TRANSFORMER-BASED TEXT-TO-SPEECH WITH WEIGHTED FORCED ATTENTION

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

This paper investigates state-of-the-art Transformer- and FastSpeech-based high-fidelity neural text-to-speech (TTS) with full-context label input for pitch accent languages. The aim is to realize faster training than conventional Tacotron-based models. Introducing phoneme durations into Tacotron-based TTS models improves both synthesis quality and stability. Therefore, a Transformer-based acoustic model with weighted forced attention obtained from phoneme durations is proposed to improve synthesis accuracy and stability, where both encoder–decoder attention and forced attention are used with a weighting factor. Furthermore, FastSpeech without a duration predictor, in which the phoneme durations are predicted by another conventional model, is also investigated. The results of experiments using a Japanese female corpus and the WaveGlow vocoder indicate that the proposed Transformer using forced attention with a weighting factor of 0.5 outperforms other models, and removing the duration predictor from FastSpeech improves synthesis quality, although the proposed weighted forced attention does not improve synthesis stability.

Index Terms— Speech synthesis, sequence-to-sequence model, Transformer, forced alignment, weighted forced attention

1. INTRODUCTION

Text-to-speech (TTS) is an important speech communication technology. A neural network-based autoregressive (AR) generative model called WaveNet outperforms conventional TTS systems [1], and neural vocoders that directly synthesize raw speech waveforms from acoustic features have also been achieved [2, 3].

Additionally, end-to-end TTS approaches directly converting text to raw speech waveforms using sequence-to-sequence (seq2seq) networks based on an attention mechanism have been investigated [4–7]. Although conventional TTS systems separately train duration models and acoustic models (AMs), seq2seq models jointly train them without a pipeline structure. Consequently, Tacotron 2 can realize end-to-end TTS for English with the same quality as natural speech by introducing a long short-term memory (LSTM)-based seq2seq model and AR WaveNet vocoder to solve the pipeline structure and source-filter vocoder problems in conventional TTS [8].

However, Tacotron 2 cannot be directly applied to pitch accent languages, such as Japanese [9], and the inference speed of AR WaveNet is quite slow owing to the AR structure [1–3]. For realizing real-time seq2seq-based TTS systems for pitch accent languages, a real-time neural TTS system for Japanese using a seq2seq model with full-context label input based on Tacotron 2 and a WaveGlow vocoder [10] has been provided [11].[1]

The training speed of Tacotron 2 is slower than that of convolutional neural network (CNN)-based models [6, 7] because it introduces an LSTM structure. To realize seq2seq-based TTS with faster training, an alternative seq2seq model based on the Transformer [14] constructed from feedforward networks (FNNs) with phoneme input has been proposed. It achieved TTS for English with almost the same quality as that of Tacotron 2 [15]. Although the training speed of the Transformer is fast, its inference time is slower than that of Tacotron 2 because it also has recurrent connections in the inference and the network size of the Transformer is larger than that of Tacotron 2 [15].[2]

Although seq2seq models based on an attention mechanism can jointly optimize duration models and AMs and realize high-fidelity synthesis without forced alignment used in conventional TTS models, they run the risk that speech samples cannot always be successfully synthesized because of attention prediction errors. In contrast, this problem does not occur in conventional duration-acoustic pipeline models because the phoneme durations can be almost always accurately predicted.

To avoid the attention prediction error problem, Tacotron-based acoustic models with phoneme alignment instead of an attention mechanism have been investigated [17].[3] In [17], an AM constructed from a bidirectional LSTM and Tacotron decoder with phoneme alignment was proposed (Fig. 1). Unlike Tacotron 2, the proposed pipeline model with full-context label input was able to realize real-time and high-quality neural TTS for Japanese with the WaveGlow vocoder without attention prediction errors.

For simultaneously solving the inference speed and attention prediction error problems in Transformer-based TTS, FastSpeech, which is based on a feedforward Transformer, was recently proposed [20]. In FastSpeech, the phoneme durations are obtained from a teacher Transformer without forced alignment, and the phoneme durations and mel-spectrograms are jointly predicted from the phoneme sequences. FastSpeech can realize stable TTS for English with much faster inference and a quality that is equal to those of Tacotron 2 and the Transformer [20].

