Attention-based Recurrent Generator with Gaussian Tolerance for Statistical Parametric Speech Synthesis

Xixin Wu[1], *Shiyin Kang*[3*], *Lifa Sun*[1], *Yishuang Ning*[2], *Zhiyong Wu*[1,2], *Helen Meng*[1,2]

[1] Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China
[2]Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Shenzhen Key Laboratory of Information Science and Technology,
Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[3]Tencent AI Lab, Tencent, Shenzhen, China

{wuxx,lfsun,zywu,hmmeng}@se.cuhk.edu.hk, ningys13@mails.tsinghua.edu.cn,
shiyinkang@tencent.com

Abstract

Conventional statistical parametric speech synthesis (SPSS) generates frame-level acoustic features in two separately optimized steps—namely, duration prediction and acoustic feature generation. It also incorporates a conditional independence assumption to generate independent output frames given textual inputs. Both factors constrain the quality of the generated speech output. This work proposes to apply the attention-based recurrent generator (ARG) with Gaussian Tolerance (GT) for SPSS, where duration prediction and acoustic feature generation are jointly optimized with attention mechanism, and the dependency across output frames is modeled by acoustic feature generation conditioned on preceding frames. GT is introduced to train ARG to acquire robustness based on previous output frames with errors. Perceptual experiments comparing the naturalness between ARG and the conventional hidden Markov model show a gain in MOS score and the effectiveness of GT.

Index Terms: Statistical parametric speech synthesis, Attention mechanism, Sequence to sequence, Joint optimization

1. Introduction

Statistical parametric speech synthesis (SPSS) has made significant progresses these years [1][2]. However, the naturalness of speech generated by SPSS still lies below that of well-built unit selection systems [3]. Two reasons leading to the unnaturalness of SPSS are: the two-step optimization and the conditional independence assumption [4]. More specifically, conventional SPSS generates frame-level acoustic features in two steps: duration prediction and acoustic feature generation. These two steps are optimized separately with their own objectives instead of being optimized globally, thus potentially limiting the naturalness of generated speech [5]. [6] tried to jointly model duration and acoustic feature with mixture density network. Besides, previous research shows that acoustic features of different frames are correlated with each other [4]. However, under the assumption of conditional independence, different frames of acoustic features are assumed to be independent of each other given the input linguistic features. Dynamic features are used to address this problem [7]. [8] tried to consider the temporal relation by introducing recurrent connections in the output layer to a long short-term memory recurrent neural network (LSTM-RNN) based SPSS system, but improvements are still needed.

Recently, attention mechanism has been successfully applied to automatic speech recognition (ASR) to jointly train components of ASR, with one objective to obtain a globally optimal solution, without the conditional independence assumption [9]. Motivated by this idea, we try to utilize attention-based recurrent generator (ARG) for SPSS. In ARG, the attentive context, a weighted sum of encoded representation of input features, is first calculated. Based on this context, the output acoustic features are generated. The calculation of these attention weights, the encoded representation and acoustic feature generation, are modeled together and optimized jointly aiming at generating high quality speech. On the other hand, ARG generates acoustic features of each frame based on the attentive context, which are calculated based on acoustic features of previous frames and input linguistic features. This connects the output frames and brings the relation across frames into consideration. However, the output errors in previous frames will affect the generation in current frame. The errors will be accumulated along the sequence, which degrades output performance significantly. [10] tried to address this problem with input quantification, which may introduce additional error. Also, different acoustic features (i.e., spectral, pitch features) need to be quantified with different scales.

In this paper, we try to completely model all the four acoustic features in a unified framework, including Mel-cepstral coefficients (MCEPs), band aperiodicities (BAPs), logarithmic fundamental frequency (LF0), and voiced/unvoiced (V/UV) decisions. Thus we need to consider the error of four features simultaneously. Under the assumption that the output error follows a Gaussian distribution, the output error is treated as Gaussian noise. We introduce Gaussian tolerance (GT) in training to improve the robustness of ARG, by leveraging the denoising ability of neural network. Our experiments show that introducing the GT can help solve the error accumulation problem. ARG can generate speech with better naturalness than HMM. Compared with previous reported results of ARG-based SPSS [10], our system models acoustic features in a more holistic way for generating speech waveform. Compared with [11] and [12], this paper focuses on the investigation of the error accumulation problem.

