TACOTRON-BASED ACOUSTIC MODEL USING PHONEME ALIGNMENT FOR PRACTICAL NEURAL TEXT-TO-SPEECH SYSTEMS

Takuma Okamoto[1]*, Tomoki Toda*[2,1]*, Yoshinori Shiga*[1]*, and Hisashi Kawai*[1]

[1]National Institute of Information and Communications Technology, Japan
[2]Information Technology Center, Nagoya University, Japan

ABSTRACT

Although sequence-to-sequence (seq2seq) models with attention mechanism in neural text-to-speech (TTS) systems, such as Tacotron 2, can jointly optimize duration and acoustic models, and realize high-fidelity synthesis compared with conventional duration-acoustic pipeline models, these involve a risk that speech samples cannot be sometimes successfully synthesized due to the attention prediction errors. Therefore, these seq2seq models cannot be directly introduced in practical TTS systems. On the other hand, the conventional pipeline models are broadly used in practical TTS systems since there are few crucial prediction errors in the duration model. For realizing high-quality practical TTS systems without attention prediction errors, this paper investigates Tacotron-based acoustic models with phoneme alignment instead of attention. The phoneme durations are first obtained from HMM-based forced alignment and the duration model is a simple bidirectional LSTM-based network. Then, a seq2seq model with forced alignment instead of attention is investigated and an alternative model with Tacotron decoder and phoneme duration is proposed. The results of experiments with full-context label input using WaveGlow vocoder indicate that the proposed model can realize a high-fidelity TTS system for Japanese with a real-time factor of 0.13 using a GPU without attention prediction errors compared with the seq2seq models.

Index Terms— Speech synthesis, neural text-to-speech, duration model, forced alignment, sequence-to-sequence model

1. INTRODUCTION

The real-time text-to-speech (TTS) technique is an important speech communication technology. In recent advances in deep learning, the duration and acoustic models (AMs) in statistical parametric speech synthesis (SPSS) have been replaced from hidden Markov models (HMMs) to deep neural networks (DNNs) [1]. Although conventional DNN-based SPSS systems with source-filter vocoders (e.g., STRAIGHT [2]) can achieve real-time synthesis [1, 3–5], the synthesized speech quality is still not sufficiently high because of the over-smoothing problem in AMs and the introduction of source-filter vocoders.

To simultaneously solve these problems, a neural network-based autoregressive (AR) generative model for raw audio, WaveNet, has been proposed [6]. Unlike conventional SPSS systems, WaveNet directly synthesizes speech waveforms from linguistic features with predicted phoneme durations and fundamental frequencies, and it outperforms conventional TTS systems based on unit selection and SPSS [6]. By introducing high-quality raw waveform modeling in WaveNet, neural vocoders that directly synthesize raw speech waveforms from acoustic features have been proposed [7, 8]; these also outperform conventional source-filter vocoders in SPSS [9].

Additionally, such neural vocoders can realize end-to-end TTS directly converting text to raw speech waveforms using sequence-to-sequence (seq2seq) neural networks. Although conventional SPSS systems separately train duration and AMs, seq2seq models jointly train them at once without a pipeline structure, and several end-to-end methods have been initially investigated [10–12]. Finally, Tacotron 2 can first realize end-to-end TTS for English with the same quality as natural speech by introducing a seq2seq model and neural vocoder to solve pipeline structure and source-filter vocoder problems in conventional SPSS [13]. In Tacotron 2, input characters are directly converted into mel-spectrograms using the seq2seq model, and speech waveforms are synthesized from the predicted mel-spectrograms using the AR WaveNet vocoder [13]. Following the success of Tacotron 2, many seq2seq approaches have been investigated for several languages, not only using character input [10–22] but also phoneme input [14–16, 19, 23–35].

However, many seq2seq approaches, including Tacotron 2, cannot achieve real-time synthesis because of the introduction of the AR WaveNet vocoder. To realize real-time high-fidelity neural seq2seq-based TTS systems, a real-time neural TTS system for Japanese using a seq2seq model based on Tacotron 2 with full-context label input that includes phoneme sequences and a WaveGlow vocoder [36] has been provided [37].

Although these seq2seq models with an attention mechanism can jointly optimize duration and AMs without phoneme alignment and achieve high-fidelity synthesis, they run the risk that speech samples sometimes cannot be successfully synthesized because of attention prediction errors, as shown in Fig. 3(b), even though attention prediction accuracy is improved [17, 20, 23, 24, 27, 34]. This is a crucial problem for practical TTS systems. By contrast, conventional duration-acoustic pipeline models are broadly used in practical TTS systems because the phoneme durations can be relatively easily predicted using simple HMM-based or DNN-based models, and there are few crucial prediction errors in the duration models. Although phoneme duration was additionally introduced into seq2seq models to control speech duration and improve attention prediction accuracy in [25, 34], these models are still based on an attention mechanism. Although FastSpeech can predict phoneme durations from a transformer-based model and stably synthesize speech waveforms, this method requires teacher-student training [38].

To realize high-fidelity practical TTS systems without attention prediction errors and teacher-student training, in this paper, Tacotron-based AMs for real-time neural vocoders with phoneme alignment instead of an attention mechanism are investigated by extending the seq2seq model with full-context label input [37] based on the following three facts.

- The model structure of Tacotron 2 has the potential to achieve higher-quality synthesis than conventional AMs.

- Many seq2seq methods introduce phoneme input [14–16, 19, 23–35] rather than character input [10–22].
- HMM-based phoneme alignment can be easily achieved when phoneme sequences are provided, and phoneme durations can be relatively easily predicted by conventional simple models.