For TTS systems, not only inference speed but also training speed are important. Therefore, this paper investigates state-of-the-art Transformer- and FastSpeech-based AMs with full-context label input for pitch accent languages because these models were only investigated with phoneme input for English. Additionally, the phoneme alignment obtained from a conventional hidden Markov model (HMM)-based forced alignment [21] is also introduced to these AMs to improve the synthesis accuracy and stability as a

[1] Seq2seq models for Chinese and English with linguistic features can also improve synthesized speech quality [12, 13].

[2] A reduction factor provided in [5] is also efficient for Transformer-based TTS. It can significantly reduce both the training and inference time, but slightly degrades the synthesis quality [16].

[3] Although the phoneme durations were additionally introduced into seq2seq models to control speech duration and improve attention prediction accuracy in [18, 19], these models are still based on an attention mechanism.

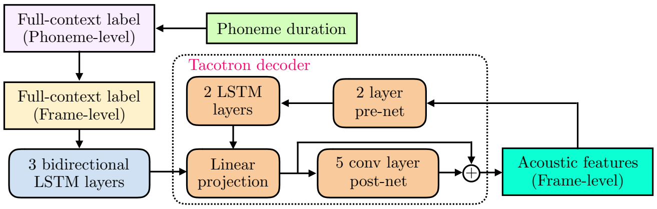

Full-context label (Phoneme-level)
Phoneme duration
Tacotron decoder
Full-context label (Frame-level)
2 LSTM layers
2 layer pre-net
Linear projection
5 conv layer post-net
3 bidirectional LSTM layers
Acoustic features (Frame-level)

Fig. 1. AM based on a bidirectional LSTM and Tacotron decoder with full-context label input using phoneme alignment [17]. This model is called "BLSTM+Taco2dec" in the experiments.

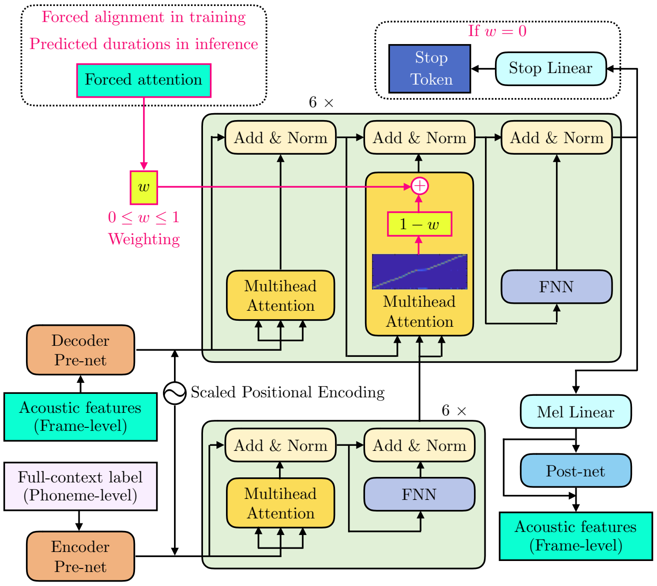


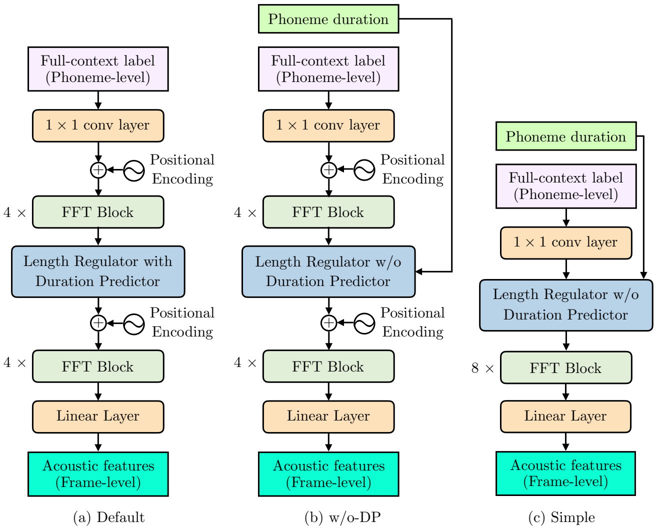
Fig. 2. Proposed Transformer-based AM with weighted forced attention for full-context label input. If $w = 0$, this corresponds to vanilla Transformer-based TTS [15].

pipeline AM with the Tacotron decoder [17]. For the Transformer, a weighted forced attention approach is proposed for simultaneously using both the encoder–decoder alignment and forced alignment with a weighting factor (Fig. 2). To investigate FastSpeech without a phoneme duration predictor network, two types of FastSpeech-based AMs using forced alignment are also compared (Figs. 3(b) and (c)).