The rest of this paper is organized as following, we briefly introduce the two-step structure of SPSS, as well as the conditional independence assumption, in Section 2. The structure of ARG will be described in Section 3. The error accumulation

*Formerly affiliated with CUHK, now Tencent AI Lab problem and the GT is illustrated in Section 4. Experimental results and conclusions will be given in Sections 5 and 6.

2. Conventional Statistical Parametric Speech Synthesis

Conventional SPSS is divided into two parts: duration prediction and frame-level acoustic feature generation. In the duration prediction stage, the duration model learns to predict the numbers of frames of basic modeling units (i.e., states, phonemes, syllables). In acoustic feature generation, the acoustic feature outputs are generated based on corresponding HMM states, or frame-level linguistic features. To formalize this problem, we denote the input linguistic feature sequence as $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, and the output frame-level acoustic feature sequence as $\boldsymbol{y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_T\}$, where N and T are the numbers of timesteps in input sequence and output sequence, respectively. Our task to convert text to speech can be formalized as the probability density function (PDF) of acoustic features $\boldsymbol{y}$ given the linguistic features $\boldsymbol{x}$:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}) &= \sum_{\forall \boldsymbol{d}} p(\boldsymbol{d}|\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{d}, \boldsymbol{x}) \\ &\approx p(\boldsymbol{d}^*|\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{d}^*, \boldsymbol{x}) \qquad (1) \\ \boldsymbol{d}^* &= \arg\max_{\boldsymbol{d}} p(\boldsymbol{d}|\boldsymbol{x}) \qquad (2) \end{aligned}$$

where $\boldsymbol{d}$ is duration of basic modeling units. $p(\boldsymbol{y}|\boldsymbol{x})$ is approximated by only considering the most probable duration $\boldsymbol{d}^*$. Then acoustic features are generated according to:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{d}^*, \boldsymbol{x}) \qquad (3)$$

where $\boldsymbol{y}^*$ is the generated acoustic features. Eq. (2) and (3) represent the duration model and the frame-level acoustic feature generation model respectively. These two models are optimized with different optimization objectives. The solution obtained in this way is not globally optimal to $p(\boldsymbol{y}|\boldsymbol{x})$, which potentially limits the quality of generated speech.

Conventionally, with the conditional independence assumption, each frame is treated independently given input linguistic feature sequence, as shown in Eq. (4).

$$p(\boldsymbol{y}|\boldsymbol{d}, \boldsymbol{x}) \approx \prod_t p(\boldsymbol{y}_t|\boldsymbol{d}_t, \boldsymbol{x}) \qquad (4)$$

However, acoustic features in different frames have temporal relations, and ignoring dependence across frames may constrain the quality of the generated speech. Recently, WaveNet shows impressive speech synthesis performance [13], where at each timestep, it generates waveform point conditioning on all previous timestep outputs.

$$p(\boldsymbol{y}|\boldsymbol{d}, \boldsymbol{x}) \approx \prod_t p(\boldsymbol{y}_t|\boldsymbol{d}_t, \boldsymbol{x}, \boldsymbol{y}_{<t}) \qquad (5)$$

To obtain a globally optimal solution and model the dependence across frames, we try to merge the two models (i.e., duration model and acoustic feature generation model) into a unified model with the attention mechanism, and generate frame-level acoustic features explicitly conditioned on the features of previous frames.

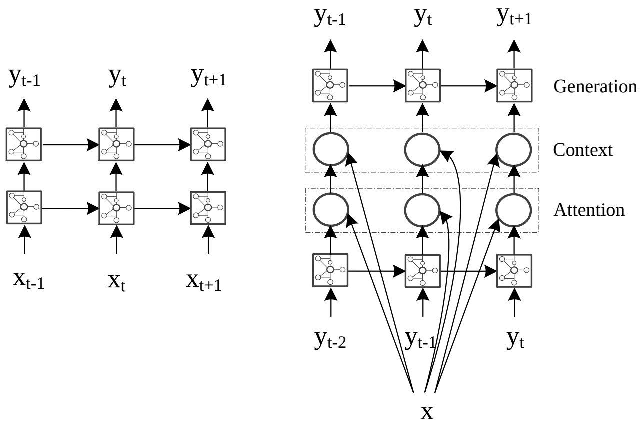

Figure 1: *Structure of LSTM-RNN (left) and ARG (right).*

3. Attention-based Recurrent Generator Model Structure

3.1. RNN & LSTM-RNN

RNN is a kind of neural network with recurrent connections between hidden units at neighboring timesteps to model the temporal relation. At each timestep, the output of hidden unit is conditioned not only on the current input, but also on the output of hidden units at the previous timestep. By replacing units in hidden layers of RNN with long short-term memory cells, LSTM-RNN can remember long-range context information [14].

