Neural Networks 125 (2020) 121–130

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

On the localness modeling for the self-attention based end-to-end speech synthesis

Shan Yang [a], Heng Lu [b], Shiyin Kang [b], Liumeng Xue [a], Jinba Xiao [a], Dan Su [b], Lei Xie [a,*], Dong Yu [b]

[a] Audio, Speech and Language Processing Group (ASLP@NPU), National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[b] Tencent AI Lab, China

ARTICLE INFO

Article history:
Received 3 September 2019
Received in revised form 14 January 2020
Accepted 28 January 2020
Available online 11 February 2020

Keywords:
Speech synthesis
Self attention
Localness modeling
Relative-position-aware
Gaussian bias

ABSTRACT

Attention based end-to-end speech synthesis achieves better performance in both prosody and quality compared to the conventional "front-end"–"back-end" structure. But training such end-to-end framework is usually time-consuming because of the use of recurrent neural networks. To enable parallel calculation and long-range dependency modeling, a solely self-attention based framework named Transformer is proposed recently in the end-to-end family. However, it lacks position information in sequential modeling, so that the extra position representation is crucial to achieve good performance. Besides, the weighted sum form of self-attention is conducted over the whole input sequence when computing latent representation, which may disperse the attention to the whole input sequence other than focusing on the more important neighboring input states, resulting in generation errors. In this paper, we introduce two localness modeling methods to enhance the self-attention based representation for speech synthesis, which maintain the abilities of parallel computation and global-range dependency modeling in self-attention while improving the generation stability. We systematically analyze the solely self-attention based end-to-end speech synthesis framework, and unveil the importance of local context. Then we add the proposed relative-position-aware method to enhance local edges and experiment with different architectures to examine the effectiveness of localness modeling. In order to achieve query-specific window and discard the hyper-parameter of the relative-position-aware approach, we further conduct Gaussian-based bias to enhance localness. Experimental results indicate that the two proposed localness enhanced methods can both improve the performance of the self-attention model, especially when applied to the encoder part. And the query-specific window of Gaussian bias approach is more robust compared with the fixed relative edges.

1. Introduction

Text to speech (TTS) aims at producing natural speech from given text. With the rapid developments of neural network research in recent years, parametric speech synthesis advances from hidden Markov models (HMM) to neural networks (NN) (Black, Zen, & Tokuda, 2007; Fan, Qian, Xie, & Soong, 2014; Ling et al., 2015; Watts, Henter, Merritt, Wu, & King, 2016; Ze, Senior, & Schuster, 2013). Traditional NN-based approach typically conducts frame-level regression modeling, so we need an extra module to align texts and corresponding acoustic features in time domain (Ze et al., 2013). The alignment module is essential since changing from state- to frame-level mapping of NN approaches is one of main factors that make it outperform HMM-based systems (Watts et al., 2016). Besides, a complex and language-dependent text analysis module is also necessary to 'decompress' and represent the rich contexts of text (Wang et al., 2017), where alignment information from an alignment module is added to distinguish different frames of same state or phoneme. Since the above modules are correlative but trained independently, errors from each individual component may compound (Wang et al., 2017). Modeling text and acoustic sequences as a whole may solve this problem. Thus attention (Bahdanau, Cho, & Bengio, 2014) based sequence-to-sequence (seq2seq) (Sutskever, Vinyals, & Le, 2014) models arouse great interests recently.

The basic idea of a seq2seq model is to encode a source sequence into latent representations and then generate a variable length target sequence with an attention module. In Wang,

* Corresponding author.
E-mail addresses: syang@nwpu-aslp.org (S. Yang), bearlu@tencent.com (H. Lu), shiyinkang@tencent.com (S. Kang), lmxue@nwpu-aslp.org (L. Xue), usar@npu-aslp.org (J. Xiao), dansu@tencent.com (D. Su), lxie@nwpu.edu.cn (L. Xie), dyu@tencent.com (D. Yu).

