Rapid Style Adaptation Using Residual Error Embedding for Expressive Speech Synthesis

Xixin Wu[1], *Yuewen Cao*[1], *Mu Wang*[2], *Songxiang Liu*[1], *Shiyin Kang*[3], *Zhiyong Wu*[*1,2], *Xunying Liu*[1], *Dan Su*[3], *Dong Yu*[3], *Helen Meng*[1,2]

[1]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, China
[2]Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[3]Tencent AI Lab, Tencent, Shenzhen, China

{wuxx, ywcao, sxliu, zywu, xyliu, hmmeng}@se.cuhk.edu.hk, wangmu16@mails.tsinghua.edu.cn,
{dansu, dyu, shiyinkang}@tencent.com

Abstract

Synthesizing expressive speech with appropriate prosodic variations, e.g., various styles, still has much room for improvement. Previous methods have explored to use manual annotations as conditioning attributes to provide variation information. However, the related training data are expensive to obtain and the annotated style codes can be ambiguous and unreliable. In this paper, we explore utilizing the residual error as conditioning attributes. The residual error is the difference between the prediction of a trained average model and the ground truth. We encode the residual error into a style embedding via a neural network-based error encoder. The style embedding is then fed to the target synthesis model to provide information for modeling various style distributions more accurately. The average model and the error encoder are jointly optimized with the target synthesis model. Our proposed method has two advantages: 1) the embedding is automatically learned with no need of manual style annotations, which helps overcome data sparsity and ambiguity limitations; 2) For any unseen audio utterance, the style embedding can be efficiently generated. This enables rapid adaptation to the desired style to be achieved with only a single adaptation utterance. Experimental results show that our proposed method outperforms the baseline model in both speech quality and style similarity.

Index Terms: speech synthesis, residual error, prosodic variation, expressiveness

1. Introduction

In recent years speech synthesis technologies have made rapid progress with successful application of deep learning techniques [1–5], generating synthesized speech with excellent intelligibility and naturalness. However, expressiveness, i.e., prosodic variations appropriate for the speech context, still has much room for improvement. One key challenge is how to incorporate prosodic control information.

In previous work, style controls have been explored with various conditioning attributes providing variational information, e.g., discriminant codes [6], unsupervised clustered labels [7], style tokens [8], etc. [9] proposed to learn style control information by training separated decision trees for different styles in hidden Markov model (HMM)-based synthesis system. [6] and [10] proposed to feed speaker codes to input or hidden layers of deep neural network (DNN) for performing multi-speaker synthesis. [11] incorporated latent variables to model emotional variation in training data with predefined partitions. [12] investigated different representations of emotional labels. In deep learning, with manual annotations, embeddings learned from the attributes can be jointly optimized with networks weights using backpropagation. However, the required training data used in these approaches are expensive to obtain. Besides, since there is no commonly accepted hard categorization of styles, manual style annotations can be ambiguous and unreliable [8].

To avoid annotation errors, researchers have proposed to learn the embeddings in an unsupervised way, i.e., without manually annotated conditioning attributes. [7] clusters the training data into several style classes. The learned cluster labels are used as automatic annotations for generating specific styles in an HMM-based synthesis. The annotation component and the synthesis model are constructed separately [13]. In order to consistently optimize the style representation and back-end synthesis model, [14] proposed to learn a sentence-level control vector by feeding the sentence index as conditioning input. The projection from the sentence index to the control vector is jointly optimized with the synthesis system. [8] proposed to automatically learn embeddings, called latent style factors, directly during the training of speech synthesis system in an unsupervised data-driven way. The style factors are combined to provide information for modeling the prosodic variations. However, synthesis of speech with unseen styles requires the inference of style embedding to be learned first [15], typically by backpropagation to obtain the style embeddings [10], hidden representations [16] or adapting an already trained model to a specific style model [17].