3.2. Attention-based Recurrent Generator

The ARG is based on sequence to sequence (seq2seq) model, which is proposed to model the mapping between input and output with various lengths [9]. In seq2seq, the input is first encoded into an intermediate vector representation, and then the output is generated step by step, by decoding the intermediate representation, where the input information is stored. The general idea of the attention mechanism is to select a subset of input units to focus on, treated as context for generating outputs. When applied to SPSS, ARG generates the acoustic feature sequence frame by frame according to the weighted selection of linguistic features and previous output acoustic features as Eq. (6). Different from Eq. (5), duration is implicitly modeled here by the attention mechanism.

$$p(\boldsymbol{y}|\boldsymbol{x}) \approx \prod_t p(\boldsymbol{y}_t|\boldsymbol{x}, \boldsymbol{y}_{<t}) \qquad (6)$$

At timestep t, the attention layer $Attention$ computes the attention weight $\boldsymbol{\alpha}_t$ paid to each timestep of linguistic feature sequence $\boldsymbol{x}$ based on $\boldsymbol{x}$ and $\boldsymbol{r}_{t-1}$. $\boldsymbol{r}_{t-1}$ is the embedded hidden representation of $\boldsymbol{y}_{t-1}$. In this work, uni-directional LSTM-RNN is applied to build the embedding layer, thus $\boldsymbol{r}_{t-1}$ contains information of $\boldsymbol{y}_{<t}$. In the attentive context layer as Eq. (8), a weighted sum of linguistic features across timesteps is calculated to form an attentive context $\boldsymbol{g}_t$ for the current timestep, according to the attention weight $\boldsymbol{\alpha}_t$. The context is then fed into a generation layer $Generation$ to generate acous-

tic features of the current step as Eq. (9).

$$\boldsymbol{\alpha}_t = Attention(\boldsymbol{r}_{t-1}, \boldsymbol{x}) \quad (7)$$

$$\boldsymbol{g}_t = \sum_{n=1}^{N} \alpha_{t,n} \boldsymbol{x}_n \quad (8)$$

$$\boldsymbol{y} = Generation(\boldsymbol{g}) \quad (9)$$

where $\alpha_{t,n}$ is the attention weight for $\boldsymbol{x}_n$ at timestep t. Inside the attention layer $Attention$, $\alpha_{t,n}$ is determined as follows:

$$e_{t,n} = \boldsymbol{w}^{\mathrm{T}} \tanh(W^x \boldsymbol{x}_n + W^r \boldsymbol{r}_{t-1} + \boldsymbol{b}^e) \quad (10)$$

$$\alpha_{t,n} = \frac{\exp(e_{t,n})}{\sum_{n=1}^{N} \exp(e_{t,n})} \quad (11)$$

where W^x, W^r are the learned weight matrices and $\boldsymbol{w}, \boldsymbol{b}^e$ are the learned vectors.

Before the linguistic features $\boldsymbol{x}$ are fed into the attention layer, they need to be encoded into hidden representations via bi-directional encoding layers, because the input linguistic features are sparse, containing lots of binary features. On the other hand, the past and the future linguistic features are important for determining the attention of the current timestep.

4. Attention-based Recurrent Generator with Gaussian Tolerance for Speech Synthesis

We apply the ARG to model all the four acoustic features simultaneously. To address the problem of output error accumulation, we introduce GT to train ARG to acquire robustness against the output error, under the assumption that the output error follows a Gaussian distribution.