In the investigation, the phoneme durations are first obtained from HMM-based forced alignment [39] and the duration model is trained using a conventional simple bidirectional long short-term memory (LSTM)-based model [40–42]. Then, a seq2seq model with forced alignment based on phoneme duration instead of an attention mechanism for phoneme-level sequences is investigated and an alternative AM with the Tacotron decoder and phoneme duration for frame-level sequences is proposed. These models are compared with the seq2seq model with an attention mechanism and a conventional bidirectional LSTM-based AM using a WaveGlow vocoder. This investigation is important because only a few seq2seq models have been compared with conventional pipeline models using neural vocoders [16, 27, 32].

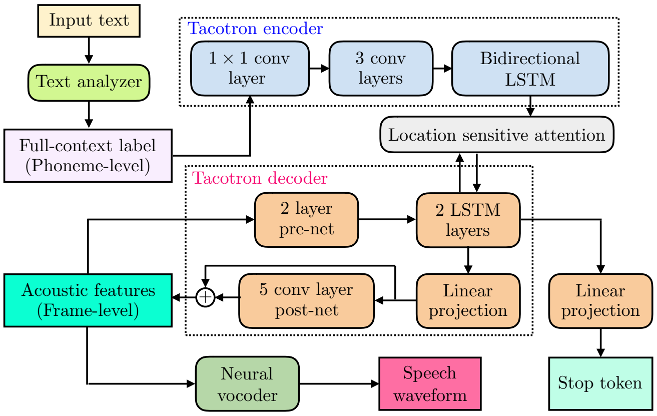


Fig. 1. Neural text-to-speech system based on a sequence-to-sequence acoustic model with full-context label input and a neural vocoder [37].

2. WAVEGLOW REAL-TIME NEURAL VOCODER

To overcome the synthesis speed problem in AR WaveNet and SampleRNN neural vocoders [6, 7], two types of solutions have been provided. The first is AR models with simple structures, such as FFTNet [43, 44], WaveRNN [45], and LPCNet [46]. In particular, WaveRNN and LPCNet can achieve real-time synthesis using a mobile CPU by introducing a sparse gated recurrent unit. The other solution is flow [47]-based non-AR models that simultaneously generate all speech samples at once, parallel WaveNet [32, 48–50], WaveGlow [36], and FloWaveNet [51]. Additionally, an alternative real-time approach, the neural source-filter (NSF) [52], has also been provided. Real-time neural TTS systems with parallel WaveNet [48] and WaveRNN [45] have been realized using linguistic features with predicted phoneme durations and fundamental frequencies in AR WaveNet TTS [6]. Compared with parallel WaveNet [32, 48–50] and NSF [52], WaveGlow models can be directly trained without teacher-student training and fundamental frequency analysis. Therefore, WaveGlow is introduced as a real-time neural vocoder.

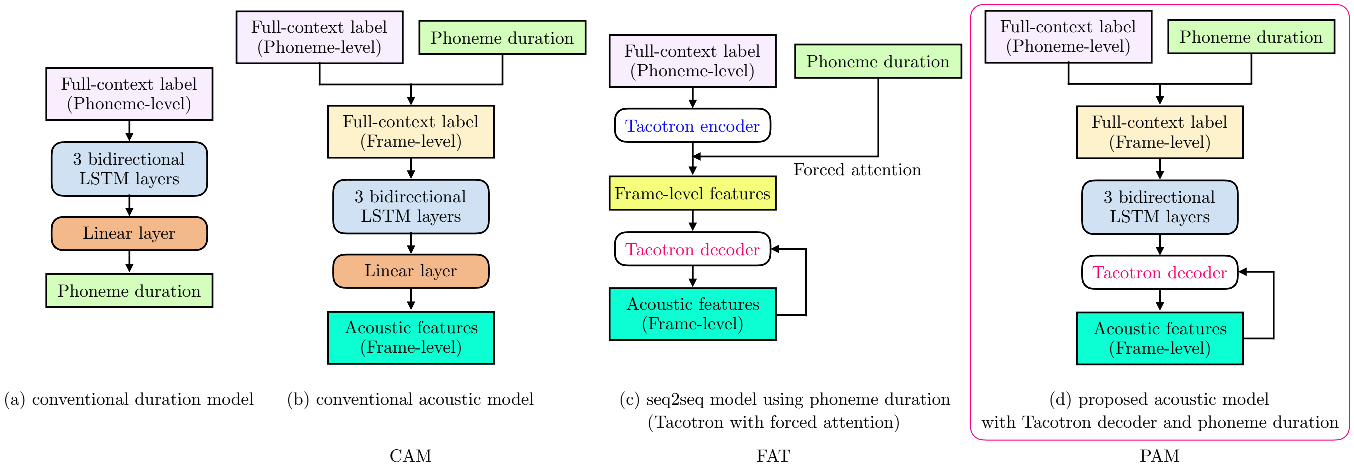
During training, input speech waveform $\boldsymbol{x}$ is converted into Gaussian white noise $\boldsymbol{z}$. Conversely, a speech waveform is generated from Gaussian white noise by the inverse operations in the inference. By introducing the invertible 1×1 convolution and affine coupling layers, the loss function of a WaveGlow vocoder with model parameters $\boldsymbol{\theta}$ conditioned on acoustic feature $\boldsymbol{h}$ is derived as $-\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{\boldsymbol{z}(\boldsymbol{x})^T \boldsymbol{z}(\boldsymbol{x})}{2\sigma^2_{\mathrm{WG}}} - \sum_{j=0} \log \boldsymbol{s}_j(\boldsymbol{x}, \boldsymbol{h}) - \sum_{k=0} \log|\det(\boldsymbol{W}_k)|$, where $\boldsymbol{s}_j$, $\boldsymbol{W}_k$, and σ^2_{WG} are the output coefficients of j-th WaveNet in the affine coupling layers, k-th weighting matrix of the invertible 1×1 convolution layers, and assumed variance of the Gaussian distribution, respectively. According to the sophisticated structure, WaveGlow models can be directly trained and all speech samples can be simultaneously synthesized at once from acoustic features $\boldsymbol{h}$ and Gaussian white noise $\boldsymbol{z}$ [36].

