Transformer-based Acoustic Modeling for Streaming Speech Synthesis

Chunyang Wu, Zhiping Xiu, Yangyang Shi, Ozlem Kalinli, Christian Fuegen, Thilo Koehler, Qing He

Facebook AI

{chunyang, zhipingxiu, yyshi, okalinli, fuegen, tkoehler, qinghe}@fb.com

Abstract

Transformer models have shown promising results in neural speech synthesis due to their superior ability to model long-term dependencies compared to recurrent networks. The computation complexity of transformers increases quadratically with sequence length, making it impractical for many real-time applications. To address the complexity issue in speech synthesis domain, this paper proposes an efficient transformer-based acoustic model that is constant-speed regardless of input sequence length, making it ideal for streaming speech synthesis applications. The proposed model uses a transformer network that predicts the prosody features at phone rate and then an Emformer network to predict the frame-rate spectral features in a streaming manner. Both the transformer and Emformer in the proposed architecture use a self-attention mechanism that involves explicit long-term information, thus providing improved speech naturalness for long utterances. In our experiments, we use a WaveRNN neural vocoder that takes in the predicted spectral features and generates the final audio. The overall architecture achieves human-like speech quality both on short and long utterances while maintaining a low latency and low real-time factor. Our mean opinion score (MOS) evaluation shows that for short utterances, the proposed model achieves a MOS of 4.213 compared to ground-truth with MOS of 4.307; and for long utterances, it also produces high-quality speech with a MOS of 4.201 compared to ground-truth with MOS of 4.360.

Index Terms: speech synthesis, transformer, emformer, streaming

1. Introduction

The progress in neural text-to-speech (TTS) technologies has led to a substantial breakthrough in audio quality of state-of-the-art TTS systems. The capability of synthesizing human-like voice finds excellent uses in a wide range of today's and future applications, from enhanced gaming experience of TTS-capable virtual reality platforms to lifestyle improvement with TTS-enabled voice assistant gadgets. On the other hand, such wide TTS applications also bring deployment challenges in many complex but essential scenarios. First, speech synthesis applications tend to be real-time, such as when used for interacting with end-users, and therefore latency can be critical. Second, it is the most desirable to perform all the TTS computations on-device in today's mobile world, which puts a tight constraint on computing complexity. Third, compatibility with streaming services is also essential so that the TTS does not require a large buffering overhead.

Practical TTS systems usually consist of two stages: an acoustic model that predicts the prosody and spectral features followed by a neural vocoder that generates the audio waveform. Recent works on acoustic modeling [1–4] have achieved significant improvements in naturalness, accompanied with neural vocoder advancements [5–10] that have enabled real-time TTS with human-like audio quality in today's applications. However, practical deployment scenarios usually bring further challenges, especially in resource-limited and real-time applications. In this paper, we focus on high quality streaming acoustic modeling with reduced complexity. Existing acoustic models such as Tacotron2 [1, 2] use the Bi-directional Long Short-term Memory (BLSTM) recurrent networks. These approaches usually suffer from inefficiency in inference and cannot effectively model long-term dependencies, resulting in a poor quality on long speech. On the other hand, attention based models such as FastSpeech [3] and [4] deliver state-of-the-art quality in modeling speech prosody and spectral features, but they are unsuitable for streaming TTS applications because attention computation is parallel over the full utterance context. Our previous work [11] proposes a streamable model architecture that computes a global attention context using low-rate features such as sentence-level, phrase-level, and word-level features and then sequentially predicts the spectral frames with an LSTM network. While this architecture delivers efficient computation and high audio quality, the prosody naturalness degrades for long utterances (e.g., > 20s) due to LSTM's inability to model strong long-term dependency.