4.1. Training & Synthesis Stage

At the training stage, the ARG learns to generate acoustic features $\boldsymbol{y}$ based on linguistic features $\boldsymbol{x}$. At each timestep t, ARG is trained to make a one-step forward prediction as stated in Eq. (6). The output $\hat{\boldsymbol{y}}_t$ is generated based on the input linguistic features $\boldsymbol{x}$ and true target output $\boldsymbol{y}_{<t}$ to approximate the target output $\boldsymbol{y}_t$. The squared errors between $\hat{\boldsymbol{y}}_t$ and $\boldsymbol{y}_t$ of all timesteps are summed up as the loss function. At the synthesis stage, for any linguistic feature vector $\boldsymbol{x}$, the acoustic features $\hat{\boldsymbol{y}}$ are generated step by step using the trained ARG model. At each timestep t, the linguistic features $\boldsymbol{x}$ and the previous acoustic features $\hat{\boldsymbol{y}}_{<t}$ are used as input. For the first timestep t=1, the initial output $\hat{\boldsymbol{y}}_0$ is assigned a small Gaussian noise value. Based on $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}_{<t}$, the trained ARG model is driven to generate the acoustic features $\hat{\boldsymbol{y}}_t$ for timestep t and the process continues until the end of the utterance. A heuristic method to determine the end of output sequence is when the end of the input sequence dominates 80% of attention weight for a constant number of timesteps η, and the generation stops. η is empirically chosen as 5.

4.2. Mismatch & Gaussian Tolerance

As stated in [15], there is a mismatch between the training stage and synthesis stage. At the training stage, the output is conditioned on the true target acoustic outputs of previous timesteps. However, at the synthesis stage, the predicted acoustic outputs are provided to the model when generating the next timestep output. This mismatch will affect the output performance, since the predicted output may contain errors, while the trained ARG model treats the predicted erroneous output as true target output without any error. The error in the predicted output will be propagated as the generation goes on, which is referred as error accumulation [10]. Although some efforts have been made to alleviate this problem, e.g., input quantification [10], schedule sampling [16], etc., they still suffer from amplified errors.

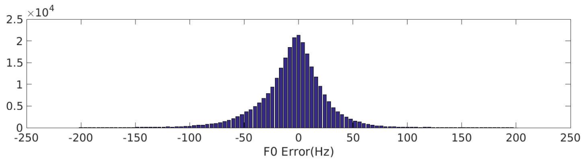

Figure 2: *Histogram of F0 output error of ARG without GT.*

We analysed the F0 output error of ARG (without GT) on a validation set, and observed that the distribution of the error follows a Gaussian distribution, as the histogram shown in Figure 2. Since the error in output is inevitable, especially in real-value output, we directly train the model on data with simulated output error. In this way, the trained model is tolerant with output error. Assume that there exists an error between the predicted output $\hat{\boldsymbol{y}}_t$ and the true output $\boldsymbol{y}_t$, following a Gaussian distribution with mean 0 and variance σ^2.

$$\epsilon = \hat{\boldsymbol{y}}_t - \boldsymbol{y}_t \sim \mathcal{N}(0, \sigma^2) \quad (12)$$

During training, we impose a Gaussian noise $\epsilon' \sim \mathcal{N}(0, (\sigma')^2)$ on the true output. The model is trained based on the artificially contaminated true output $\boldsymbol{y}_t + \epsilon'$. The motivation is to leverage the denoising ability of RNN to reduce noise from speech [17]. At the synthesis stage, as long as the distribution of the error between the predicted output and the true output is similar to our imposed Gaussian noise, i.e., $\sigma \approx \sigma'$, then the error can be reduced as noise, i.e., the error is tolerant by the model. Here σ is referred as Gaussian tolerance. Experimental results suggest that GT can be applied to improve output accuracy.

5. Experiment

5.1. Corpus

In our experiments, we use a corpus of female Mandarin native speaker, consisting of 5,428 utterances (around 5 hours), including 98,623 syllable samples. These syllable samples can be partitioned into 1,660 tonal syllable classes. We use 5,000 utterances as training data (5% is used as validation data), and the remaining 428 as testing data. The speech signals are sampled at 16 kHz, windowed by a 25-ms window and shifted every 5 ms. We extract 39-order MCEPs plus log energy, 25 BAPs, LF0, and V/UV decision as frame-level acoustic features. 80% of the starting and ending silence frames are removed to reduce the impact of many silence frames [8].