3. SEQUENCE-TO-SEQUENCE ACOUSTIC MODEL WITH FULL-CONTEXT LABEL INPUT

Seq2seq AMs for Japanese, with separately embedded phoneme and accentual-type sequences instead of characters, have been investigated [27]. However, these seq2seq AMs were found to be inferior to conventional pipeline models with full-context label input [27]. This result indicates the importance of full-context labels for pitch accent languages. Additionally, a seq2seq model for Chinese has been improved using full-context label input [31]. Then, a seq2seq AM with full-context label input rather than phoneme and accentual-type sequences was provided by extending the seq2seq architecture of Tacotron 2 [37].

Phoneme-level full-context labels are obtained as linguistic features from a text analyzer. In conventional pipeline TTS frameworks, duration models are first trained from the label vectors, and AMs that predict acoustic features for source-filter vocoders are then trained from the phoneme- or HMM state-aligned frame-level vectors [4, 5]. In [37], a seq2seq-AM that predicts mel-spectrograms for neural vocoders was directly trained from the phoneme-level full-context label vectors. Full-context label vectors typically include past and future 2 contexts. As in [4], these past and future 2 contexts are also reduced in the seq2seq AM because it can access the past and future contexts throughout their bidirectional recurrent connections. The seq2seq AM architecture is almost the same as that of Tacotron 2, except for the input modules (Fig. 1). Compared with Tacotron 2, phoneme-level full-context label vectors extracted from a text analyzer are input to a 1×1 convolution layer instead of character input and an embedding layer. The seq2seq AM is not an end-to-end framework but a language-independent framework because phoneme-level full-context labels for all languages can be directly introduced. Using the seq2seq AM with a WaveGlow vocoder, a high-fidelity real-time neural TTS system for Japanese is realized using a GPU [37]. By introducing full-context label input, it is easier to add new words compared with end-to-end TTS frameworks, and this is more suited to practical TTS systems, although text analyzers are required.

However, this model is also based on an attention mechanism and has the problem that speech samples sometimes cannot be successfully synthesized because of attention prediction errors, as shown in Fig. 3(b). Therefore, to realize high-fidelity practical TTS systems without attention prediction errors, in this paper, Tacotron-based AMs for real-time neural vocoders with phoneme alignment instead of an attention mechanism are investigated by extending the seq2seq AM. In [37], the seq2seq models with AR WaveNet, WaveRNN, and WaveGlow neural vocoders were only evaluated via the listening test and not compared with other AMs.


Full-context label (Phoneme-level)
3 bidirectional LSTM layers
Linear layer
Phoneme duration
Full-context label (Phoneme-level)
Phoneme duration
Full-context label (Frame-level)
3 bidirectional LSTM layers
Linear layer
Acoustic features (Frame-level)
Full-context label (Phoneme-level)
Phoneme duration
Tacotron encoder
Forced attention
Frame-level features
Tacotron decoder
Acoustic features (Frame-level)
Full-context label (Phoneme-level)
Phoneme duration
Full-context label (Frame-level)
3 bidirectional LSTM layers
Tacotron decoder
Acoustic features (Frame-level)

(a) conventional duration model

(b) conventional acoustic model

CAM

(c) seq2seq model using phoneme duration (Tacotron with forced attention)

FAT

(d) proposed acoustic model with Tacotron decoder and phoneme duration

PAM

Fig. 2. Duration and acoustic models for pipeline neural text-to-speech systems with full-context label input: (a) conventional duration model with simple bidirectional LSTM and linear layers [40, 41]; (b) conventional acoustic model with simple bidirectional LSTM and linear layers [3, 5]; (c) Tacotron with forced attention using phoneme alignment [27]; (d) proposed acoustic model with the Tacotron decoder and phoneme duration. Acoustic models (b)–(d) are called "CAM," "FAT," and "PAM" in the experiments, respectively.

4. TACOTRON-BASED ACOUSTIC MODELS USING PHONEME ALIGNMENT

4.1. Conventional DNN-based duration model

In conventional duration-acoustic pipeline models, HMM-based phoneme alignment [39] can be easily achieved when phoneme sequences are given, and the phoneme durations can be relatively easily predicted by conventional duration models compared with acoustic features in the AMs. According to previous work in [42], simple mono-phone HMM-based forced alignment is introduced. Using the phoneme-level full-context labels and phoneme alignment, DNN-based duration models are trained with simple bidirectional LSTM and linear layers [40, 41], as shown in Fig. 2(a).

4.2. Conventional DNN-based acoustic model: CAM

To compare the seq2seq model with full-context label input using a conventional DNN-based AM, a simple bidirectional LSTM-based AM [3, 5] is introduced, as shown in Fig 2(b). In the AM, the frame-level full-context labels are generated from the phoneme-level labels and forced aligned phoneme durations. This AM is called "CAM" in the experiments.

4.3. Tacotron with forced attention based on phoneme alignment: FAT

The direct approach to introduce phoneme duration into the seq2seq model with full-context label input is to replace the attention mechanism with forced alignment based on phoneme duration, as shown in Fig. 2(c). In this approach, phoneme-level full-context label vectors are first encoded into phoneme-level hidden features by the Tacotron encoder. Frame-level hidden features are then generated from the phoneme-level hidden features and forced aligned phoneme durations in the same manner as CAM. The frame-level hidden features are input to the Tacotron decoder. This is equivalent to the scenario of setting the attention weight that corresponds to the phoneme to one and the others to zero for a frame according to the forced aligned phoneme durations. This operation is called "forced attention" in this paper. In the inference, the phoneme durations predicted by the duration model in Fig. 2(a) are used as CAM. In the seq2seq AM in Fig. 1, not only the loss for acoustic features but also the loss for "stop token" are simultaneously minimized. By contrast, this model only minimizes the former loss and is expected to generate more accurate acoustic features without attention prediction errors compared with the seq2seq AM. This model is called "FAT" in the experiments.