Transformers-based models [12] have shown promising results in a wide range of sequential tasks, including natural language processing [13, 14] and speech [4, 15, 16]. Conventional recurrent neural networks, e.g., long short-term memory (LSTM) [17], use a hidden state to pass temporal information across the input sequence. In comparison, transformer networks introduce a multi-head self-attention mechanism that connects arbitrary pairs of positions in the input sequences directly, enabling long-term dependency and global information to be modeled explicitly. However, two major issues make the generic transformer model impractical for streaming tasks. First, it needs to access the complete input sequence before it can start generating output; Second, the computational cost and memory usage grow quadratically as the input sequence length increases. If the input is sufficiently long, transformers will result in high latency and high real-time factor (RTF). In order to make transformers streamable, several strategies have been investigated. The time-restricted self-attention approach [18–20] restricts the computation of attention only on the past frames and a fixed number of future frames. The block processing strategy used in [21] chunks the input utterances into segments, and self-attention performs on each segment. In this way, the time and space cost will not grow quadratically. Based on block processing, a recurrent connection can be further introduced to explicitly extracts history embeddings from the previous segment, including transformer-XL [22], augmented-memory transformer [23, 24] and efficient memory transformer [25] (Emformer).

In this paper, we propose a transformer-based acoustic model for streaming speech synthesis. This work is an extension to our previous work on the multi-rate attention model [11]

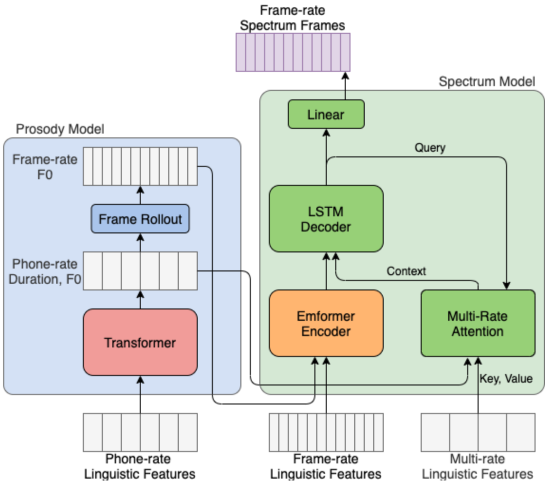

Figure 1: *Transformer-based acoustic model:* The prosody model adopts a transformer network to predict the phone-rate duration and F0 features, which are then rolled-out using repetition to the frame rate. The spectrum model uses the predicted prosody features, along with multi-rate linguistic features to predict the frame-rate spectral features. In the spectrum model, a streamable Emformer network is used as encoder, and decoding is performed on the encoder output and a global multi-rate attention context computed using low-rate linguistic features [11].

for streaming speech synthesis and the Emformer model [16,25] originally designed for streaming speech recognition. The acoustic model is composed of a prosody model and a spectrum model. The prosody model is modeled as a transformer network [12]; the spectrum model is modeled as a streamable Emformer network [25]. These transformer-based architectures are leveraged to improve the TTS quality, particularly on long speech. Our MOS and RTF evaluations show that the proposed method synthesizes high-quality speech while remaining a constant inference speed independent of input lengths. It outperforms a baseline acoustic model that uses a BLSTM with content-based global attention [26] as prosody model and an LSTM with multi-rate attention [11] as spectrum model. On long utterances, it can preserve the high speech quality close to the ground truth.

The rest of this paper is organized as follows: we present the model architecture in Section 2. Section 3 discusses the experimental results, followed by a summary in Section 4.

2. Transformer-based Acoustic Model

The diagram of the transformer-based acoustic model is illustrated in Figure 1. It consists of a prosody model and a spectrum model. The prosody model unrolls the phone-rate linguistic features to predict the phone-rate prosody features: duration and f_0. Then, the spectrum model takes in the linguistic features and the prosody features from the prosody model to predict the frame-rate spectral features in a streaming manner. The spectrum model is a multi-rate attention model [11]. The encoder-decoder module processes frame-rate input features to generate the spectral features. The multi-rate attention bootstraps the decoder with a frame-rate context vector, using the decoder's hidden state as query and the multi-rate linguistic features as key and value. In this work, to leverage the transformer architecture, we investigate the transformer network [12] as the prosody model and the Emformer network [25] as the encoder in the spectrum model. The self-attention mechanism in transformer and Emformer involves more long-term information, thus has the potential to improve the TTS quality, particularly on long utterances.