Inspired by the successful application of deep residual learning in tasks such as image classification [18], our previous research found that the residual error between the predicted outputs by using the average style model and the corresponding target outputs of specific styles can be used as auxiliary features to improve style during adaptation [19]. In this paper, we further explore the use of prosodic style residual errors as an efficient form of conditioning attributes to improve the performance of expressive speech synthesis [20]. The motivation of utilizing the residual error is that the average style model can capture phonetic information and average prosodic style, while

This work was done when Xixin Wu was an intern at Tencent AI Lab

*Corresponding author

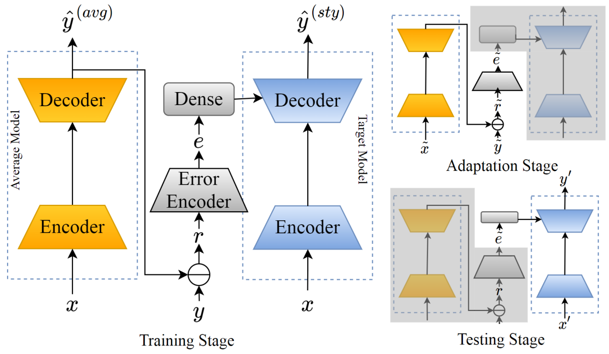

Figure 1: *The architecture of error encoding networks (EEN). At the training stage, the residual error* $\boldsymbol{r}$ *is calculated based on the prediction of the average model* $\hat{\boldsymbol{y}}^{(avg)}$ *and the corresponding ground truth* $\boldsymbol{y}$*. The style embedding* $\boldsymbol{e}$ *is generated using the error encoder with* $\boldsymbol{r}$ *as input. The target model generates the output* $\hat{\boldsymbol{y}}^{(sty)}$ *based on input text* $\boldsymbol{x}$ *and the dense layer outputs. At the adaptation stage, the embedding* $\tilde{\boldsymbol{e}}$ *is calculated with the adaptation sample (*$\tilde{\boldsymbol{x}}$*,* $\tilde{\boldsymbol{y}}$*). At the testing stage, the outputs* $\boldsymbol{y}'$ *are generated using the target model with the embedding* $\tilde{\boldsymbol{e}}$ *and the testing inputs* $\boldsymbol{x}'$*.*

the style variation information can be obtained from the residual errors. It is also efficient to use the residual error as conditioning attributes to save the effort in separating the average style and style variations. Recently, Tacotron, a successful application of encoder-decoder architecture, has achieved state-of-the-art performance in speech synthesis in neutral prosody [3, 21]. We extend the use of Tacotron to model prosodic styles for expressive speech synthesis using a diverse and expressive speech corpus of children's audiobooks [22].

We propose to incorporate an error encoder to encode residual errors into style embeddings, which are used as control factors for modeling varying prosodic styles in the Tacotron based synthesis model [21]. Our method has two advantages:

- The embedding is automatically learned with no need of manual style annotations, which helps overcome data sparsity and ambiguity limitations.
- For any unseen audio utterance, the style embedding can be efficiently generated. This enables rapid adaptation to the desired style to be achieved with only a single adaptation utterance.

This paper is organized as follows: Section 2 reviews the encoder-decoder structure and the error encoding network based on Tacotron's structure. Section 3 presents the details of the experiments. The conclusion is drawn in Section 4.

2. Model Architecture

2.1. Encoder-Decoder Structure

The encoder-decoder structure has been applied successfully to various sequence modeling problems [21, 23]. Both the encoder and decoder are composed using neural networks. In speech synthesis, assume the input is linguistic vector $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ and the output is acoustic vector $\boldsymbol{y} = \{y_1, y_2, ..., y_t\}$. The encoder first encodes the linguistic vectors into hidden representations. The information of input is stored in the hidden representations. In the decoding stage, a weighted sum of the hidden representations, the attended context, is generated for each step. The acoustic outputs are decoded conditioning on the context.

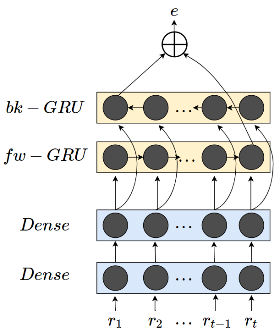

Figure 2: *The structure of error encoder.*

2.2. Error Encoding Networks

In order to provide the needed information source for modeling style variation in the outputs, we propose to incorporate the residual errors as conditioning attributes into the Tacotron framework [21]. Encoding errors for modeling multiple distributions in outputs has been utilized to predict video frames conditioned on previous frames [24]. In this work, we apply the idea to modeling multiple style distributions in the acoustic outputs.