https://doi.org/10.1016/j.neunet.2020.01.034
0893-6080/

Xu, and Xu (2016), the authors firstly proposed an end-to-end TTS framework with an attention-based seq2seq model, but an extra HMM-based aligner is needed to guide the attention matrix. Char2wav (Sotelo et al., 2017) and Tacotron (Wang et al., 2017) are more stable and advanced seq2seq models that directly predict vocoder features or raw spectrogram, respectively, without extra modules. In this way, these seq2seq systems can model the alignments between text representations and acoustic features during learning the mapping relations. Furthermore, they use characters or graphemes as input, which discards the language-dependent text analyzer and simplifies the pipelines of TTS.

Though the above seq2seq models outperform conventional TTS structures, there are still some obvious shortcomings because of the widely-used recurrent units (Li et al., 2019; Ping et al., 2017; Tachibana, Uenoyama, & Aihara, 2018) in the seq2seq models. In Char2wav and Tacotron, recurrent neural networks (RNNs) are adopted in both text encoder and auto-regressive decoder, and the gated recurrent units (GRUs) also exist in the attention layer of Tacotron (Sotelo et al., 2017; Wang et al., 2017). Since recurrent architecture relies on the entire past information when computing hidden states, it is hard to enable parallel training, which usually costs several days or even weeks to train such a model. Besides, considering the vanishing gradient problem of RNNs, in fact, it is hard to model long-range dependencies between input and output. To summarize, the major issues of the above seq2seq approaches are the low training efficiency and limited ability of global context modeling.

Considering the above issues, the solely self-attention based framework was recently proposed to achieve efficient parallel training and long-range dependency modeling with decent performance improvement over RNN-based architecture in machine translation domain (Vaswani et al., 2017). The self-attention discards any recurrence and convolution unit to make the computation to be parallelizable. For the latent feature representation, self-attention functions act as intra-attention (Lin et al., 2017; Parikh, Täckström, Das, & Uszkoreit, 2016), which has a shorter path to model long distance context. Self-attention itself does not contain any sequential information. But the position information is critical to the model performance (Vaswani et al., 2017). Besides, local contexts generally play an important role in sequential modeling (Shaw, Uszkoreit, & Vaswani, 2018; Yang et al., 2019; Yang, Tu, Wong, Meng, et al., 2018), but self-attention conducts weighed averaging on the whole sentence, which may lead to the dispersion of the attention distribution (Yang et al., 2018). In speech synthesis, the pronunciation of each word or phoneme mostly depends on the current input word or phoneme and its neighbors. As a result, such dispersion may undervalue the importance of local information, resulting in pronunciation errors like repeating and skipping. Therefore, the stability of the auto-regressive inference process for self-attention models needs to be improved.

In this paper, we introduce two methods to enhance localness modeling in self-attention based end-to-end speech synthesis, aiming at improving model stability while maintaining training efficiency and global-dependency. Firstly we analyze the solely self-attention based architecture, and show the importance of local context modeling. We then introduce the relative-position-aware (Shaw et al., 2018) approach to add extra local edges in self-attention based TTS, which enhances the local contribution in the latent representation. With local edges connections, the model can efficiently converge to better performance than the solely self-attention architecture. Besides, extra positional information is not necessary with the use of the relative-position-aware approach. It also can maintain the parallel computation and global-dependency modeling properties of self-attention at the same time.

Besides, we find the number of extra local edges (i.e. length of enhanced local window) affects the model performance according to our experiments. Too short or too long window may result in the mispronunciation or repeating problems, although it is better than the solely self-attention model. And the local edge weights are fixed after model training, which is apparently not suitable for every frame. So we further propose to inject a learnable and dynamic Gaussian bias window to enhance the localness. Instead of the fixed window size in relative-position-aware approach, the Gaussian bias method conducts query-specific window to enhance localness, where the window length varies in the whole query sequence. In this way, the predicted deviation of the Gaussian indicates the steepness of the curve or the importance of related local context in self-attention. Hence we do not need to manually adjust the sensitive hyper-parameter of relative window size. Experimental results show that the learnable Gaussian bias approach is more robust than the relative edge connections.