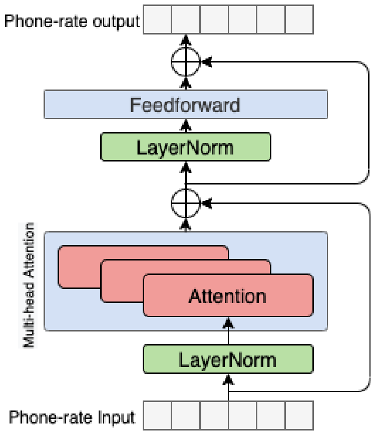

Figure 2: *Transformer architecture in prosody model:* It consists of a multi-head attention block and a feed-forward block. Layer normalization and residual connection are applied to each block.

2.1. Transformer-based prosody model

The prosody model is modeled as a transformer network. Figure 2 shows the transformer topology. A transformer layer consists of a multi-head attention [12] block and a feed-forward block composed of two linear transformations and a nonlinear activation function. To allow stacking multiple transformer layers, layer normalization [27] and residual connection are added to each block.

The multi-head attention block contains multiple self-attention components that individually operate the input sequence. Given the input sequence $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$ where $\boldsymbol{x}_t \in \mathbb{R}^D$ and T stands for the input sequence length, one self-attention first projects the input respectively to query $\boldsymbol{Q}$, key $\boldsymbol{K}$ and value $\boldsymbol{V}$,

$$\boldsymbol{Q} = \boldsymbol{W}_{\mathrm{q}}\boldsymbol{X}, \quad \boldsymbol{K} = \boldsymbol{W}_{\mathrm{k}}\boldsymbol{X}, \quad \boldsymbol{V} = \boldsymbol{W}_{\mathrm{v}}\boldsymbol{X} \tag{1}$$

where $\boldsymbol{W}_{\mathrm{q}}, \boldsymbol{W}_{\mathrm{k}}, \boldsymbol{W}_{\mathrm{v}}$ are parameters to optimize. Next, dot product is applied to get an attention distribution over $\boldsymbol{Q}$ and $\boldsymbol{K}$: for position t in $\boldsymbol{Q}$, a distribution $\boldsymbol{\alpha}_t$ is given as

$$\alpha_{t\tau} = \frac{\exp(\frac{1}{\sqrt{D}}\boldsymbol{Q}_t^{\top}\boldsymbol{K}_{\tau})}{\sum_{\tau'}\exp(\frac{1}{\sqrt{D}}\boldsymbol{Q}_t^{\top}\boldsymbol{K}_{\tau'})}. \tag{2}$$

The output of self-attention is computed via

$$\boldsymbol{z}_t = \mathrm{attn}(\boldsymbol{Q}_t, \boldsymbol{K}, \boldsymbol{V}) = \sum_{\tau}\alpha_{t\tau}\boldsymbol{V}_{\tau}. \tag{3}$$

The self-attention outputs are finally concatenated and linearly transformed to form the output of the multi-head attention block.

The transformer model requires $\mathcal{O}(T^2)$ time to compute the self-attention, which grows quadratically with input length. However, in this paper's TTS framework, the prosody model processes the data in phone rate. The length of phone-rate data is sufficiently small. The computation can be negligible even on long sequences, compared with the spectrum model that processes data in frame rate. Therefore, we use this non-streaming transformer architecture for the prosody model. As operating on the whole input sequence, it can model better prosody for long utterances. The streaming challenge of the proposed acoustic model mainly depends on the frame-rate spectrum model. It will be discussed in the following section.