As shown in Fig. 1, error encoding networks (EEN) consists of four parts: 1) the average model, 2) the error encoder, 3) dense projection layers and 4) the target model. The average model has the same structure as Tacotron, directly trained with multi-style data. The loss function of the average model is:

$$\mathcal{L}^{(avg)} = \| \boldsymbol{y} - \hat{\boldsymbol{y}}^{(avg)} \| \tag{1}$$

where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}^{(avg)}$ are the ground truth of acoustic vectors and the prediction of the average model, respectively. By optimizing the loss function $\mathcal{L}^{(avg)}$, the average model will be generally driven to predict average acoustic outputs $\hat{\boldsymbol{y}}^{(avg)}$ given input linguistic features $\boldsymbol{x}$.

The error encoder is designed to encode the frame-level residual error sequence into a style embedding vector. The residual error is defined as:

$$\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}}^{(avg)} \tag{2}$$

where $\boldsymbol{r}$ is a sequence of t frame-level residual error [25]. The structure of error encoder is shown in Fig. 2. The residual error sequence is first fed to dense layers. To capture the temporal information, bi-direction recurrent neural networks (RNN) are utilized to encode the dense layer outputs. The state values at the final timestep of the forward direction and the first timestep of the backward direction are concatenated as the style embedding vector $\boldsymbol{e}$.

The style embedding vector $\boldsymbol{e}$ is then fed to dense projection layers before connected to the decoder RNN of the target model. Similarly, the loss function of the target model is:

$$\mathcal{L}^{(sty)} = \| \boldsymbol{y} - \hat{\boldsymbol{y}}^{(sty)} \| \tag{3}$$

where $\hat{\boldsymbol{y}}^{(sty)}$ is the prediction of the target model generating speech with desired prosodic styles. Note that the acoustic features generated by the trained average model capture the phonetic information with an average style learned from training data. The part that still needs to be learned is the difference between the average style and the target style. Thus we utilize the residual errors from the trained average model to provide information for the target model to model the style distributions more accurately.

At the training stage, the two losses can be optimized together or sequentially by first optimizing $\mathcal{L}^{(avg)}$ and then $\mathcal{L}^{(sty)}$. The error encoder is optimized jointly with the target model. At the adaptation stage, given an adaptation utterance with the desired style, including the acoustic outputs $\tilde{\boldsymbol{y}}$ and the corresponding linguistic inputs $\tilde{\boldsymbol{x}}$, the residual error $\tilde{\boldsymbol{r}}$ is first calculated using the trained average model with the adaptation utterance. The residual error $\tilde{\boldsymbol{r}}$ is then fed to the error encoder to obtain the style embedding $\tilde{\boldsymbol{e}}$. At the synthesis stage, the style embedding $\tilde{\boldsymbol{e}}$ obtained in the adaptation stage is first projected to the input space of the decoder in the target model. The testing linguistic inputs $\boldsymbol{x}'$ are then fed to the target model to generate the target acoustic outputs $\boldsymbol{y}'$, with the projected style embedding added to the inputs of the target model decoder.

2.3. Tacotron-based Error Encoding Networks

One successful model of encoder-decoder structures applied in speech synthesis is Tacotron [21]. Tacotron can generate high-quality speech with neutral style. However, when it is applied to synthesizing speech with high style variations, the error accumulation problem at the output becomes serious, as the errors contained in the average-style output are larger [26]. [8] tried to solve this problem by introducing latent style factors to control Tacotron to synthesize various styles. Following [8], we add the style control factor, i.e., the residual error style embedding, to the input of RNN in the Tacotron decoder to implement EEN.

2.4. Zero Style Embedding

Assume that the model training ultimately converges, the outputs of the target model are sufficiently close to the ground truth model, i.e., $\hat{\boldsymbol{y}}^{(sty)} \approx \boldsymbol{y}$. When the residual error is zero, according to Eq. 2, the prediction of the average model equals to the ground truth, i.e., $\hat{\boldsymbol{y}}^{(avg)} - \boldsymbol{y} = 0$. Thus we have $\hat{\boldsymbol{y}}^{(sty)} \approx \hat{\boldsymbol{y}}^{(avg)}$, and can infer that when the style embedding is close to zero, the output style is close to the average style in the training data. We also evaluate this special embedding value in our experiments.

