbf{x} - \mu_k^{(x)})), \quad (2)$$

$$p_k(\mathbf{x}) = \pi_k \mathcal{N}(\mathbf{x}|\mu_k^{(x)}, \mathbf{\Sigma}_k^{(xx)}) / \sum_{l=1}^{K} \pi_l \mathcal{N}(\mathbf{x}|\mu_l^{(x)}, \Sigma_l^{(xx)}), \quad (3)$$

where $p_k(\mathbf{x})$ is the posterior probability of the source vector $\mathbf{x}$ belonging to the k^{th} Gaussian component. After the converted feature vector sequence $\hat{\mathbf{Y}}$ is obtained, we pass the feature vector sequence to a synthesis filter to reconstruct a speech signal.

3. PROPOSED CONDITIONAL RESTRICTED BOLTZMANN MACHINE FOR VOICE CONVERSION

During training the JD-GMM model, each mean vector or covariance matrix is obtained by averaging all the training data (each training vector is assigned a occupation probability). Thus, the converted speech will suffer the over-smoothing problem. In addition, full covariance matrices are usually adopted to better estimate the relationship between source and target. While when the parallel training data is limited, the full covariance matrices will be inaccurate and result in the over-fitting problem.

To better represent the relationship between source and target speech, we proposed to employ conditional restricted Boltzmann machine [13] to learning the linear and non-linear relationship simultaneously. In this selection, we will introduce the conditional restricted Boltzmann machine, and the training and generation processes for voice conversion task.

The model structure of the conditional restricted Boltzmann machine (CRBM) is presented in Fig. 1. There is a hidden layer and a visible layer, where the hidden units can only have binary value (0 is to deactivate the unit and 1 is to activate the unit). We note that hidden layer usually has sparse activated unit, which will avoid over-fitting problem. In the visible unit, y_t is the target frame vector at time t, and x_t is the corresponding source vector.

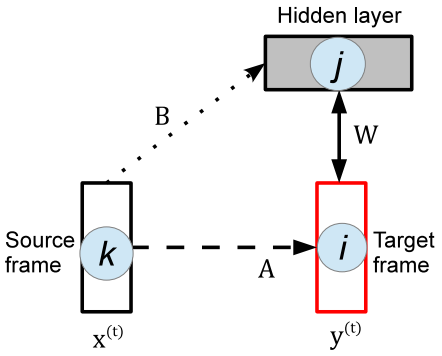

Fig. 1. Model structure of the conditional restricted Boltzmann machine. The target frame y_t and hidden units are symetricly connected, and both conditioned on x_t

The energy function of conditional restricted Boltzmann machine is written as follows [13]:

$$E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}; \mathbf{h}) = \sum_i \frac{(y_i^{(t)} - a_i - \sum_k A_{k,i} x_k^{(t)})^2}{2\sigma_i^2} - \sum_{ij} W_{ij} \frac{y_i^{(t)}}{\sigma_i} h_j - \sum_j (b_j + \sum_k B_{kj} x_k^{(t)}) h_j \quad (4)$$

where $y_i^{(t)}$ is the input for the i^{th} target unit at time t, $x_k^{(t)}$ is the input for the k^{th} source unit at time t, and h_j is the j^{th} hidden unit of the hidden layer represented by $\mathbf{h}$. σ_i is the variance for the i^{th} visible units. a_i and b_j are the biases of i^{th} visible and j^{th} hidden units, respectively. $\mathbf{A}$ and $\mathbf{B}$ are the autoregressive weights and source-to-hidden weights, respectively. $\mathbf{W}$ are the symmetric weights between visible and hidden units.

$$p(\mathbf{y}^{(t)}, \mathbf{h}; \theta | \mathbf{x}^{(t)}) = \frac{\exp(-E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{h}; \theta))}{Z(\theta)}, \quad (5)$$

where $Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(-E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{h}; \theta))$ is the partition function.