5.2. Experimental Setup

We trained two models, HMM and LSTM-RNN, as baseline systems. The HMM-based system is built with eleven-state HMM syllable models with left-to-right topology. Each state (i.e., the eleven states of one syllable) is modeled by a single Gaussian and related to certain output frames. For each frame, the four acoustic features are generated and then fed into STRAIGHT [18] to synthesize speech waveform.

For LSTM-RNN and ARG system, we use syllable features as well as prosodic features, e.g., prosodic word and prosodic phrase features. Theoretically, the prosodic features needed for generating acoustic features can be learned by the model. However, we only have about 5 hours of training data, which is insufficient to cover all possible prosodic contexts. Thus we add prosodic features to input linguistic features. The input linguistic features for each syllable include binary feautres for categorical linguistic contexts and numerical features for numerical linguistic features following [19]. The output frame-level acoustic features are the same as those modeled in HMM. All numerical linguistic features and acoustic features are normalized to have zero mean and unit variance.

LSTM-RNN system consists of two separately trained parts, i.e., the duration model and the acoustic feature generation model [8]. The duration model consists of 1 unidirectional hidden LSTM layer with 256 memory blocks and 1 linear output layer with linear activation function. The feature generation model is composed of 3 unidirectional LSTM layers with 256 memory blocks and another linear output layer.

In the ARG system, we use 2 bi-directional LSTM-RNN layers with 512 memory blocks per layer as the encoding layers. 2 and 4 uni-directional LSTM-RNN layers, with 256 memory blocks per layer, are respectively used as the embedding layers and generation layers.

5.3. Part-to-Whole Training for ARG

Initialization of attention weights as rough alignment is helpful for ARG training [10]. Since the updates of attention calculation and feature generation affect each other, separately training these two parts at first can help reach a better initialization. In this work, we adopt a part-to-whole training method to train ARG, consisting of two stages, part-training and fine-tuning. In the part-training stage, we train the attention layer with rough alignment results (from manual labelling knowledge or forced alignment) with context layer and generation layer fixed. Then we train the context layer and generation layer with attention layer fixed. In the fine-tuning stage, we optimize all layers of the whole model. More specifically, in part-training, if a frame is related to a certain syllable according to alignment labels, then the attention weight for the syllable is trained to be 95%, and the rest 5% averagely divided to the other syllables. The RMSProp algorithm and stochastic gradient descent (SGD) based back-propagation algorithm are respectively adopted in part-training and fine-tuning stages.

5.4. Objective Evaluation

To evaluate our system, we use Mel-cepstral distortion (MCD), root mean squared error (RMSE) of F0, and V/UV error rate to measure the performance. MCD is often used in speech synthesis to measure the distance between synthesized speech and the target speech [19].

GT is an important factor that will affect the final performance. Large GT is good for correcting error existing in model output. However, it will also increase the generation error of model. We conducted experiments to explore effect of various GT values. Table 1 shows that introducing of GT improves the output performance. Values around 0.1 are appropriate options of GT. Subjective evaluation results in Table 3 also demonstrate the effectiveness of utilizing GT.

The objective measures, e.g., MCD, are calculated frame by frame. In order to compare the generated speech with the target speech, we need to generate speech with the same duration as target speech. For HMM and LSTM-RNN, we use manually labeled duration to synthesize speech. For ARG, to generate speech with the same duration as target speech, and thus comparable with target speech, acoustic feature outputs are conditioned on true previous timestep acoustic features. The experimental results are shown in Table 2. In this table, our proposed ARG model achieves the best qualitative results. One of the reasons might be that the ARG model can generate acoustic features of each timestep conditioned on true acoustic features of previous steps, thus can use more contextual information than other systems.

Table 1: *Objective metrics on various values of GT in training.*

Gaussian Tolerance (σ)	MCD (dB)	F0 RMSE (Hz)	V/UV Error Rate (%)
0	5.92	26.38	10.79
0.01	5.77	26.50	10.7
0.1	**5.72**	**25.36**	**10.0**
0.5	5.72	25.82	10.51
1	5.84	25.73	10.5

Table 2: *Objective metrics on results of HMM, LSTM-RNN and ARG system.*

System	MCD (dB)	F0 RMSE (Hz)	V/UV Error Rate (%)
HMM	5.97	29.45	12.9
LSTM-RNN	5.85	29.32	11.1
ARG	**5.72**	**25.36**	**10.0**