3. Experiments

3.1. Corpus

We evaluate our proposed method on the audiobook corpus from Blizzard Challenge 2016, which is recorded by a native female speaker [27]. The speaker tries to utter in different styles in the recording, including emotions, mimicked role characters' voice. There are 50 books in the audiobook data. We use the books of "A Midsummer Night's Dream" and "Romeo and Juliet" as testing data, and the other 48 books as training data (around 4.3 hours). Alphabet character sequences are used as linguistic inputs. We extract the logarithmic magnitude linear-scale spectrograms and 80-band mel-scale spectrograms with 50-ms Hanning window, 12.5-ms shift, and 2048-point Fourier transform. The output spectrogram magnitudes are converted to waveforms with the Griffin-Lim algorithm.

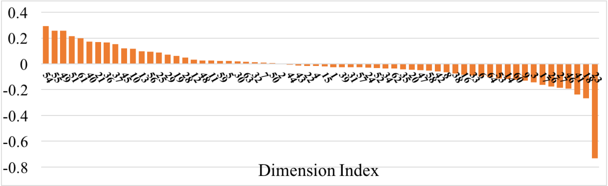

Figure 3: *Pearson correlation coefficients between each dimension of the style embeddings and the mean F0 values of training data.*

3.2. Systems

Our baseline is the Tacotron system [21]. We aim to investigate the EEN from two aspects: 1) whether EEN can have better prosodic control for expressive speech than the baseline Tacotron; 2) whether EEN can adapt to the target style with only a single adaptation utterance. As far as we know, existing methods, e.g., backpropagation of input codes [10] and adaptation of top layers [17], cannot perform well with only a single utterance sample for adaptation.

We implement the Tacotron system strictly following [21]. For the EEN system, the structure of the average model is the same as Tacotron. The error encoder consists of three layers, i.e., two dense layers and one bi-directional gated recurrent unit (GRU) layer from bottom to top, as shown in Fig. 2. The two dense layers have 128 units per layer, activated by ReLU, dropped out with rate of 0.5. The bi-directional GRU layer has 32 memory blocks in each direction. The state values of the final timestep in forward direction and the first timestep in the backward direction are concatenated as the 64-dimension style embedding. The residual error is calculated with the *teach-forced prediction* of the average model [8]. We use one dense projection layer between the error encoder and the decoder of target model. The dense layer has 256 units without the bias vector, activated by a linear function. The target model has the same structure as Tacotron, except that the inputs of RNN in the decoder are added with the output of the dense projection layer. Though it is suggested that training the average model and the target model sequentially works better in [24], our preliminary experiments show that training the two models simultaneously together with error encoder works better. We compare the synthesized audio samples of three systems:

- Tacotron—This system is trained directly with the training data.
- EEN with style embedding $\boldsymbol{e} = 0$ (EEN-0)—This system is based on the EEN trained with training data. The acoustic outputs are generated with the target model in the EEN, fed with linguistic input from testing sample and conditioned on style embedding with zero values.
- EEN with style embedding obtained from the adaptation sample (EEN-adpt)—This system is the same as the EEN-0, except that the style embedding is calculated based on the adaptation sample, with the average model and the error encoder.

3.3. Style Embedding Analysis

To investigate how the learned style embedding vectors control the synthesized prosodic variations, we calculate Pearson correlation coefficients between each dimension of the embeddings

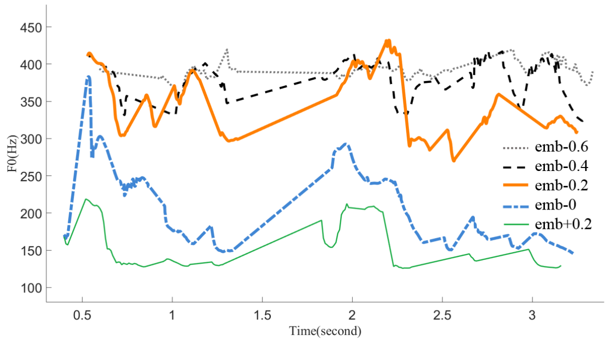

Figure 4: *Smoothed F0 trajectories of the manipulated embeddings.*

and the mean fundamental frequency (F0) of all training samples. As shown in Fig. 3, the 23-rd dimension of the embedding has strong negative correlation with the mean F0 values. To verify the control ability of the embedding, we compare the F0 trajectories of the synthetic speech when the embedding varies only in the 23-rd dimension. We first extract an embedding vector, denoted as emb-0, from a random reference sample in the training set. Then we add $+0.2$, -0.2, -0.4 and -0.6 to the 23-rd dimension of emb-0 to obtain four new embeddings, denoted as emb+0.2, emb-0.2, emb-0.4 and emb-0.6. We then use these embeddings to generate audio samples. The smoothed F0 trajectories (linear interpolation in unvoiced frames) of synthetic speeches are shown in Fig. 4. We can observe that the F0 trajectories increase as the value of the 23-rd dimension decreases. This demonstrates that the learned embedding is able to control the synthesis of prosody in different scales[1].