2.2. Emformer-based spectrum model

The spectrum model processes the frame-rate feature sequence. For streaming reasons, it is required to incrementally process the data. In this work, the Emformer [25] architecture is used as the encoder model of the spectrum model. It is a streamable transformer variant that processes sequence data in a block processing [21] fashion to achieve controllable latency. The Emformer model resolves the transformer's quadratic computation complexity via an augmented memory mechanism [23]. The long-range history is distilled into an augmented memory bank to alleviate the heavy computation on attention.

Figure 3 shows the Emformer architecture. On the n-th Emformer layer, the input sequence $\boldsymbol{X}$ is chunked into multiple non-overlapping fixed-size segments. For the i-th segment $\boldsymbol{C}_i^n$, to alleviate the boundary effects, it is processed together with left context $\boldsymbol{L}_i^n$ and right context $\boldsymbol{R}_i^n$. The right context $\boldsymbol{R}_i^n$ is directly used. For efficiency, the left context $\boldsymbol{L}_i^n$ is indirectly introduced via key $\mathbf{K}_{i,L}^n$ and value $\mathbf{V}_{i,L}^n$ which has already been calculated in the previous segments. The memory bank $\boldsymbol{M}_i^n$ is internally maintained in the Emformer. It stores distilled embeddings for the processed segments. Usually, the memory bank keeps a fixed length. Only a fixed number of the most recent slots are stored in the memory. In this way, long-range information can be efficiently captured according to the memory instead of a large number of raw input frames. The attention part of Emformer is computed as follows,

$$\boldsymbol{Q}_i^n = [\boldsymbol{C}_i^n, \boldsymbol{R}_i^n], \tag{4}$$

$$\boldsymbol{K}_i^n = [\boldsymbol{W}_{\mathrm{k}}\boldsymbol{M}_i^n, \boldsymbol{K}_{i,L}^n, \boldsymbol{W}_{\mathrm{k}}\boldsymbol{C}_i^n, \boldsymbol{W}_{\mathrm{k}}\boldsymbol{R}_i^n], \tag{5}$$

$$\boldsymbol{V}_i^n = [\boldsymbol{W}_{\mathrm{v}}\boldsymbol{M}_i^n, \boldsymbol{V}_{i,L}^n, \boldsymbol{W}_{\mathrm{v}}\boldsymbol{C}_i^n, \boldsymbol{W}_{\mathrm{v}}\boldsymbol{R}_i^n], \tag{6}$$

where the memory bank $\boldsymbol{M}_i^n$ is recursively accumulated in processing successive segments and layers[1],

$$\boldsymbol{s}_{i-1}^{n-1} = \mathrm{mean}(\boldsymbol{C}_{i-1}^{n-1}), \tag{7}$$

$$\boldsymbol{m}_{i-1}^{n} = \mathrm{attn}(\boldsymbol{s}_{i-1}^{n-1}; \boldsymbol{K}_{i-1}^{n-1}, \boldsymbol{V}_{i-1}^{n-1}) \tag{8}$$

$$\boldsymbol{M}_i^n = [\boldsymbol{M}_{i-1}^n, \boldsymbol{m}_{i-1}^n]. \tag{9}$$

Emformer processes the data with a $\mathcal{O}(T)$ time complexity, theoretically giving a constant RTF. When running on frame rate, it is able to operate efficiently regardless of the sequence length. In addition, the segment size of Emformer determines how many frames to process in each update, which can be viewed as an algorithmic latency. The latency can directly be controlled via the segment size.