We summarize the main contributions of this paper as follows:

- To address the stability problem of the self-attention based speech synthesis model, this paper introduces two localness enhancement approaches: a relative-position-aware approach and a query-dependent learnable Gaussian bias approach. With these approaches, the positional encoding for injecting sequential information in conventional self-attention model is no longer required.
- The proposed localness enhancement method maintains the advantages of self-attention mechanism in terms of global context modeling and parallel training. Empirically it is about 4 times faster than Tacotron2 in model training.
- Our approach outperforms the RNN-based Tacotron2 model and the self-attention-based Transformer model with simple character inputs according to subjective tests. Compared with the conventional self-attention based Transformer, the localness enhanced self-attention is much more stable, which impressively reduces the error rate from 17% to 3%.

The rest of the paper is organized as follows. Section 2 introduces the related works. Section 3 overviews the self-attention based seq2seq framework. Section 4 introduces the basic theory of multi-head attention and self-attention for feature representation. In Section 5, we introduce the two proposed methods to enhance local information. Section 6 provides the experiments and results. We conclude this paper in Section 7.

2. Related works

2.1. Fully convolution-based speech synthesis framework

As mentioned above, training a seq2seq model with recurrent units is time-consuming. A fully CNN-based seq2seq framework was recently proposed to enable parallel training (Gehring, Auli, Grangier, Yarats, & Dauphin, 2017). It shares the basic attention-based encoder–decoder structure but substitutes all recurrent units with convolutions. When modeling contexts, RNN considers the sequential information by time dependency of state computing, while a CNN layer collects local dependencies by its kernels. The kernel size and the number of layers decide the length of receptive field of a stacked CNN structure (Gehring et al., 2017). The fully CNN-based architecture was also successfully applied in speech synthesis frameworks, such as DCTTS (Tachibana et al., 2018) and Deep Voice 3 (Ping et al., 2017).

DCTTS (Tachibana et al., 2018) contains a Text2Mel module to synthesize Mel spectrogram from input texts. Both encoder and decoder in Text2Mel are composed of causal fully convolutional

layers to achieve parallel training. But in fully CNN-based framework, the generated speech often misses or repeatedly reads some words (Ping et al., 2017). To efficiently learn the alignments between two sequences, DCTTS conducts a guided attention constraint to prompt the attention matrix to be 'nearly diagonal', which matches the nature of speech production process (Chorowski, Bahdanau, Serdyuk, Cho, & Bengio, 2015; Shen et al., 2017). In synthesis stage, it adds a hard rule to modify unsuited attention positions to make it more robust.

In Deep Voice 3 (Ping et al., 2017), the encoder and decoder are also fully CNN-based architecture. In order to inject sequential information, it adds a positional encoding to both the key and the query vectors. The positional encoding module encodes the timestep index t of sequence frames into a continuous representation in each dimension (Gehring et al., 2017). Since the generated speech also suffers from the repeating or skipping problems, it computes the softmax of attention weights only over a fixed window to make the attention monotonic in inference.

2.2. Self-attention in speech synthesis

Though fully CNN-based system shows the ability in parallel computation compared to RNNs, it also has some shortcomings such as limited receptive field in modeling long range dependencies (Gehring et al., 2017; Vaswani et al., 2017). Recently, a more parallelizable architecture named *transformer* (Vaswani et al., 2017) was proposed to get surprisingly decent results in machine translation. It is solely based on attention mechanisms, dispensing with any recurrence and convolution. There are two major advantages of the use of self-attention in *transformer*: highly parallelizable and global dependency modeling between input and output.

These benefits of self-attention have also been considered in the speech synthesis task. In Pascual, Bonafonte, and Serrà (2018), the authors utilize the parallelizable property to replace the RNN layers with self-attention in traditional frame-level TTS systems, which significantly reduces the training and inference time. And in Yasuda, Wang, Takaki, and Yamagishi (2018), self-attention is added into Tacotron2 (Shen et al., 2017) to capture long dependencies for a pitch-accent language, which achieves the best performance among their proposed seq2seq systems.