4.4. Proposed model with the Tacotron decoder and phoneme duration: PAM

FAT with phoneme and accentual-type sequences and oracle durations was also investigated in [27] to evaluate the accuracy of duration modeling using the attention mechanism. The results in [27] demonstrated that the synthesized speech quality of FAT was worse than those of the seq2seq and pipeline models. This might be because the duplicated frame-level hidden features are input to the Tacotron decoder and it might be redundant for the Tacotron decoder.

To solve the redundancy problem in FAT, an alternative AM is proposed. In the proposed AM, the Tacotron decoder is combined with CAM without a linear layer, as shown in Fig. 2(d). The main difference between FAT and the proposed AM is the method of generating frame-level features. In the proposed AM, the frame-level features are generated before the bidirectional LSTM layers and redundancy is reduced for the Tacotron decoder.

In [27], a shallow AR bidirectional LSTM-based AM (SAR) [9, 53, 54] was introduced to a pipeline model and outperformed the seq2seq models. PAM can be regarded as an extension of SAR with additional pre- and post-nets because two LSTM layers are introduced in PAM instead of a Gaussian mixture model-based mixture density network in SAR.

Therefore, the proposed AM is expected to generate more accurate acoustic features than the other models without attention prediction errors using the sophisticated Tacotron decoder structure. The proposed AM is called "PAM" in the experiments.

Additionally, CAM only with the post-net of the Tacotron decoder and PAM with the Tacotron encoder instead of bidirectional

LSTM layers have also been investigated in preliminary experiments. However, these models do not outperform PAM and were not included in the experiments conducted in the next section. Furthermore, WaveGlow vocoders directly conditioned on frame-level full-context labels with predicted phoneme durations and fundamental frequencies as WaveNet and WaveRNN TTS systems [6, 45, 48] have also been investigated with several types of neural networks[1]. However, high-quality synthesis using these models has not been achieved yet.

5. EXPERIMENTS

5.1. Experimental conditions

To evaluate the seq2seq AM, CAM, FAT, and PAM with full-context label input, experiments were conducted using a Japanese female speech corpus (neutral data) with a sampling frequency of 24 kHz. A total of 25,046 (18 h) and 80 utterances were used as the training set and test set, respectively. Both mel-spectrograms and vocoder features (VF) constructed from the fundamental frequency and mel-cepstra [55] were also evaluated as acoustic features for conventional pipeline SPSS systems. Additionally, the AR single Gaussian (SG)-WaveNet vocoder [49] with VF [37, 50] was included because the speech quality synthesized using a WaveGlow vocoder with VF was not sufficiently high. Both the analysis-synthesis (AS) and TTS conditions were evaluated. In the AS condition, WaveGlow and AR SG-WaveNet vocoders were trained from mel-spectrograms or VF, and the AS waveforms were synthesized with the test sets' acoustic features. For the TTS condition, mel-spectrograms or VF were predicted by the AMs with full-context label input, and the TTS waveforms were synthesized with the predicted acoustic features by WaveGlow and AR SG-WaveNet vocoders trained with the ground-truth acoustic features in the AS condition. Additionally, a STRAIGHT vocoder [2] was included as [37].

Acoustic features:

As acoustic features $\boldsymbol{h}$ for mel-spectrograms, 80-dimensional log-mel-spectrograms were analyzed every 12.5 ms over a Hann window with length 85.3 ms, with a frequency band 125–7,600 Hz and normalized to the range [0, 1], as in [13, 37, 50].

Acoustic features $\boldsymbol{h}$ for VF were analyzed every 5 ms over a Hann window with length 25 ms. Fundamental frequency f_o, analyzed by an NDF algorithm [56] was used in all vocoders with VF [37, 50] and STRAIGHT. Additionally, 35-dimensional mel-cepstra were analyzed from a simple short-time Fourier transform of windowed speech waveforms with warping coefficient $\alpha = 0.46$. In the neural vocoders with VF, $(1 + 1 + 35 =)$ 37-dimensional vectors constructed from continuous logarithmic f_o, a voice/unvoice one-hot vector, and mel-cepstra (normalized to have a zero mean and unit variance) were used.

Duration and acoustic models:

In TTS, the full-context labels were extracted by the text analyzer used in [37, 57]. Although the number of dimensions of the linguistic feature vectors for a frame-wise DNN AM was 483 [57], that for the seq2seq AM was 130 because the past and future 2 contexts were reduced, as described in Section 3. The label vectors were normalized to the range [0, 1].

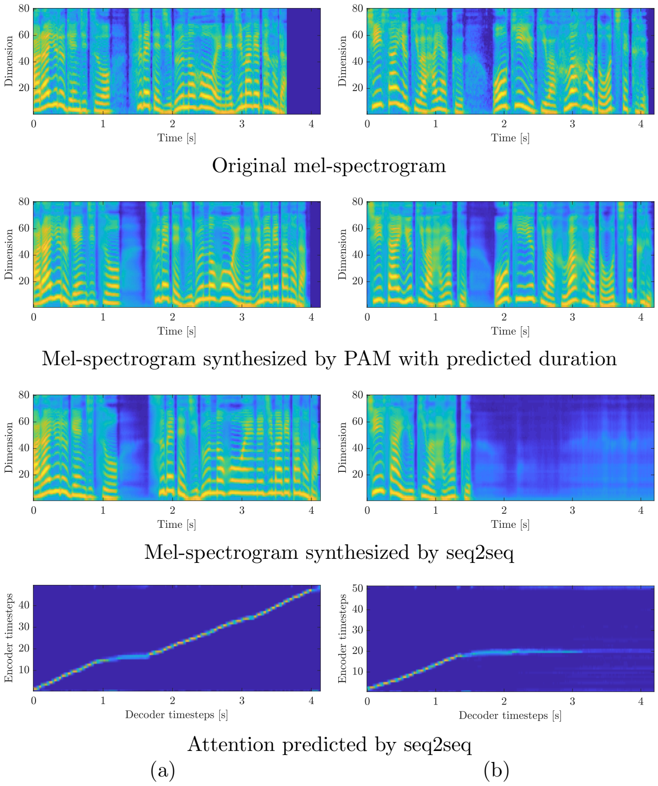


Fig. 3. Results of original mel-spectrograms, mel-spectrograms synthesized by PAM and seq2seq models, and attention weights predicted by the seq2seq model: (a) case in which both models can successfully synthesize; (b) case in which the seq2seq model cannot successfully synthesize because of attention prediction errors.