3.4. Subjective Evaluation

We perform subjective evaluation using the mean opinion score (MOS) and ABX tests. 20 utterances are each synthesized based on unseen text inputs by the Tacotron and the EEN-0 system. For the EEN-adpt system, 20 utterances with texts are first randomly selected from the testing data as the adaptation samples. For each adaptation sample, the style embedding is calculated. Based on the style embedding, 20 utterances are synthesized with the same unseen text inputs as the Tacotron and EEN-0 system. In the MOS test, each subject listens to each utterance synthesized by the three systems and give a 5-point scale score of naturalness (5: excellent, 4: good, 3: fair, 2: poor, 1: bad). In the preference tests, each subject listens to 20 pair of utterances synthesized by two different systems, and the corresponding reference utterance. The listeners are required to provide a style similarity choice among 3 options: 1) the former in the pair is more similar to the reference utterance; 2) the latter is more similar; 3) no preference or neutral (i.e., the difference between the paired utterances cannot be perceived or can be perceived but it is difficult to choose which one is more similar). We conduct our subjective evaluation on the crowdsourcing platform of Amazon Mechanical Turk. 27 feedbacks are solicited for the MOS test and 34 feedbacks for each of the preference tests.

As shown in Fig. 5, both the EEN-0 system (p <1e-9) and the EEN-adpt system (p <1e-5) achieve better performance than the Tacotron system, which indicates that the introduction of style embedding can provide prosodic information for the EEN to better model the high prosodic variations in the training data. Interestingly, the EEN-0 outperforms the other two systems. Possible reasons may be: (1) In the training data, the number of utterances with less style variations or neutral prosody are larger than that of utterances with high variations, i.e., the ratio of style embedding reaching zero is high. Hence the EEN-0 gets trained better than the EEN-adpt. (2) There exist multiple style distributions in the training data. Without conditioning information, Tacotron cannot model the distributions separately and tends to compromise to an average of the multiple distributions. While for the EEN system, the conditioning residual error provide separate information to the weights to fit each distributions accurately.

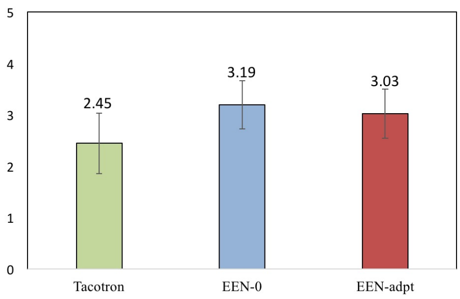

Figure 5: *MOS test results of various systems.*

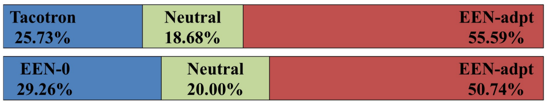

Figure 6: *Preference test results of various systems.*

The results of the preference tests are given in Fig. 6. The EEN-adpt system significantly outperforms both Tacotron ($p < 0.001$) and EEN-0 ($p < 0.01$) system, which demonstrates that EEN can efficiently capture the style in the adaptation sample[1].

4. Conclusions

In this paper, we propose to apply error encoding network to model the prosodic styles with the prosodic variation information provided by style embeddings, which are encoded from the residual error between the prediction of trained average model and the ground truth. The embeddings are fed to the Tacotron based synthesis model as the conditioning control factors. Our proposed methods can simultaneously provide the following advantages: 1) obtaining style variation information from the residual error; 2) learning the style embedding based on the residual error in an unsupervised way and 3) rapid adaptation with a single adaptation utterance. Experimental results show that introducing style embeddings helps to improve synthetic speech quality and similarity, and the learned embeddings can efficiently control the output prosodic style. In the future, we will investigate the combination of prosodic styles learned from training data into desired unseen styles.