More recently, the entire *transformer* architecture is applied in the seq2seq-based speech synthesis (Li et al., 2019; Yang, Lu, Kang, Xie and Yu, 2019), which is mostly related to this paper. In Li et al. (2019), it follows the *transformer* architecture and further adds Tacotron2-like CNN-based pre-net and post-net to build the final system, leading to an efficient training process and better performance than Tacotron2. But this approach still uses phoneme-level text representation that needs language-dependent knowledge to build the synthesis model. Our previous work (Yang, Lu et al., 2019) also adopts self-attention in seq2seq-based speech synthesis and finds the importance of localness modeling especially when directly modeling from the character-level inputs. Motivated by these recent works, in this paper, we firstly analyze the learning process of solely self-attention based system and confirm the importance of localness modeling in speech synthesis. Then we propose two effective localness-enhancement modeling methods for the self-attention based speech synthesis. A series of experiments are designed to systematically analyze the effectiveness of these methods.

3. Model architecture

Fig. 1 illustrates the architecture of the proposed seq2seq framework that shares the basic encoder–decoder architecture with (Li et al., 2019; Vaswani et al., 2017; Yang, Lu et al., 2019).

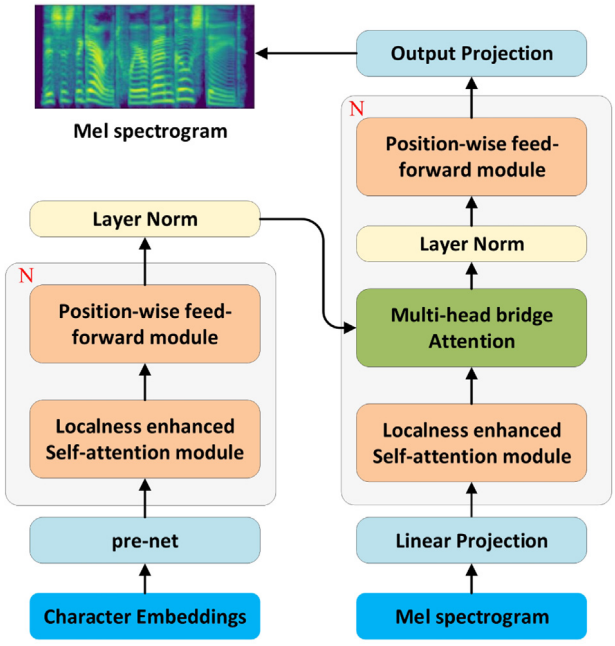

Fig. 1. System overview.

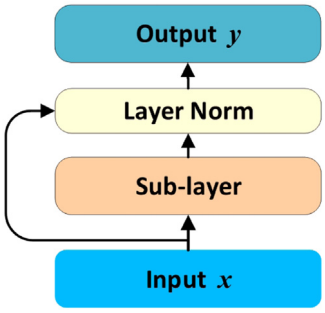

Fig. 2. An example of intra-module in both encoder and decoder.

We mainly focus on the self-attention component for latent feature learning. This architecture contains an N-block encoder, an N-block self-attention based decoder, and the multi-head bridge attention to connect the encoder and each decoder block. For each block in both encoder and decoder, it consists of a localness enhanced self-attention module and a position-wise feed-forward module. The self-attention and feed-forward modules contain a self-attention sub-layer or a projection sub-layer, respectively. The residual connection is applied in each of the two sub-layers, followed by layer normalization (Ba, Kiros, & Hinton, 2016). An example of the sub-module is shown in Fig. 2. So the output latent feature of each module is:

$$y = \text{LayerNorm}(x + \text{Sublayer}(x)) \tag{1}$$

where x is the input of each module, y is the latent output feature.