The conditional probability given the hidden layer to observe $y_i^{(t)}$ is given as follows:

$$P(y_i^{(t)}|\mathbf{h}, \mathbf{x}^{(t)}) = \mathcal{N}(a_i + \sum_k A_{ki} x_k^{(t)} + \sum_j W_{ij} h_j, \sigma_i^2) \quad (6)$$

Similarly, the conditional probability of hidden unit given visible layer is:

$$P(h_i|\mathbf{y}^{(t)}, \mathbf{x}^{(t)}) = \frac{1}{1 + \exp(-b_i - \sum_k B_{kj} x_k^{(t)} - \sum_i W_{ij} y_i^{(t)})} \quad (7)$$

3.1. Parameters estimation

To estimate the parameters $\theta = \{\mathbf{W}, \mathbf{A}, \mathbf{B}, \mathbf{b}, \mathbf{a}\}$, gradient descent method is adopted. The gradient of the parameters $\theta = \{\mathbf{W}, \mathbf{A}, \mathbf{B}, \mathbf{b}, \mathbf{a}\}$ can be obtained by minimizing the negative log-likelihood $\mathcal{L}(\theta) = -\log p(\mathbf{y}^{(t)}; \theta|\mathbf{x}^{(t)})$.

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}(-\log \sum_{\mathbf{h}} p(\mathbf{y}^{(t)}, \mathbf{h}; \theta|\mathbf{x}^{(t)})) \quad (8)$$

$$= \frac{\partial}{\partial \theta}(-\log \sum_{\mathbf{h}} \frac{\exp(-E(|\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{h}; \theta))}{Z(\theta)}) \quad (9)$$

Thus, we could obtain that:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{y}^{(t)}, \mathbf{x}^{(t)}) \frac{\partial E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{h}; \theta)}{\partial \theta} \quad (10)$$

$$- \sum_{\mathbf{h}, \mathbf{y}^{(t)}} p(\mathbf{h}, \mathbf{y}^{(t)}|\mathbf{x}^{(t)}) \frac{\partial E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{h}; \theta)}{\partial \theta} \quad (11)$$

$$= \mathbb{E}[\frac{\partial E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{h}; \theta)}{\partial \theta}|\mathbf{y}^{(t)}] - \mathbb{E}[\frac{\partial E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{h}; \theta)}{\partial \theta}] \quad (12)$$

Therefore, the gradients are written as follows.

$$\frac{\partial \mathcal{L}(\theta)}{\partial W_{ij}} = \mathbb{E}[y_i h_j|\mathbf{y}, \theta] - \mathbb{E}[y_i h_j|\theta], \quad (13)$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial A_{ki}} = \mathbb{E}[y_i x_k|\mathbf{y}, \theta] - \mathbb{E}[y_i x_k|\theta], \quad (14)$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial B_{kj}} = \mathbb{E}[h_j x_k|\mathbf{y}, \theta] - \mathbb{E}[h_j x_k|\theta], \quad (15)$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial a_i} = \mathbb{E}[y_i|\mathbf{y}, \theta] - \mathbb{E}[y_i|\theta], \quad (16)$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial b_j} = \mathbb{E}[h_j|\mathbf{y}, \theta] - \mathbb{E}[h_j|\theta], \quad (17)$$

where $\mathbb{E}[*|\mathbf{y}, \theta]$ is an expectation with respect to the training data distribution, and $\mathbb{E}[*|\theta]$ is an expectation with respect to the reconstructed data which is obtained by Gibbs sampling[14, 13].

3.2. Converted speech generation

After the model parameters are obtained, we can generate converted speech feature vector y_τ given source speech x_τ. As shown in (6), we assume the target feature vector follow Gaussian distribution, therefore, we use the mean as the converted feature vector given each source feature vector. The conversion process is done as follows:

Step 1: Initialize the target vector y_τ;

Step 2: Calculate hidden unit posteriori using Eq. (7);

Step 3: Reconstruct target vector y_τ;

Step 4: Return Step 2 and repeat Step 2 and 3 until the change of y_τ is smaller than a threshold;

Step 5: Output y_τ as the converted feature vector.

We note that there are two strategies to initialize the target vector: 1) randomly initialization; 2) use source vector or JD-GMM converted vector with some random noise for initialization. We also note that the second initialization method gives fast convergence.