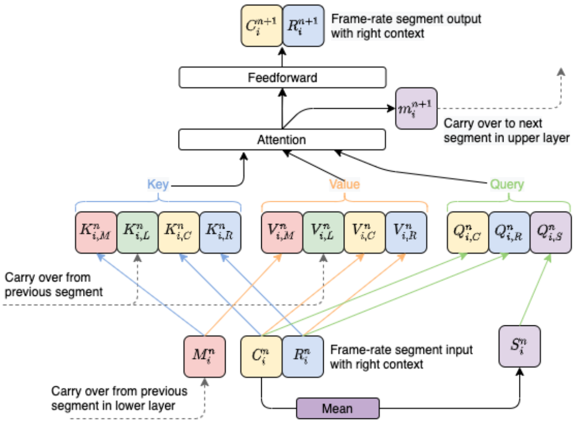

Figure 3: *Emformer architecture in spectrum model:* The attention block is described in detail. Each input segment (yellow) is processed together with right context (blue), and left context (green). Left context is introduced via the key and value carried over from previous segments. Memory (red) is recursively accumulated between successive layers and segments, described in the purple path. Layer normalization and residual connection are omitted.

3. Experiments

3.1. Dataset and feature extraction

The dataset used in our experiments was recorded in a voice production studio by contracted professional voice talents. The training set consists of about 110K utterances from six speakers, approximately 90 hours of data with a 24kHz sampling rate. For evaluation, we excluded testing utterances from different speakers in the training data. In the MOS experiments, two evaluation sets were prepared according to the audio length. The "normal" set includes 112 utterances with audio lengths ranging from 1 to 10 seconds. The "long" set includes 58 utterances with lengths ranging from 15 to 40 seconds. The long set was used to evaluate the modeling ability in capturing long-form information. In the RTF experiments, a separate set was prepared, containing utterances with lengths ranging from 1 to 60 seconds.

The feature used in the experiment includes the linguistic feature, phone-level duration feature and f_0 feature. The phone-level duration feature was extracted with an unsupervised alignment algorithm using the softDTW loss [28]. The f_0 feature was extracted for each frame using spectrum analysis. More details can be found in [11].

3.2. Experiment setup

The baseline acoustic model is of a similar topology as Figure 1. Its prosody model is a one-layer BLSTM of 256 hidden units with content-based global attention [26]. Its spectrum model is a multi-rate attention model using a one-layer LSTM encoder of 512 hidden units and a one-layer LSTM decoder of 512 hidden units. The baseline model is described with more details in [11]. We refer to the baseline prosody model as "BLSTM with self-attention", the baseline spectrum model as "multi-rate attention", in the following discussion.

The proposed transformer-based acoustic model used in evaluation is presented as follows. The transformer-based prosody model contains six transformer layers. On each transformer layer, the input and output have 512 nodes. The multi-head attention block contains eight attention heads, and the output dimension is 512. In the feedforward block, the ReLU activation function is used, and the in-block hidden dimension is 256. Between each pair of fully connected layers, a dropout of 0.5 is applied. The Emformer-based spectrum model is a multi-rate attention model using an Emformer encoder. The decoder

[1] Emformer uses successive layers to compute memory to improve the training efficiency. More details are discussed in [25].

System	Prosody	Spectrum	Normal	Long
Groundtruth	–	–	4.307 ± 0.037	4.360 ± 0.044
Baseline [11]	BLSTM with self-attention [26]	Multi-rate attention [11]	4.173 ± 0.042	4.019 ± 0.055
Ours-1	Transformer	Multi-rate attention	4.174 ± 0.042	4.107 ± 0.052
Ours-2	BLSTM with self-attention	Emformer with multi-rate attention	4.192 ± 0.041	4.034 ± 0.053
Ours-3 (best)	Transformer	Emformer with multi-rate attention	**4.213 ± 0.042**	**4.201 ± 0.048**

Table 1: *MOS with 95% confidence intervals:* the proposed model leads to small improvements over the baseline model for normal length audio (0-15 seconds) and more significant improvements for long audio (20-40 seconds).