We treat the character-level text as the input of encoder, where no extra language-dependent knowledge is needed. So a potential role of the encoder is to built inner language-related dependency while learning the latent sentence representation. The output of the our architecture is the Mel-scale spectrogram of the target speech. We use WaveNet (Oord et al., 2016; Tamamori, Hayashi, Kobayashi, Takeda, & Toda, 2017) neural vocoder conditioned on the Mel spectrogram to reconstruct waveform. The details of each component will be described in Section 6.2.

4. Self-attention for latent feature learning

4.1. Multi-head attention

Attention mechanism is a key link in seq2seq model that decides the contribution of each source state in generating a specific target observation (Bahdanau et al., 2014). Assume a target sequence as query Q and a source representation as memory V, an attention function tends to output a weighted average of the memory V. Since Q and V locate in different latent space with different dimensions and lengths, the attention mechanism builds a bridge space K as the key to the memory V. Thus the weights of each memory vector can be predicted from the Q and K. Traditional attention mechanisms often conduct a single function among Q, K and V as:

$$\text{Attention}(Q, K, V) = f(Q, K)V \tag{2}$$

where $f(Q, K)$ is a score function with softmax to compute the weights of V, such as Bahdanau (Bahdanau et al., 2014) and Luong (Luong, Pham, & Manning, 2015) score functions. The weighted average result of $Attention(Q, K, V)$ is the so called *context vector* to generate next timestep output. Here we use scaled dot-product score function (Vaswani et al., 2017). Since this function has fewer computation operations, it is faster than Bahdanau (Bahdanau et al., 2014) score function and more space-efficient in practice.

$$f(Q, K) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}) \tag{3}$$

where $\sqrt{d_k}$ is the scaling factor related to the dimension of K.

Multi-head attention (Vaswani et al., 2017) shares the basic idea of the above single head attention. Differently, it projects and splits Q, K and V into h different subspace with different linear projections, where h is the number of heads. It allows the model to jointly learn from different representations of all heads, which is proven to be beneficial for many tasks (Chen et al., 2018; Wang et al., 2018).

For each head i, the context vector is computed as:

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$

The attention function of all heads can be performed in parallel to generate h outputs with Eq. (4). Finally all the h outputs are concatenated and projected to obtain the final attention values.

4.2. Self-attention based representation

As discussed above, the bridge attention, which connects the encoder and the decoder, is applied on the latent representation of the encoder input and the latent states of decoder queries. The Tacotron family (Shen et al., 2017; Wang et al., 2017, 2018) and Char2wav (Sotelo et al., 2017) both use RNNs to catch sequential information and then derive the latent representations of text inputs through the encoder. RNN is also applied in the decoder to get latent states for bridge attention. Then the learned latent features contain the context and sequential information in the recurrent way.

In *transformer*, the latent state sequence is derived from the solely attention-based architecture, which refers to self-attention (Vaswani et al., 2017) or intra-attention (Lin et al., 2017; Parikh et al., 2016). The self-attention based feature representation reduces the computational complexity of each layer and has shorter path to the long-range dependencies (Vaswani et al., 2017). As a result, the whole training process can be parallelized. Besides, it can model the global-range dependencies since it explicitly connects to the whole sequence when learning latent output features in each self-attention layer.

Self-attention is a special version of the multi-head attention, where the Q, K and V are from the same sentence $x = (x_1, x_2, \ldots, x_n)$. A goal of self-attention is to get a latent representation that considers global contexts. We look into Eq. (3) to see how self-attention works. Given a sequence x of n elements, we want to get a latent representation z with the same length n:

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V) \tag{5}$$

where α_{ij} is the weight computed from the score function described in Eq. (3):

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} \tag{6}$$

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d_k}} \tag{7}$$

Therefore, each latent z_i can assemble global dependencies on the whole sequence x. This kind of attention mechanisms is also call *global attention* since the weighted average results consider all the source states (Luong et al., 2015).

4.3. Sequential information injection

Note that there are no recurrence or convolution in self-attention. Hence it is hard to model sequential information solely with self-attention. But effective sequential modeling is essential in tasks like machine translation and speech synthesis. To overcome this problem, *transformer* injects position embeddings in the network inputs by sine and cosine functions (Vaswani et al., 2017).