In the seq2seq AM, the number of output channels of the 1×1 convolution layer was 512. The model parameters of the seq2seq AM after the three convolution layers were the same as those used in Tacotron 2 [13, 37]. The learning rate and batch size were 0.001 and 64, respectively. This model was trained using two NVIDIA Tesla V100 GPUs.

Mono-phone HMM-based forced alignment was introduced using HTK[2] for CAM, FAM, and PAM. The oracle phoneme durations as the numbers of frames for mel-spectrogram (12.5 ms) and VF (5 ms) were then obtained based on forced alignment. In the duration model, the numbers of input, hidden, and output channels were 130, 512, and 1, respectively. The learning rate and batch size were 0.0001 and 64, respectively. The duration model was trained using an NVIDIA Tesla V100 GPU.

In CAM, frame-level full-context label vectors were obtained from phoneme-level vectors with three numerical features for the coarse-coded position of the current frame in the current phoneme and one numerical feature for the duration of the current segment, as in [4][3]. Additionally, to generate smooth parameter trajectories, a maximum likelihood parameter generation (MLPG) algorithm [58] was introduced, except for the voice/unvoice one-hot vector in VF [4]. Then, the numbers of input, hidden, and output channels for mel-spectrograms were $(130 + 4 =)$ 134, 512, and $(80 \times 3 =)$ 240,

[1] This is because the methods of inputting conditioning vectors, such as linguistic features, to WaveNet and WaveRNN were not disclosed in [6, 45, 48].

[2] http://htk.eng.cam.ac.uk

[3] https://github.com/CSTR-Edinburgh/merlin/tree/master/src/frontend/

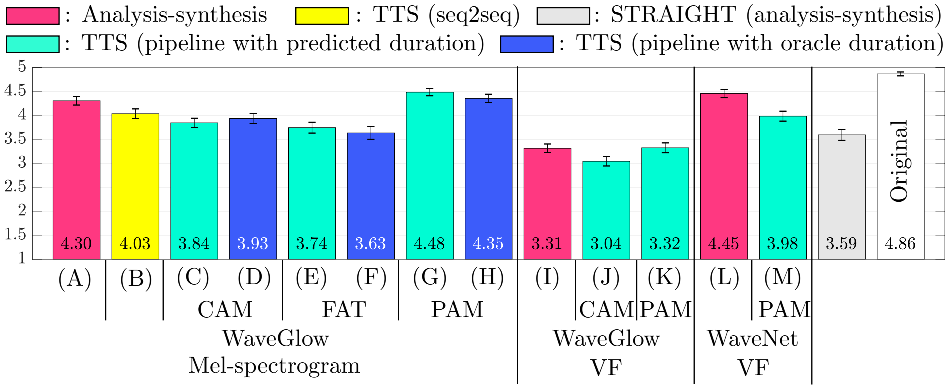


Fig. 4. Results of the MOS test with 15 listening subjects. “TTS” and “VF” denote text-to-speech and vocoder features, respectively. “seq2seq,” “CAM,’ “FAT,” and “PAM” are acoustic models described in Figs. 1 and 2(b)–(d), respectively. Confidence level of the error bars is 95 %.

and those for VF were $(130 + 4 =)$ 134, 512, and $(36 \times 3 + 1 =)$ 109, respectively. The learning rate and batch size were 0.0001 and 64, respectively.

In FAT, all the model parameters were the same as those used in the seq2seq AM. In PAM, the model parameters of the Tacotron decoder were the same as those used in the seq2seq AM and FAT, and the number of hidden channels in the LSTM layer was 512. Four numerical features for obtaining frame-level features and the MLPG algorithm introduced in CAM were not used in FAT and PAM so that they could be directly compared with the seq2seq AM. Therefore, the numbers of input channels of FAT and PAM were both 130, which is the same as those of the seq2seq AM. The learning rate and batch size for FAT and PAM were also 0.001 and 64, respectively. The models of CAM, FAT, and PAM were trained using an NVIDIA Tesla V100 GPU.

WaveGlow and WaveNet neural vocoders and STRAIGHT:

In WaveGlow, all the model parameters were the same as those used in [36, 37]. The batch length and batch size were 16,000 samples and 8, respectively. As in [36,37], the learning rate was initially set to 0.0001 and reduced to 0.00005. In this paper, $\sigma_{\mathrm{WG}} = 1.0$ in the WaveGlow loss function was used for both training and inference.

In AR SG-WaveNet, the numbers of residual and skip channels were both set to 128. Twenty layers (10 dilations × 2 cycles) with a kernel size of 2 were used for the dilated causal convolution layers, as in [37,49,50]. The learning rate, batch length, and batch size were 0.0002, 12,000 samples, and 8, respectively. Similar to [37], a noise shaping filtering [59] was also introduced. A parameter to control noise energy in the formant regions was set to 0.5 [37, 50].

Transposed convolution was applied to the upsampling layers [6] in WaveGlow and AR SG-WaveNet vocoders, and an Adam optimization algorithm [60] was introduced in all the neural network models. WaveGlow and AR SG-WaveNet neural vocoders were trained using four NVIDIA Tesla V100 GPUs.