is a one-layer LSTM decoder of 512 hidden units. The Emformer encoder contains 2 Emformer layers. On each Emformer layer, the input and output have 256 nodes. The attention block is composed of 8 attention heads and the output dimension is 256. The feedforward block uses the ReLU activation function, and the hidden dimension is 256. The length of the Emformer memory bank is 4. Due to the latency constraints, the segment size is fixed as 32 frames in this paper. Left context and right context are respectively set to 12 frames. The spectrum model ultimately predicts a 19-dim spectrum feature vector consisting of a 13-dim MFCC feature, a 1-dim f_0 feature, and a 5-dim periodicity feature. The loss function to train the spectrum model is a weighted sum of the mean square errors of MFCC, f_0 and periodicity. The MFCC, f_0 and periodicity loss weights are set to 1.0, 10.0 and 5.0, respectively.

In training, both the prosody and spectrum models were optimized using the Adam [29] optimizer with an initial learning rate of 10^{-4} and a decay factor of 0.95 for every 45K updates. The prosody model was trained in 500k updates. The spectrum model was trained in 700K updates. All models were trained using 1 Nvidia V100 GPU. To evaluate RTFs, a single core on the Intel(R) Xeon(R) 2.0GHz CPU was used.

In inference, the baseline and proposed acoustic models were used to predict spectral features. The synthesized speech was finally generated through a WaveRNN [5] conditional neural vocoder with hidden dimension 1024.

3.3. Results

We conducted the mean opinion score study on the two evaluation sets. Each MOS test had 400 participants who rated each sample between 1-5 (1:bad - 5:excellent). Table 1 summarizes the results of the MOS studies. Overall, the proposed transformer acoustic models achieved better MOS scores than the baseline model[2]. In the comparison of prosody models (ours-1 and baseline), they achieved similar performance on the normal set, however the transformer outperformed the BLSTM with self-attention on the long set. This indicates transformer can estimate a better prosody for long speech. The prosody and spectrum models showed some level of complementarity; the TTS quality was further improved when using transformer architectures on both models (ours-3 *v.s.* our-1 & 2). The baseline showed a quality degradation on the long set. However, the proposed model (ours-3) was only influenced slightly. The robustness of the proposed model on the long set demonstrates that transformer-based architecture can capture long-term information in speech synthesis and deliver long but natural TTS speech that is close to human speech.

We then evaluated the inference speed by measuring the real-time factor, which is a key metric for TTS products. The RTF is defined as $\frac{\text{total synthesize time}}{\text{audio length}}$. As discussed in Section 2.1, the computation on the prosody model can be negligible, we fixed the transformer prosody model and compared the baseline and our Emformer spectrum models. Also, we included a transformer spectrum model to illustrate how the RTF grows on a non-streamable model concerning the input length. Figure 4 compares the log-scale RTF with audio length ranging from 1 to 60 seconds. Similar to the baseline, the streamable Emformer model approximately stays a constant RTF regardless of audio length. Its RTF is only slightly higher on short utterances. The non-streamable transformer model operates on the complete input sequence, which the RTF curve shows a quadratic growth.

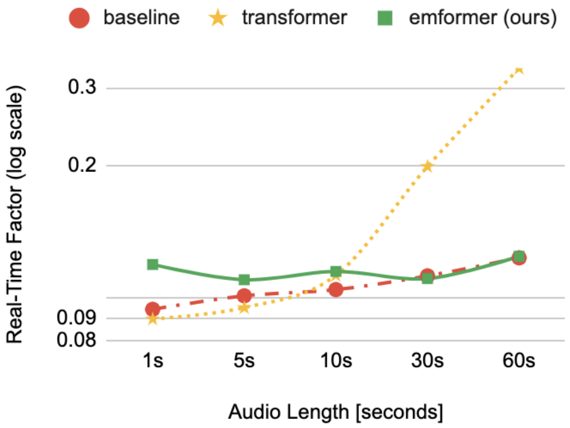

Figure 4: *Log-scale RTF comparison:* RTF is evaluated for the computation intensive spectral model with audio length ranging from 1 second to 60 seconds. The proposed Emformer based model shows comparable RTF as the multi-rate attention baseline [11], both of which deliver constant compute speed per second of audio generation regardless of audio length. A non-streaming transformer spectrum model (yellow) is also reported to indicate how the RTF grows concerning audio length.