2. PROPOSED TRANSFORMER-BASED AM WITH WEIGHTED FORCED ATTENTION

Transformer-based TTS employs a multi-head self-attention mechanism in the encoder and decoder instead of the LSTM structures in Tacotron 2 and can be trained more quickly than Tacotron 2 [15]. In vanilla Transformer-based TTS, phoneme sequences are input to the trainable embedding layer as in Tacotron 2. To investigate Transformer-based TTS with full-context label input, a 1×1 convolution layer is also introduced in the encoder pre-net instead of the embedding layer as in [11, 17].

To directly introduce forced alignment in the Transformer, the encoder–decoder attention is replaced by monotonic attention based on forced alignment, as investigated in Tacotron-based AMs [9, 17, 22]. However, the monotonic forced attention approach for Japanese is inferior to vanilla Tacotron and pipeline models [9,17]. This might be because the duplicated frame-level hidden features are input to the decoder, and they might be redundant for the decoder [17]. To avoid the redundancy problem, the pipeline AM with Tacotron decoder (Fig. 1) was proposed [17]. However, this model also includes LSTM networks.


Phoneme duration
Full-context label (Phoneme-level)
Full-context label (Phoneme-level)
Phoneme duration
1 × 1 conv layer
1 × 1 conv layer
Full-context label (Phoneme-level)
Positional Encoding
Positional Encoding
1 × 1 conv layer
4 ×
FFT Block
4 ×
FFT Block
Length Regulator with Duration Predictor
Length Regulator w/o Duration Predictor
Length Regulator w/o Duration Predictor
Positional Encoding
Positional Encoding
4 ×
FFT Block
4 ×
FFT Block
8 ×
FFT Block
Linear Layer
Linear Layer
Linear Layer
Acoustic features (Frame-level)
Acoustic features (Frame-level)
Acoustic features (Frame-level)

(a) Default (b) w/o-DP (c) Simple

Fig. 3. FastSpeech-based AMs with full-context label input: (a) default model, (b) FastSpeech without a duration predictor, (c) simple FastSpeech without duration predictor and positional encodings. "FFT block" denotes the feedforward Transformer network of [20]. These models do not introduce teacher–student training and HMM-based forced alignment is used in training.

To avoid the redundancy problem in Transformer-based TTS, a weighted forced attention method is proposed, as shown in Fig. 2. This model can simultaneously use both the encoder–decoder attention and forced attention with a weighting factor $0 \leq w \leq 1$. Although the Transformer has multi-head attention, the same weighting factor is applied to all the attention heads in this initial investigation. The vanilla Transformer and one with forced attention correspond to the cases with $w = 0$ and $w = 1$, respectively. When using $w > 0$, this model only minimizes the loss for acoustic features without minimizing the loss for the "stop token," as in [17]. Additionally, when using $0 < w < 1$, this model can avoid the redundancy problem in the case of $w = 1$ by effectively using both the encoder–decoder attention and forced attention. Therefore, this approach should improve encoder–decoder attention accuracy. As a result, the synthesis accuracy and stability can also be improved, as in the Tacotron-based AM with phoneme alignment [17].

3. FASTSPEECH WITHOUT DURATION PREDICTOR

To investigate FastSpeech with full-context label input, a 1×1 convolution layer is also used to replace the embedding layer in vanilla FastSpeech, as shown in Fig. 3(a). All of the network structure except for the 1×1 convolution layer is the same as the structure in [20].

In vanilla FastSpeech, the mean squared error losses for both acoustic features and phoneme durations are simultaneously minimized [20]. However, phoneme durations can be almost accurately predicted by conventional models [17]. Therefore, FastSpeech without a duration predictor, where phoneme durations are predicted by another model based on a teacher Transformer, as in [20], or forced alignment, as in conventional pipeline models, is also investigated (Fig. 3(b)). This model should also improve synthesis accuracy because this model only minimizes the loss for acoustic features, as does the Tacotron-based AM [17] and the proposed Transformer with weighted forced attention. Furthermore, a simple FastSpeech without a duration predictor and positional encodings, as shown in Fig. 3(c), is investigated. In this model, the hidden features from the 1×1 convolution layer are directly input to the length regulator. Then, the encoder and decoder are directly connected as simple feedforward Transformer blocks.