In STRAIGHT, only the AS condition was evaluated. Both 35-dimensional mel-cepstra, with $\alpha = 0.46$ for the smooth vocal tract spectrum and aperiodicity components were obtained from the original STRAIGHT spectrum and aperiodicity coefficients (1025 dimensions) and the vocoded waveforms were synthesized using the compressed mel-cepstra [37].

5.2. Real-time factor evaluation

The experimental conditions for the combination of AMs and neural vocoders, including the RTFs for the inference measured using an NVIDIA Tesla V100 GPU, are shown in Table 1. All modules were also realized using simple PyTorch[4] implementations, as in [37]. The RTF of the duration model was approximately 0.002 and it was much faster than those of the AMs and WaveGlow vocoders. The results of the RTFs indicate that neural TTS systems with all the AMs and WaveGlow vocoder can synthesize speech waveforms in real-time using a GPU, even with the use of simple PyTorch implementations, although the RTFs of those with the AR SG-WaveNet vocoder were approximately 200, as in [37].

5.3. Attention prediction error for the test set in seq2seq AM

Figure 3 shows examples of the original mel-spectrograms, mel-spectrograms synthesized by PAM and the seq2seq AM, and attention weights predicted by the seq2seq AM. In the seq2seq AM, four utterances out of a total of 80 test set utterances could not be successfully synthesized because of attention prediction errors. The attention prediction errors tend to occur in silent sections, as shown in Fig. 3(b). By contrast, the pipeline models, CAM, FAT, and PAM successfully synthesized all 80 test set utterances with predicted phoneme durations.

5.4. Subjective evaluation

To subjectively evaluate the synthesized speech waveforms, mean opinion score (MOS) tests [61] were conducted. Twenty utterances successfully synthesized by the seq2seq AM from the test set were used as the evaluation set. These were presented through headphones to 15 Japanese adult native speakers without hearing loss (20 utterances × 15 conditions, including the original test set waveforms = 300 utterances).

The MOS results are plotted in Fig. 4. First, PAM for mel-spectrograms with predicted durations outperformed the other conditions, including all AS conditions and the seq2seq model, which also achieved high-quality synthesis, with MOS values over 4.0. These results indicate that PAM with full-context label input and a WaveGlow vocoder for mel-spectrograms were successfully trained. Additionally, the AMs with predicted durations in (E) and (G) slightly

[4] https://pytorch.org

Table 1. Experimental conditions of neural text-to-speech including real-time factors (RTFs) for the inference using a GPU. ”WG” and “WN” in the first term denote the neural vocoders WaveGlow and WaveNet. “MELSPC” and “VF” in the second term denote the acoustic features mel-spectrogram and vocoder features. “AS” and “TTS” in the third term denote the synthesis conditions analysis-synthesis and text-to-speech. “seq2seq,” “CAM,” “FAT,” and “PAM” in the fourth term denote the acoustic models described in Figs. 1 and 2(b)–(d), respectively. TTS conditions (D), (F), and (H) use oracle durations in the inference. Other TTS conditions use predicted durations. “AM RTF” and “Total RTF” denote the real-time factor only for acoustic models and total real-time factor for duration and acoustic models, and a neural vocoder.

Method	AM RTF	Total RTF
(A):WG-MELSPC-AS	-	0.066
(B):WG-MELSPC-TTS-seq2seq	0.063	0.13
(C):WG-MELSPC-TTS-CAM	0.015	0.08
(D):WG-MELSPC-TTS-CAM (OD)	0.015	0.08
(E):WG-MELSPC-TTS-FAT	0.049	0.12
(F):WG-MELSPC-TTS-FAT (OD)	0.049	0.12
(G):WG-MELSPC-TTS-PAM	0.061	0.13
(H):WG-MELSPC-TTS-PAM (OD)	0.061	0.13
(I):WG-VF-AS	-	0.06
(J):WG-VF-TTS-CAM	0.045	0.10
(K):WG-VF-TTS-PAM	0.138	0.20
(L):WN-VF-AS	-	200
(M):WN-VF-TTS-PAM	0.06	200

Table 2. Results of mean square errors of durations predicted by the simple duration model (Fig. 2(a)) and mel-spectrograms predicted by PAM (Fig. 2(d)) with oracle durations for the test set. These models were trained using 18-hour (full), 10-hour, and 5-hour training data with HMM-based phoneme alignment, and using 18-hour (full) training data with simulated incorrect alignment, respectively.

	18 h	10 h	5 h	18 h (incorrect alignment)
Duration	1.14	**1.11**	1.13	1.33
MELSPC	**0.87**	0.92	0.94	0.94

outperformed those with oracle durations in (F) and (H). These results suggest that durations predicted by the simple model were sufficient for the AMs and the predicted durations, which tended to be slightly longer than the original durations, which might have been more suitable for the listening subjects. As expected in Section 4.4 and reported in [27], the synthesized quality of FAT was worse than not only the seq2seq AM but also CAM.

Compared with a WaveGlow vocoder for mel-spectrograms, that for VF could not achieve high-quality synthesis, although the loss score of the WaveGlow vocoder for VF in the training was lower than that for mel-spectrograms. Improving the synthesis quality of a WaveGlow vocoder for VF is future work. Additionally, the synthesis quality of PAM for VF with WaveNet vocoder (M) was worse than that of the AS condition (L). In (M), both the fundamental frequencies and mel-cepstra were simultaneously trained in a single model. To improve PAM for VF, it might be better to separately train them with different networks, as in [9, 27, 54].