The similar approach was also applied in fully CNN-based architecture (Gehring et al., 2017; Ping et al., 2017). Such position information is useful since it provides a sense of which portion of the sequence is currently dealt with (Gehring et al., 2017). But in our previous work (Yang, Lu et al., 2019), we find that such kind of position embeddings are not enough for speech synthesis in a solely attention-based model, resulting in generating mostly nonsense speech.

5. Enhanced self-attention for localness modeling

Although self-attention shows its ability in global-dependency modeling, the weighted averaging computation may lead to the dispersion of the attention distribution, and result in overlooking the relation of neighboring signals (Yang et al., 2018). In the speech synthesis task, local context dependency is critical to the performance, especially in generating correct pronunciation. Our previous work (Yang, Lu et al., 2019) show that the solely self-attention model with character inputs generates mostly nonsense speech from time to time even with position encoding. Our experiments in Section 6.3 will examine this phenomenon in detail. In this section, we will introduce two methods to further enhance localness modeling.

5.1. Relative-position-aware based self-attention

When modeling global dependencies, the self-attention ignores the distances between symbols or frames. However, local contexts usually play a critical role in speech synthesis to learn what to speak. Inspired by Shaw et al. (2018), we propose to add neighboring edge connections between elements x_i and x_j to enhance the local relations. Given current x_i, Eq. (7) shows that

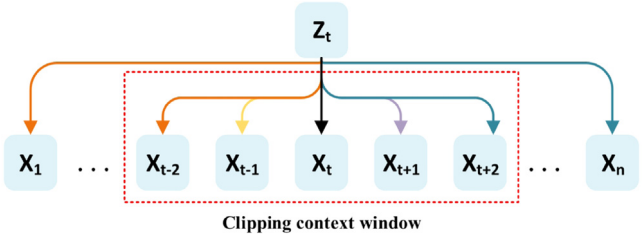

Fig. 3. Example of n relative edges representation with $m = 2$. The red dotted box shows the clipping context window. For $j \leq t-2$ or $j \geq t+2$, the representation is clipped to the same ω^K_{-2} and ω^K_2, respectively.

the part $x_j W^K$ mainly decides the contribution of x_j for generating e_i. So we modify Eq. (7) to additionally model the localness:

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K + a^K_{ij})^T}{\sqrt{d_k}} \tag{8}$$

where a^K_{ij} is the edge representation for matrix K, which strengthens the relative contribution of x_j for e_i. In Shaw et al. (2018), another edge a^V_{ij} for V is also injected in Eq. (5). But we find it does not bring improvements in our system. So we only use relative representations a^K to achieve relation enhancement.

A main goal of using sine and cosine functions in position embedding is to handle the variable sequence lengths (Vaswani et al., 2017). In order to generate different sequence lengths not seen in training by our relative-position-aware mechanism, we clip the maximum relative position to m in both directions. Then we get $2m+1$ unique relative representations to enhance neighboring relations. Each edge representation can be written as:

$$a^K_{ij} = \omega^K_{clip(j-i,m)} \tag{9}$$

$$clip(x, m) = \max(-m, min(m, x)) \tag{10}$$

The relative position representations can be simply learned by $\omega^K = (\omega^K_{-m}, \ldots, \omega^K_m)$. Fig. 3 shows an example of relative edges representation, where different colors represent different vectors.

5.2. Learnable Gaussian bias for self-attention

Although the above relative-position-aware approach can enhance local contributions of neighboring states, there are also two shortcomings. Firstly, it learns a fixed edge connection weight matrix ω^K to enhance localness. When the whole model is well-trained, all the generation process shares the same connection matrix. But different characters or frames may have different dependencies on their neighbor signals, so the fixed edge weights are not suitable for every character or frame to enhance its relative localness. Secondly, our experiments in Section 6.4.1 also show that the clipping window length affects the model performance, so empirically choosing a suitable window length for each self-attention layer is actually time consuming. To achieve dynamic local area to enhance, we further utilize Gaussian distribution as additive bias in self-attention, where window size is predicted from the each query. A similar approach is treated as *local attention* between encoder and decoder (Luong et al., 2015), where the attention only focuses on a small window of context.