5. Acknowledgement

This work is partially supported by the General Research Fund from the Research Grants Council of Hong Kong SAR Government (Project No. 14205814)

[1] Some samples are available in “http://www1.se.cuhk.edu.hk/~wuxx/IS18/eenstyle.html”

6. References

[1] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] 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," in *Proc. NIPS*, 2017.

[3] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018.

[4] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[5] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. ICASSP*. IEEE, 2013, pp. 8012–8016.

[6] N. Hojo, Y. Ijima, and H. Mizuno, "An investigation of dnn-based speech synthesis using speaker codes." in *Prof. Interspeech*, 2016, pp. 2278–2282.

[7] F. Eyben, S. Buchholz, and N. Braunschweiler, "Unsupervised clustering of emotion and voice styles for expressive tts," in *Proc. ICASSP*. IEEE, 2012, pp. 4009–4012.

[8] Y. Wang, R. Skerry-Ryan, Y. Xiao, D. Stanton, J. Shor, E. Battenberg, R. Clark, and R. A. Saurous, "Uncovering latent style factors for expressive speech synthesis," *arXiv preprint arXiv:1711.00520*, 2017.

[9] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, "A style control technique for hmm-based expressive speech synthesis," in *IEICE Transcations on Information and Systems*, vol. 90, 2007, pp. 1406–1413.

[10] H.-T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, "Adapting and controlling dnn-based speech synthesis using input codes," in *Proc. ICASSP*. IEEE, 2017, pp. 4905–4909.

[11] G. E. Henter, J. Lorenzo-Trueba, X. Wang, and J. Yamagishi, "Principles for learning controllable tts from annotated and latent variation," in *Proc. Interspeech*, 2017, pp. 3956–3960.

[12] J. Lorenzo-Trueba, G. E. Henter, S. Takaki, J. Yamagishi, Y. Morino, and Y. Ochiai, "Investigating different representations for modeling and controlling multiple emotions in dnn-based speech synthesis," *Speech Communication*, 2018.

[13] M. Tachibana, S. Izawa, T. Nose, and T. Kobayashi, "Speaker and style adaptation using average voice model for style control in hmm-based speech synthesis," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4633–4636.

[14] O. Watts, Z. Wu, and S. King, "Sentence-level control vectors for deep neural network speech synthesis," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] Z. Malisz, H. Berthelsen, J. Beskow, and J. Gustafson, "Controlling prominence realisation in parametric dnn-based speech synthesis," in *Proc. Interspeech*, 2017, pp. 1079–1083.

[16] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Unsupervised speaker adaptation for dnn-based tts synthesis," in *Proc. ICASSP*. IEEE, 2016, pp. 5135–5139.

[17] Y. Fan, Y. Qian, F. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for dnn-based tts synthesis," in *Proc. ICASSP*. IEEE, 2015, pp. 4475–4479.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] X. Wu, L. Sun, S. Kang, S. Liu, Z. Wu, X. Liu, and H. Meng, "Feature based adaptation for speaking style synthesis," in *Proc. ICASSP*, 2018.

[20] F. Meng, H. Meng, Z. Wu, and L. Cai, "Synthesizing expressive speech to convey focus using a perturbation model for computer-aided pronunciation training," in *Second Language Studies: Acquisition, Learning, Education and Technology*, 2010.

[21] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech syn," in *Proc. Interspeech*, 2017.

[22] K. Sawada, C. Asai, K. Hashimoto, K. Oura, and K. Tokuda, "The nitech text-to-speech system for the blizzard challenge 2016," in *Blizzard Challenge 2016 Workshop*, 2016.

[23] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[24] M. Henaff, J. Zhao, and Y. LeCun, "Prediction under uncertainty with error-encoding networks," *arXiv preprint arXiv:1711.04994*, 2017.

[25] J. Wu, D. Huang, L. Xie, and H. Li, "Denoising recurrent neural network for deep bidirectional lstm based voice conversion," *Proc. Interspeech 2017*, pp. 3379–3383, 2017.

[26] X. Wu, S. Kang, L. Sun, Y. Ning, Z. Wu, and H. Meng, "Attention-based recurrent generator with gaussian tolerance for statistical parametric speech synthesis," in *Workshop on Affective Social Multimedia Computing*, 2017.

[27] S. King and V. Karaiskos, "The blizzard challenge 2016," in *Proc. Blizzard Challenge Workshop*, 2016.