5.5. Objective evaluations using PAM

The pipeline approaches, CAM, FAT, and PAM, are based on forced alignment of phoneme durations. To evaluate the prediction accuracies of durations and acoustic features, objective evaluations using PAM with mel-spectrograms were conducted. The relationship between the prediction accuracy and amount of training data, and the mean square errors of durations predicted by the duration model and those of mel-spectrograms predicted by PAM with oracle durations for the test set were evaluated using 18-hour (full), 10-hour, and 5-hour training data with HMM-based phoneme alignment. Additionally, to evaluate the influence of the accuracy of forced alignment, incorrect forced alignment was artificially simulated by randomly shifting the phoneme boundaries of the HMM-based phoneme alignment to -1, 0, and $+1$ at the frame level. Table 2 shows the results of objective evaluations for the test set. The results suggest the following: (1) durations were sufficiently predicted by the model using a relatively small amount of training data, such as 5 hours; (2) a large amount of training data was required to accurately predict acoustic features; and (3) the accuracy of forced alignment was important for accurately predicting both durations and acoustic features.

Consequently, the proposed Tacotron-based AM using phoneme duration based on accurate forced alignment with full-context label input realized a high-fidelity real-time neural TTS system for Japanese with an RTF of 0.13 using a GPU without attention prediction errors compared with the seq2seq AM.

6. FUTURE WORK

Improving the synthesis quality of a WaveGlow vocoder will be investigated because the synthesis quality of a WaveGlow vocoder with mel-spectrograms has not yet reached that of natural speech. Furthermore, the synthesis quality with VF was worse than with the STRAIGHT vocoder. Then, these vocoders will be compared with other real-time neural vocoders, such as parallel WaveNet [48, 49], WaveRNN [45], LPCNet [46], NSF [52], and FloWaveNet [51]. Additionally, sparse WaveRNN and LPCNet will also be investigated for real-time TTS systems with a mobile CPU, as in [45,46]. Furthermore, PAM should be compared with other AMs, such as SAR [53] with sophisticated fundamental frequency prediction modeling [54] and other seq2seq models, including transformer-based TTS [24] and FastSpeech [38]. PAM should also be applied to other languages with only phoneme sequence input instead of full-context label input.

7. CONCLUSIONS

To realize high-quality practical TTS systems without attention prediction errors that are sometimes caused in seq2seq models based on an attention mechanism, in this paper, Tacotron-based AMs with phoneme alignment instead of an attention mechanism were investigated. A seq2seq model with forced alignment instead of attention for phoneme-level sequences, that is, FAT, was investigated and an alternative model with the Tacotron decoder with phoneme duration for frame-level sequences, that is, PAM, was proposed. These models were then compared with the seq2seq AM and the conventional simple bidirectional LSTM-based AM, that is, CAM. The results of experiments with full-context label input using a WaveGlow vocoder indicated that the proposed Tacotron-based AM using phoneme duration realized a high-fidelity TTS system for Japanese with an RTF of 0.13 using a GPU without attention prediction errors compared with the seq2seq AM.

8. REFERENCES

[1] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. 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 Process. Mag.*, vol. 32, no. 3, pp. 35–52, May 2015.

[2] 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 Commun.*, vol. 27, no. 3–4, pp. 187–207, Apr. 1999.

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

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

[5] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," in *Proc. SSW9*, Sept. 2016, pp. 218–223.

[6] A. van den 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," in *Proc. SSW9*, Sept. 2016, p. 125.

[7] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. ICLR*, Apr. 2017.

[8] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. Interspeech*, Aug. 2017, pp. 1118–1122.

[9] X. Wang, J. Lorenzo-Trueba, S. Takaki, L. Juvela, and J. Yamagishi, "A comparison of recent waveform generation and acoustic modeling methods for neural-network-based speech synthesis," in *Proc. ICASSP*, Apr. 2018, pp. 4804–4808.

[10] W. Wang, S. Xu, and B. Xu, "First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention," in *Proc. Interspeech*, Sept. 2016, pp. 2243–2247.

[11] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *Proc. ICLR*, Apr. 2017.

[12] Y. Wang, RJ Skerry-Ryan, D. Stanton, Y. Wu, R. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, Aug. 2017, pp. 4006–4010.

[13] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, RJ Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, Apr. 2018, pp. 4779–4783.

[14] RJ Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *Proc. ICML*, July 2018, pp. 4700–4709.

[15] Y. Wang, D. Stanton, Y. Zhang, RJ Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, July 2018, pp. 5167–5176.

[16] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, Apr. 2018.

[17] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. ICASSP*, Apr. 2018, pp. 4784–4788.

[18] D.-R. Liu, C.-Y. Yang, S.-L. Wu, and H.-Y. Lee, "Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition," in *Proc. SLT*, Dec. 2018, pp. 640–647.

[19] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for TTS synthesis," in *Proc. ICASSP*, May 2019, pp. 5906–5910.

[20] S. Yang, H. Lu, S. Kang, L. Xie, and D. Yu, "Enhancing hybrid self-attention structure with relative-position-aware bias for speech synthesis," in *Proc. ICASSP*, May 2019, pp. 6910–6914.

[21] Y. Cao, X. Wu, S. Liu, J. Yu, X. Li, Z. Wu, X. Liu, and H. Meng, "End-to-end code-switched TTS with mix of monolingual recordings," in *Proc. ICASSP*, May 2019, pp. 6935–6939.

[22] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *Proc. ICASSP*, May 2019, pp. 6945–6949.

[23] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *Proc. ICASSP*, Apr. 2018, pp. 4789–4739.

[24] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Neural speech synthesis with transformer network," in *Proc. AAAI*, Feb. 2019, pp. 6706–6713.

[25] J. Park, K. Han, Y. Jeong, and S. W. Lee, "Phonemic-level duration control using attention alignment for natural speech synthesis," in *Proc. ICASSP*, May 2019, pp. 5896–5900.

[26] T. Kim Y. Lee, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *Proc. ICASSP*, May 2019, pp. 5911–5915.

[27] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *Proc. ICASSP*, May 2019, pp. 6905–6909.

[28] K. Mametani, T. Kato, and S. Yamamoto, "Investigating context features hidden in end-to-end TTS," in *Proc. ICASSP*, May 2019, pp. 6920–6924.