Vanilla FastSpeech is based on teacher–student training, and a teacher Transformer is required for sequence-level knowledge distillation [23] and weight initialization [20]. In this paper, teacher–student training is not introduced in FastSpeech-based AMs in Fig. 3.

4. EXPERIMENTS

4.1. Experimental conditions

To evaluate the proposed Transformer with weighted forced attention and FastSpeech without a duration predictor and compare these AMs with Tacotron-based AMs [17], 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 [11, 17].

The full-context labels were extracted by the text analyzer used in [11, 17, 24]. Although the number of dimensions of the linguistic feature vectors for a frame-wise FNN-based AM was 483 [24], the number used in the experiments was 130 as in [11, 17], because the two past and future contexts can also be reduced for Transformer- and FastSpeech-based AMs with self-attention structures.[4] The label vectors were normalized to the range [0, 1].

Mel-spectrograms are used as acoustic features. 80-dimensional log-mel-spectrograms were analyzed every 12.5 ms over a Hann window with a length of 85.3 ms, with a frequency band of 125–7,600 Hz, and normalized to the range [0, 1], as in [8, 11, 17].

For real-time inference, the WaveGlow vocoder [10] trained with the ground-truth mel-spectrograms was employed to convert the predicted mel-spectrograms to speech waveforms. In WaveGlow, all the model parameters were the same as those used in [10, 11, 17].

In the experiments, 12 AMs with full-context label input were trained and evaluated, as shown in Table 1, which lists the training period, real-time factor (RTF) for AMs and total RTF. In addition, results for a duration model and a WaveGlow vocoder using an NVIDIA Tesla V100 GPU are listed. Models (A) to (C) are seq2seq AMs and (D) to (L) are AMs using phoneme alignment. The number of output channels of the 1×1 convolution layer for the full-context label vector input was 512 for (A) to (C) and (E) to (L).

Model (A) is the Tacotron 2, and the network parameters were the same as those used in [11, 17]. The batch size was 64 and this model was trained using two NVIDIA Tesla V100 GPUs. Although the learning rate was 0.001 in [11, 17], the synthesis quality was improved using a learning rate of 0.0001 in the experiments.

Models (B) and (C) are respectively a Transformer-based AM using FNN layers, as used in [15], and a Transformer-based AM using 1×1 CNN layers instead of FNN layers, as proposed in [20].

Table 1. Experimental conditions of AMs (A) to (L), duration model and WaveGlow vocoder including real-time factors (RTFs) for inference using a GPU and PyTorch. (B) and (C) Transformer-based AMs with FNN and 1×1 CNN layers, respectively. (D) A simple bidirectional-LSTM AM and (E) the Tacotron-based AM in Fig. 1. (F) to (I) The proposed Transformer-based AMs using weighted forced attention with $w = 0.2$, 0.5, 0.7 and 1.0 in Fig. 2, respectively. (J) to (L) The FastSpeech-based AMs in Figs. 3(a) to (c), respectively. "TP," "AM RTF," and "Total RTF" denote the training period, real-time factor (only for AMs), and total real-time factor for duration and AMs, and WaveGlow vocoder, respectively.

Method	TP (days)	AM RTF	Total RTF
(A):Tacotron 2	24	0.063	0.13
(B):TF (FNN)	6	0.55	0.62
(C):TF (Conv1D)	6	0.55	0.62
(D):BLSTM	3	0.015	0.12
(E):BLSTM+Taco2dec	12	0.061	0.13
(F)-(I):TF-WFA	6	0.55	0.62
(J):FS (Default)	6	0.004	0.070
(K):FS (w/o-DP)	6	0.004	0.072
(L):FS (Simple)	6	0.004	0.072
Duration model	2	-	0.002
WaveGlow vocoder	30	-	0.066

Models (D) and (E) are respectively a simple bidirectional LSTM-based AM [26] and the Tacotron-based AM in Fig. 1. The network parameters were the same as those used in [17].

Models (F) to (I) are the proposed Transformer-based AMs using weighted forced attention with weighting factors of $w = 0.2$, 0.5, 0.7, and 1.0, respectively. In (F) to (I), FNN layers were used, as in (B). In the Transformer-based AMs (B), (C), and (F) to (I), eight heads were used in the multi-head attention and six layers were used in the encoder and decoder blocks. The learning rate was 0.00005. All other network parameters except for those of the 1×1 convolutional layers were the same as those used in [15].