4. EXPERIMENTS

To evaluate the performance of the proposed method, we conduct voice conversion experiments by using the CMU ARCTIC corpus. The speech data of two male (RMS and BDL) and two female (SLT and CLB) speakers are selected, and the voice conversion experiments are performed for four speaker pairs, including inter-gender and intra-gender conversions. We evaluate the performance by varying the number of training utterances. The training sets are consisting of 2, 10, 20, 30 and 40 utterances (these utterances are all selected from the a0001 to a0040 utterances in CMU ARCTIC corpus). 20 utterances (from b0451 to b0470), which are not included in the training sets, are used as testing sets. We note that the experimental results are reported by averaging all the four speaker pairs.

To analyze the speech signal which is sampled at 16 kHz, we first employ STRAIGHT [15] to decompose the speech signal into spectral envelope, fundamental frequency (F0) and aperiodic envelope, then the 24-dimensional mel-cepstral coefficients (MCC) are derived from spectral envelope, and band aperiodicity is converted from aperiodic envelope. During conversion, the 24-dimensional MCCs are converted using the methods as described above, and the energy coefficient (the zeroth cepstral coefficient) and log-scale F0 are converted by equalizing the mean and variance of the source and target speakers, while the aperiodic component is not converted.

4.1. Implementation details

The implementation details in conditional restricted Boltzmann machine (CRBM) are presented as follows. Before training, we normalize each dimension of the whole training data to zero mean and unit variance. We set the mini-batch size to 50 and the use 500 hidden units. The number of visible units is the same the the dimensionality of the mel-cepstral coefficients, which is 24. During training, the learning rate is set to 0.0001 except that the learning rate for autoregressive matrix A is set to 0.00005. We use same weight decay, 0.005, for all the parameters. The momentum parameter is set to 0.8 and only used after 5 epochs of training. All the parameters are randomly initialized. 500 epochs are iterated to estimate the parameters. During testing, after MCCs are generated from CRBM, we normalize the MCCs to have the same mean and variance as that of JD-GMM converted MCCs.

4.2. Objective evaluation

We employ two objective measures in this study. The first measure is the mel-cepstral distortion (MCD), which is calculated between the converted and target MCC vectors and defined as follows:

$$\mathrm{MCD[dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(c_d - c_d^{\mathrm{converted}})^2} \qquad (18)$$

where c_d and $c_d^{\mathrm{converted}}$ are the d-th original target frame and the converted MCCs, respectively. The lower of MCD value, the smaller distortion.

The second objective measure is the correlation coefficient, which is calculated between the original target and the converted MCC trajectories. The correlation coefficient has been used to measure the similarity between the synthsized/converted and the original fundamental frequency trajectories [16, 17]. The correlation coefficient for the d-th dimension MCC trajectory is defined as follows:

$$\gamma_d = \frac{\sum_{n=1}^{N}(c_{n,d} - \overline{c_d})(c_{n,d}^{converted} - \overline{c_d}^{converted})}{\sqrt{\sum_{n=1}^{N}(c_{n,d} - \overline{c_d})^2}\sqrt{\sum_{n=1}^{N}(c_{n,d}^{converted} - \overline{c_d}^{converted})^2}} \qquad (19)$$

where γ_d is the correlation coefficient calculated between the original target frame and converted d-th dimension MCC trajectories, and $c_{n,d}$ and $c_{n,d}^{converted}$ are the d-th dimension coefficients of the n-th original target and converted MCC vectors, respectively. $\overline{c_d}$ and $\overline{c_d}^{converted}$ are the mean values of the d-th dimension of the original target and converted MCC trajectories, respectively. The correlation coefficient is calculated for each dimension and we report the average correlation. We note that a higher correlation coefficient indicates the more similarity between converted and target MCC trajectories.

The spectral distortion results as a function of the number of parallel training utterances are presented in Fig. 2. As we increase the number of the training utterances, the spectral distortions of both JD-GMM and CRBM methods decrease. From 2 to 40 training utterances, CRBM outperforms JD-GMM method consistently in terms of spectral distortion, and CRBM give 0.3 dB lower distortion than JD-GMM method for all the training situations.

The average correlation coefficient results with respect to the number of parallel training utterances are shown in Fig. 3. From the results, we can find that as we increase the number of parallel training utterances, the correlation between converted and original target MCC trajectories increases consistently for both methods, and CRBM always obtains higher correlation coefficient than JD-GMM method. We note that for both spectral distortion and correlation coefficient results, the number of model parameters in CRBM is fixed. While the number of parameters of JD-GMM is varied, and the best result is reported for each training situation.