[29] R. Fu, J. Tao, Z. Wen, and Y. Zheng, "Phoneme dependent speaker embedding and model factorization for multi-speaker speech synthesis and adaptation," in *Proc. ICASSP*, May 2019, pp. 6930–6934.

[30] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and RJ Skerry-Ryan, “Semi-supervised training for improving data efficiency in end-to-end speech synthesis,” in *Proc. ICASSP*, May 2019, pp. 6940–6944.

[31] Y. Lu, M. Dong, and Y. Chen, “Implementing prosodic phrasing in Chinese end-to-end speech synthesis,” in *Proc. ICASSP*, May 2019, pp. 7050–7054.

[32] M. Wang, X. Wu, Z. Wu, S. Kang, D. Tuo, G. Li, D. Su, D. Yu, and H. Meng, “Quasi-fully convolutional neural network with variational inference for speech synthesis,” in *Proc. ICASSP*, May 2019, pp. 7060–7064.

[33] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and V. Klimkov, “Effect of data reduction on sequence-to-sequence neural TTS,” in *Proc. ICASSP*, May 2019, pp. 7075–7079.

[34] X. Zhu, Y. Zhang, S. Yang, L. Xue, and L. Xie, “Pre-alignment guided attention for improving training efficiency and model stability in end-to-end speech synthesis,” *IEEE Access*, vol. 7, pp. 65955–65964, 2019.

[35] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, “Hierarchical generative modeling for controllable speech synthesis,” in *Proc. ICLR*, May 2019.

[36] R. Prenger, R. Valle, and B. Catanzaro, “WaveGlow: A flow-based generative network for speech synthesis,” in *Proc. ICASSP*, May 2019, pp. 3617–3621.

[37] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, “Real-time neural text-to-speech with sequence-to-sequence acoustic model and WaveGlow or single Gaussian WaveRNN vocoders,” in *Proc. Interspeech*, Sept. 2019, pp. 1308–1312.

[38] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “FastSpeech: Fast, robust and controllable text to speech,” in *Proc. NeurIPS*, Dec. 2019.

[39] D. T. Toledano, L. A. H. Gómez, and L. V. Grande, “Automatic phonetic segmentation,” *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 617–625, Nov. 2003.

[40] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, “Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks,” in *Proc. Interspeech*, Sept. 2014, pp. 2268–2272.

[41] B. Chen, T. Bian, and K. Yu, “Discrete duration model for speech synthesis,” in *Proc. Interspeech*, Aug. 2017, pp. 789–793.

[42] M. Li, Z. Wu, and L. Xie, “On the impact of phoneme alignment in DNN-based speech synthesis,” in *Proc. SSW9*, Sept. 2016, pp. 196–201.

[43] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, “FFTNet: A real-time speaker-dependent neural vocoder,” in *Proc. ICASSP*, Apr. 2018, pp. 2251–2255.

[44] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, “Improving FFTNet vocoder with noise shaping and subband approaches,” in *Proc. SLT*, Dec. 2018, pp. 304–311.

[45] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, “Efficient neural audio synthesis,” in *Proc. ICML*, July 2018, pp. 2415–2424.

[46] J.-M. Valin and J. Skoglund, “LPCNet: Improving neural speech synthesis through linear prediction,” in *Proc. ICASSP*, May 2019, pp. 5826–7830.

[47] D. Rezende and S. Mohamed, “Variational inference with normalizing flows,” in *Proc. ICML*, July 2015, pp. 1530–1538.

[48] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, “Parallel WaveNet: Fast high-fidelity speech synthesis,” in *Proc. ICML*, July 2018, pp. 3915–3923.

[49] W. Ping, K. Peng, and J. Chen, “ClariNet: Parallel wave generation in end-to-end text-to-speech,” in *Proc. ICLR*, May 2019.

[50] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, “Investigations of real-time Gaussian FFTNet and parallel WaveNet neural vocoders with simple acoustic features,” in *Proc. ICASSP*, May 2019, pp. 7020–7024.

[51] S. Kim, S.-G. Lee, J. Song, J. Kim, and S. Yoon, “FloWaveNet : A generative flow for raw audio,” in *Proc. ICML*, June 2019, pp. 3370–3378.

[52] X. Wang, S. Takaki, and J. Yamagishi, “Neural source-filter-based waveform model for statistical parametric speech synthesis,” in *Proc. ICASSP*, May 2019, pp. 5916–5920.

[53] X. Wang, S. Takaki, and J. Yamagishi, “An autoregressive recurrent mixture density network for parametric speech synthesis,” in *Proc. ICASSP*, Mar. 2017, pp. 4895–4899.

[54] X. Wang, S. Takaki, and J. Yamagishi, “Autoregressive neural F0 model for statistical parametric speech synthesis,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 8, pp. 1406–1419, Aug. 2018.

[55] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, “Mel-generalized cepstral analysis — A unified approach to speech spectral estimation,” in *Proc. ICSLP*, Sept. 1994, pp. 1043–1046.

[56] H. Kawahara, A. de Cheveigné, H. Banno, T. Takahashi, and T. Irino, “Nearly defect-free F0 trajectory extraction for expressive speech modifications based on STRAIGHT,” in *Proc. Interspeech*, Sept. 2005, pp. 537–540.

[57] K. Tachibana, T. Toda, Y. Shiga, and H. Kawai, “Model integration for HMM- and DNN-based speech synthesis using product-of-experts framework,” in *Proc. Interspeech*, Sept. 2016, pp. 2288–2292.

[58] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” in *Proc. ICASSP*, June 2000, pp. 1315–1318.

[59] K. Tachibana, T. Toda, Y. Shiga, and H. Kawai, “An investigation of noise shaping with perceptual weighting for WaveNet-based speech generation,” in *Proc. ICASSP*, Apr. 2018, pp. 5664–5668.

[60] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, May 2015.

[61] ITU-T Recommendation P. 800, *Methods for subjective determination of transmission quality*, 1996.