Compared to the relative-position-aware approach, Gaussian distribution naturally focuses more on the closer positions. We can inject the Gaussian bias to mask the results in score function before the softmax layer in Eq. (3):

$$f(Q, K) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}} + G) \tag{11}$$

where $G \in \mathbb{R}^{N \times N}$ is the Gaussian bias matrix, and N is the length of sequence. Each bias element G_{ij} means the relation between current query x_i and position j:

$$G_{ij} = -\frac{(j - P_i)^2}{2\sigma_i^2} \tag{12}$$

where P_i is the central position of x_i, σ_i is the standard deviation.

More specifically, when generating the ith latent representation z_i, we inject a Gaussian bias vector G_i in Eq. (5). The mean and deviation of G_i, which decide the curve of the distribution, are related to the query x_i. Noted that $G_{ij} \in (-\infty, 0]$, adding such bias in the logits before softmax approximates to multiplying $(0, 1]$ after applying softmax layer. We can also multiply $exp(G)$ after softmax like (Luong et al., 2015).

The key problem of the Gaussian-based self-attention is how to choose suitable P_i and σ_i. An intuitive choice for P_i is to set $P_i = i$, because e_i is highly related to the current input x_i. In the above relative-position-aware approach, it also treats the current timestep as the central position of the clipping window. We can also predict the central position P_i from x_i:

$$P_i = N \cdot \text{sigmoid}(v_p^\top \tanh(W_p x_i)) \tag{13}$$

where *sigmod* activation is applied because its output is in $(0, 1)$, and N is the sequence length to make P_i lie in $(0, N)$. Then the final predicted position is $\lceil P_i \rceil$.

The value of σ_i determines the steepness of the curve, where a bigger σ_i means a smoother distribution. When the P_i and σ_i are both fixed, the Gaussian bias acts as a special case of relative-position-aware approach, where the weights of edge connections follow Gaussian distribution.

The σ_i is usually empirically set as $\sigma_i = \frac{D_i}{2}$ with $D_i = 10$, and D_i is the window size that indicates the local area to enhance (Luong et al., 2015; Yang et al., 2018). In order to discard the hyper-parameter m in relative-position-aware representation or D_i in the Gaussian bias, we tend to predict the window length to achieve query-related enhancements. Similar to Eq. (13), the window length can be predicted by:

$$D_i = N \cdot \text{sigmoid}(v_d^\top \tanh(W_d x_i)). \tag{14}$$

6. Experiments

6.1. Basic setup

All experiments are conducted on the public available English corpus from Blizzard Challenge 2011,[1] which contains about 13 h neutral speech of a single female English speaker. We trim long silence (> 0.1 s) at the beginning and the end of each utterance. Log magnitude spectrogram is extracted as target speech representations with 50 ms frame length and 12.5 ms frame shift with Hanning window. Character-level text representation is directly used as input. For all different systems in our experiments, we trained about 500k steps with a single Nvidia Tesla P40 GPU.

To reconstruct audios with predicted Mel spectrogram, we train a WaveNet vocoder conditioned on the ground-truth Mel spectrogram (Oord et al., 2016; Shen et al., 2017; Tamamori et al., 2017). Although a WaveNet trained with *ground truth-aligned* predicted features can improve the model performance (Shen et al., 2017), we tend to use the same WaveNet for fair comparison among all systems.

We conduct mean opinion score (MOS) and preference test to evaluate the performance of different systems. There are 20 listeners taking part in each subjective evaluation, where 30 randomly selected utterances are provided in each testing session.[2]

[1] The dataset is available at http://www.cstr.ed.ac.uk/projects/blizzard/2011/lessac_blizzard2011/.

[2] Samples can be found at https://syang1993.github.io/localness/index.html.