Models (J) to (L) are FastSpeech, FastSpeech without a duration predictor, and a simple Fastspeech without a duration predictor and positional encodings, as shown in Fig. 3, respectively. In these models, 1×1 CNN layers were also used to replace the FNN layers in the feedforward Transformer blocks, following [20]. The learning rate was 0.00005. All network parameters except for those of the 1×1 convolutional layers were the same as those used in [20]. Transformer- and FastSpeech-based AMs were trained with eight NVIDIA Tesla V100 GPUs.

Mono-phone HMM-based forced alignment was employed in (D) to (L). The phoneme durations were then obtained based on the forced alignment. A simple bidirectional LSTM-based duration model [27] was also compared, as in [17].

All the training steps and inferences were implemented using PyTorch [28]. An Adam optimization algorithm [29] was introduced in all the neural network models. The training period and RTF for the duration model and WaveGlow are also provided in Table 1.

To subjectively evaluate the speech waveforms synthesized by these TTS models, mean opinion score (MOS) tests [30] were conducted. As in [11, 17], the analysis-synthesis conditions of WaveGlow and STRAIGHT [31] vocoders were also included. Twenty utterances successfully synthesized by all the AMs from the test set

[4] Tacotron 2 for Japanese with full-context label input including the past and future contexts was also investigated in [25].

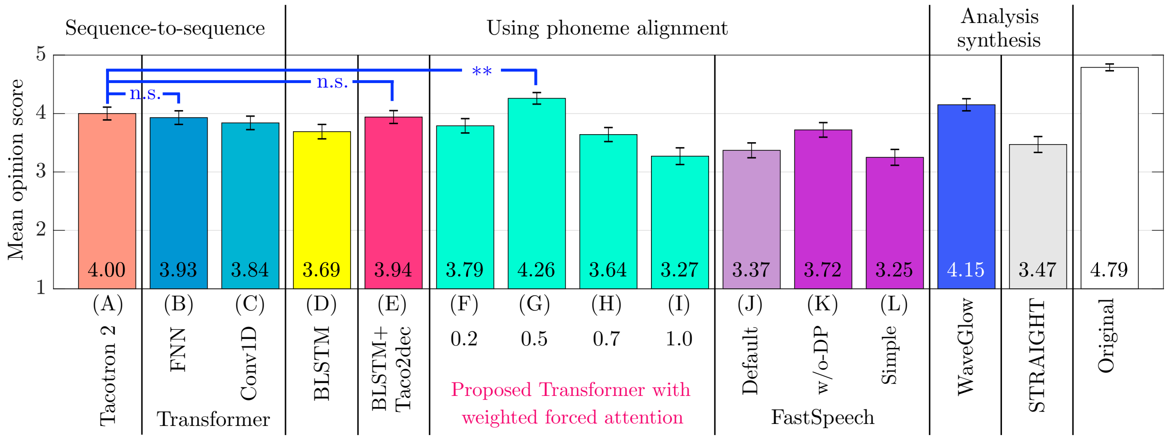


Fig. 4. Results of the MOS test with 20 listening subjects. The confidence level of the error bars is 95 %.

were used as the evaluation set, as in [20]. These were presented through headphones to 20 Japanese adult native speakers without hearing loss (20 utterances × 15 conditions, including the original test set waveforms = 300 utterances).

4.2. Results and discussion

The MOS results are plotted in Fig. 4. First, the Transformer- and FastSpeech-based AMs with full-context label input for pitch accent languages can be successfully trained faster than Tacotron-based AMs (A) and (E). However, the synthesis speed of the Transformer-based AMs was slower than that of other AMs although the RTF was kept below 1.0 by using a GPU.

Although (E) outperformed (A) and four test set utterances out of 80 could not be successfully synthesized by (A) in [17], the performance of (A) could reach that of (E) and all 80 test set utterances could be successfully synthesized by (A) by changing the learning rate to 0.0001 in the experiments. Additionally, (B) could also realize the same synthesis quality as (A) and (E).