5.5. Subjective Evaluation

We conduct mean opinion score (MOS) test to subjectively evaluate the naturalness of synthesized speech. 11 utterances are randomly selected from the test set. These utterances are synthesized by HMM, LSTM-RNN, ARG without GT and ARG with GT respectively, and thus we have 44 utterances to be evaluated[1]. We invite 20 subjects without hearing impairment to participate in the MOS test. Each of them listens to 44 utterances individually and rates the naturalness and clearness of each utterance based on a 5-point scale (5: excellent, 4: good, 3: fair, 2: poor, 1: bad). MOS scores of different systems with confidence interval (CI) at confidence level 0.95 are given in Table 3. It shows that our proposed method achieves better performance than HMM, but is still worse than LSTM-RNN. One possible reason is that salient errors in output features are beyond the capture of GT and the errors are amplified stepwise.

Table 3: *Subjective evaluation of HMM, LSTM-RNN, ARG without GT and ARG with GT.*

System	MOS	95% CI
HMM	2.99	±0.14
LSTM-RNN	3.50	±0.12
ARG w/o GT	2.12	±0.13
ARG w/ GT	3.10	±0.13

[1] Some synthesized speech samples are presented in https://sites.google.com/site/argttsdemo/

6. Conclusions

This paper proposes to apply attention-based recurrent generator (ARG) with Gaussian tolerance (GT) to SPSS, to solve the two existing problems in conventional SPSS, i.e., separate, local optimization and conditional independence assumption. We completely model four streams of acoustic features in ARG. To address the problem of error accumulation, we introduce GT to train ARG to acquire robustness against conditioned output errors. Both objective and subjective evaluation demonstrate the effectiveness of GT. For the moment, ARG achieves better performance than HMM, while has not exceeded that of LSTM-RNN. A possible reason is that salient errors in output feature generation are amplified along the output sequence. Hence, our next step is to improve the performance of the ARG by introducing context-dependent acoustic feature distribution to acoustic feature generation.

7. Acknowledgements

This project is partially supported by a grant from the NSFC/RGC Joint Research Scheme (Project No. N_CUHK404/15)

8. References

[1] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *ICASSP*. IEEE, 2013, pp. 7962–7966.

[2] Y. Fan, Y. Qian, F. Xie, and F. Soong, "TTS synthesis with bidirectional lstm based recurrent neural networks." in *Interspeech*, 2014, pp. 1964–1968.

[3] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *ICASSP*. IEEE, 2015, pp. 4460–4464.

[4] B. Uria, I. Murray, S. Renals, C. Valentini-Botinhao, and J. Bridle, "Modelling acoustic feature dependencies with artificial neural networks: Trajectory-rnade," in *ICASSP*. IEEE, 2015, pp. 4465–4469.

[5] Z. Ling, S. Kang, H. Zen, A. Senior, M. Schuster, X. Qian, H. 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.

[6] K. Tokuda, K. Hashimoto, K. Oura, and Y. Nankaku, "Temporal modeling in neural network based statistical parametric speech synthesis," in *Proc. ISCA Speech Synthesis Workshop*, 2016, pp. 113–118.

[7] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *ICASSP*, vol. 3. IEEE, 2000, pp. 1315–1318.

[8] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *ICASSP*. IEEE, 2015, pp. 4470–4474.

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

[10] W. Wang, S. Xu, and B. Xu, "First step towards end-to-end parametric tts synthesis: Generating spectral parameters with neural attention," in *Interspeech*, 2016, pp. 2243–2247. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-134

[11] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, C. A., and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *ICLR 2017 workshop*, 2017.

[12] 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," *arXiv preprint arXiv:1703.10135*, 2017.

[13] A. 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.

[14] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[15] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[16] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *NIPS*, 2015, pp. 1171–1179.

[17] A. Maas, Q. Le, T. ONeil, O. Vinyals, P. Nguyen, and A. Ng, "Recurrent neural networks for noise reduction in robust asr," in *Interspeech*, 2012.

[18] H. Kawahara, "Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical science and technology*, vol. 27, no. 6, pp. 349–353, 2006.

[19] S. Kang and H. Meng, "Statistical parametric speech synthesis using weighted multi-distribution deep belief network." in *Interspeech*, 2014, pp. 1959–1963.