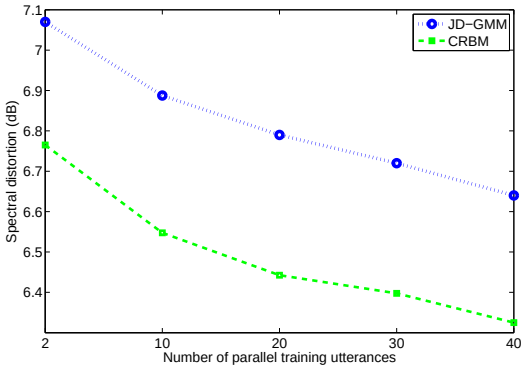

Fig. 2. Average spectral distortion as a function of the number of parallel training utterances

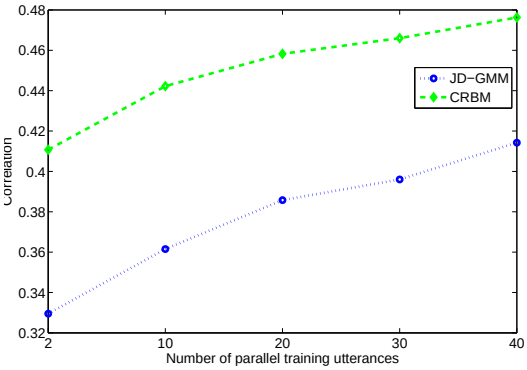

Fig. 3. Average correlation coefficient as a function of the number of parallel training utterances

5. CONCLUSION AND DISCUSSION

In this study, conditional restricted Boltzmann machine (CRBM) is proposed to estimate a conversion function for voice conversion. Two objective evaluation measures, mel-cepstral distortion and correlation coefficients, are adopted to measure the performance. The objective results both show that the proposed method outperforms the conventional mainstream joint-density Gaussian mixture model (JD-GMM) approach. By varying the number of training utterances from 2 to 40, we make different training situations. For the different training situations, the number of model parameters in CRBM is fixed and the model training configurations, such as learning rate, are also fixed. The different training situations show that the proposed model can estimate robust mapping function for the voice conversion task. In the future work, we will conduct formal subjective evaluation test to check the effectiveness of our proposed methods.

6. REFERENCES

[1] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[2] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 912–921, 2010.

[3] Z. Wu, T. Kinnunen, E. Chng, and H. Li, "Mixture of factor analyzers using priors from non-parallel speech for voice conversion," *Signal Processing Letters, IEEE*, vol. 19, no. 12, pp. 914–917, 2012.

[4] V. Popa, H. Silen, J. Nurminen, and M. Gabbouj, "Local linear transformation for voice conversion," in *ICASSP 2012*.

[5] M. Narendranath, H.A. Murthy, S. Rajendran, and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks," *Speech communication*, vol. 16, no. 2, pp. 207–216, 1995.

[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. 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.

[8] 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.

[9] D. Saito, S. Watanabe, A. Nakamura, and N. Minematsu, "Statistical voice conversion based on noisy channel model," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 6, pp. 1784–1794, 2012.

[10] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu, "Voice conversion with smoothed gmm and map adaptation," in *Eurospeech-2003*, 2003, pp. 2413–2416.

[11] A.R. Mohamed and G. Hinton, "Phone recognition using restricted boltzmann machines," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4354–4357.

[12] Alexander Kain and Michael W Macon, "Spectral voice conversion for text-to-speech synthesis," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. IEEE, 1998, vol. 1, pp. 285–288.

[13] G.W. Taylor, G.E. Hinton, and S.T. Roweis, "Modeling human motion using binary latent variables," *Advances in neural information processing systems*, vol. 19, pp. 1345, 2007.

[14] G.E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[15] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[16] Z.Z. Wu, T. Kinnunen, E.S. Chng, and H. Li, "Text-independent f0 transformation with non-parallel data for voice conversion," *Proc. Interspeech 2010*, pp. 1732–1735, 2010.

[17] Y. Qian, Z. Wu, B. Gao, and F.K. Soong, "Improved prosody generation by maximizing joint probability of state and longer units," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1702–1710, 2011.