The most important contribution of this paper is that the proposed Transformer-based AM using weighted forced attention with $w = 0.5$ (G) significantly outperformed other AMs.[5] However, other weighting conditions (F), (H), and (I) could not outperform (B) (with $w = 0$). This result indicates the redundancy of the encoder input is increased when $w > 0.5$ and the effect of forced attention cannot be sufficiently used when $w < 0.5$. Therefore, it is better to use the same weights for the encoder–decoder attention and forced attention to effectively use both types of attention.

Although FastSpeech-based AMs (J) to (L) could synthesize speech waveforms faster than other AMs, their synthesis qualities were not high. This might be because the sequence-level knowledge distillation [23] and weight initialization using a teacher Transformer [20] were not employed. However, FastSpeech without a duration predictor (K) significantly outperformed FastSpeech with a duration predictor (J). Therefore, the use of conventional duration models is also effective for FastSpeech. In contrast, the simple model (L) was not effective.

All 80 test set utterances could be successfully synthesized by Tacotron 2 (A), LSTM-based AMs (D) and (E), and FastSpeech-based AMs (J) to (K). However, 6–10 speech waveforms out of 80 test set utterances synthesized by each Transformer-based AM included word skipping and repeats, even though the proposed weighted attention with $w = 1.0$ was used. This is because the word skipping and repeating problem occurs if the self-attention weights of the encoder and decoder are not diagonal even when the encoder–decoder attention weights are diagonal. Unsuccessfully synthesized utterances differed according to the models. Further analysis of the error tendencies and additional investigation, such as modification of the self-attention mechanism, will be required in future work.

Consequently, the proposed Transformer-based TTS using weighted forced attention with $w = 0.5$ can improve the synthesis quality, although the synthesis stability cannot be improved. Additionally, FastSpeech without a duration predictor combined with another duration model can also improve the synthesis quality.

5. FUTURE WORK

The proposed Transformer-based AM with weighted forced attention should be further investigated to improve the synthesis stability to the level of FastSpeech. Although fixed weighting factors were introduced in this initial investigation, trainable weighting factors could also be introduced for multi-head attention weights. Additionally, the proposed weighted forced attention could also be directly introduced to Tacotron 2. Furthermore, teacher–student training should also be introduced to FastSpeech with full-context label input for pitch accent languages to improve the synthesis accuracy.

6. CONCLUSIONS

This paper investigated Transformer- and FastSpeech-based neural TTS with full-context label input to realize training that is faster than that of Tacotron-based AMs. Additionally, Transformer-based AM with weighted forced attention was proposed to improve the synthesis accuracy and stability. FastSpeech without a duration predictor was also investigated. The results of the experiments suggest that the proposed Transformer using weighted forced attention with $w = 0.5$ significantly outperforms other AMs, and FastSpeech without duration prediction could realize higher synthesis accuracy than FastSpeech with duration prediction, although the proposed weighted forced attention did not improve the synthesis stability.

[5] Like the Tacotron-based AM in [17], (B) slightly outperformed the analysis-synthesis condition of WaveGlow. This might be because the predicted durations tended to be slightly longer than the original durations, and this might have been more suitable for the listening subjects.

7. REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] H. Guo, F. K. Soong, L. He, and L. Xie, "Exploiting syntactic features in a parsed tree to improve end-to-end TTS," in *Proc. Interspeech*, Sept. 2019, pp. 4460–4464.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need,," in *Proc. NIPS*, Dec. 2017, pp. 5998–6008.

[15] 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.

[16] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. Yalta, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on Transformer vs RNN in speech applications," in *Proc. ASRU*, Dec. 2019, pp. 449–456.

[17] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, "Tacotron-based acoustic model using phoneme alignment for practical neural text-to-speech systems," in *Proc. ASRU*, Dec. 2019, pp. 214–221.

[18] 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.

[19] 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.

[20] 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, pp. 3165–3174.

[21] 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.

[22] O. Watts, G. Eje Henter, J. Fong, and C. Valentini-Botinhao, "Where do the improvements come from in sequence-to-sequence neural TTS?," in *Proc. SSW10*, Sept. 2019, pp. 217–222.

[23] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proc. EMNLP*, Nov. 2016, pp. 1317–1327.

[24] 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.

[25] T. Fujimoto, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Impacts of input linguistic feature representation on Japanese end-to-end speech synthesis," in *Proc. SSW10*, Sept. 2019, pp. 166–171.

[26] 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.

[27] 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.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison andA. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, Dec. 2019, pp. 8024–8035.

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

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

[31] 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.