4. Conclusion

In this work, we propose a transformer-based acoustic model for streaming speech synthesis. It uses a transformer prosody model to predict the phone-rate prosody features, and then an Emformer spectrum model to predict the frame-rate spectral features in a streaming manner. The attention mechanism in transformer and Emformer involves more long-range information, thus improving the naturalness of long-form speech. In our experiments, we use a WaveRNN neural vocoder that takes in the predicted spectral features and generates the final audio. The overall architecture achieves speech quality close to human naturalness both on short and long utterances while maintaining a low latency and constant RTF. Future work will reduce the RTF and model size of transformer-based TTS models, e.g., Conformer-based network [30] for spectrum models.

[2] Audio: https://transformer-tts-accoustic-model.github.io/samples/

5. References

[1] 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 synthesis," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[3] 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 *Advances in Neural Information Processing Systems*, 2019, pp. 3171–3180.

[4] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6706–6713.

[5] 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 *ICML*, 2018.

[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 *9th ISCA Speech Synthesis Workshop*, 2016, pp. 125–125.

[7] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[8] 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," *arXiv preprint arXiv:1710.07654*, 2017.

[9] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *arXiv preprint arXiv:1910.06711*, 2019.

[10] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[11] Q. He, Z. Xiu, T. Koehler, and J. Wu, "Multi-rate attention architecture for fast streamable text-to-speech spectrum modeling," *Proc. ICASSP*, 2021.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[13] J. Devlin, M.-W. Chang, K. Lee *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] A. Radford, K. Narasimhan, T. S. *et al.*, "Improving language understanding by generative pre-training," 2018.

[15] S. Karita, N. Chen, T. Hayashi *et al.*, "A Comparative Study on Transformer vs RNN in Speech Applications," *arXiv preprint arXiv:1909.06317*, 2019.

[16] Y. Wang, Y. Shi, F. Zhang, C. Wu, J. Chan, C.-F. Yeh, and A. Xiao, "Transformer in action: a comparative study of transformer-based acoustic models for large scale speech recognition applications," *arXiv preprint arXiv:2010.14665*, 2020.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] D. Povey, H. Hadian, P. Ghahremani *et al.*, "A time-restricted self-attention layer for ASR," in *Proc. ICASSP*, 2018, pp. 5874–5878.

[19] N. Moritz, T. Hori, and J. L. Roux, "Streaming automatic speech recognition with the transformer model," *arXiv preprint arXiv:2001.02674*, 2020.

[20] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. ICASSP*, 2020, pp. 7829–7833.

[21] L. Dong, F. Wang, and B. Xu, "Self-attention Aligner: A Latency-control End-to-end Model for ASR Using Self-attention Network and Chunk-hopping," *Proc. ICASSP*, 2019.

[22] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[23] C. Wu, Y. Shi, Y. Wang, and C.-F. Yeh, "Streaming Transformer-based Acoustic Modeling Using Self-attention with Augmented Memory," in *Proc. INTERSPEECH*, 2020.

[24] Y. Shi, Y. Wang, C. Wu, C. Fuegen, F. Zhang, D. Le, C.-F. Yeh, and M. L. Seltzer, "Weak-attention suppression for transformer based speech recognition," *arXiv preprint arXiv:2005.09137*, 2020.

[25] Y. Shi, Y. Wang, C. Wu, C.-F. Yeh, and Others, "Emformer: Efficient Memory Transformer Based Acoustic Model For Low Latency Streaming Speech Recognition," in *Proc. ICASSP*, 2021.

[26] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[27] J. Lei Ba, J. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[28] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," *arXiv preprint arXiv:1703.01541*, 2017.

[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[30] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.