6.2. Model details of common components

As shown in Fig. 1 in Section 3, the basic architecture of our seq2seq model contains an N-block encoder, an N-block decoder and the bridge multi-head attention in each decoder block. For the self-attention sub-module in each block, the hidden size of the linear projections for query, key, and value is 512. The position-wise feed-forward sub-module consists of two linear transformations with 2048 and 512 hidden units, respectively. ReLU activation is conducted in the output of the first linear transformation. Besides, dropout is applied in the self-attention weights, sub-layer outputs and the ReLU activation with probability 0.1. We find this setup is critical to reduce the repeating problem in the synthesized speech. For the encoder pre-net, we use three feed-forward layers with ReLU activation, followed by dropout regularization as described in Tacotron (Wang et al., 2017).

In the decoder part, the hyper-parameters of self-attention and feed-forward modules are the same with the corresponding encoder modules. Besides, an 8-head bridge attention is added in each decoder block after the self-attention module to receive text information from the encoder. Noted that in the RNN based decoder, unidirectional RNN cells are applied to only use previous information, but self-attention layer will consider both leftward and rightward contexts, which conflicts with the auto-regressive property of the decoding process. Hence we add negative infinite bias before softmax to mask out illegal rightward connections. In this way, the decoder will only access to previous information during training and inference.

We built different systems to analyze the performance of self-attention based model and the proposed localness enhancement methods. The difference of each system will be described in the following experiments. For the WaveNet vocoder, there are totally 30 dilated layers, where each of 10 layers shares the dilatation pattern $2^0, 2^1, \ldots, 2^9$. The causal convolution and 1×1 residual convolution both have 256 channels, while the channel number of the convolutions in skip-connection is 2048.

6.3. Analysis of self-attention based feature learning

As discussed above, self-attention itself does not contain any sequential information, which is injected by extra position encoding or embedding (Vaswani et al., 2017). But our previous work found that simple absolute position encoding cannot provide enough sequential dependencies for speech synthesis (Yang, Lu et al., 2019). We firstly build such self-attention based framework without any recurrent or convolutional unit, named as system SELF-P, to analyze the details of self-attention layers. For the encoder part, it contains 3 feed-forward layers as pre-net, followed by 6 self-attention encoder block. There are also 6 self-attention blocks in auto-regressive decoder. For each self-attention layer, we use multi-head attention with 8 heads to learn the latent features.

Since for the encoder, the input of the self-attention layers does not contain any context information except for the absolute position encoding, we can judge whether the self-attention can model relations among symbols through attention weights of each layer. Fig. 4 shows the attention weights of the first self-attention layer in the encoder after more than 1M steps training. For the first self-attention layer, we can find that the attention weights of most heads (e.g. Head 3) mainly lie in the current input state for each query, which is similar to a general feed-forward layer. And the rest heads (e.g. Head 7) also consider the global contexts while focusing on current query state.

We also analyze the rest self-attention layers to see how it works. In the second self-attention layers, there are only a few specific heads that have nearly diagonal alignments. And we cannot find any attention pattern in the other heads, as well as the rest four self-attention layers. We also look into the bridge attention weights, where the distribution of attention alignments is also blurry. As a result, the generation process is quite unstable, and only a few words in the generated speech are intelligible. Besides, We find that dropout in the self-attention blocks and some hyper-parameters such as learning rate are critical to the performance according to our experiments, especially for the solely self-attention based architecture. Without careful settings, the system is hard to converge to model the sequential dependency even after more than 2M training steps, and hence the generated speech are totally nonsense (Yang, Lu et al., 2019).

Table 1
MOS evaluation of SELF-P systems with different inputs with 95% confidence interval.

Encoder inputs (m)	MOS
Character	1.78 ± 0.21
Phoneme	3.03 ± 0.16

Based on this observation, we argue that it is very hard for such solely self-attention based representation to model the relevance between discrete character symbols in the speech synthesis task. For character-level inputs, the character itself does not contain enough information for pronunciation. For example, we cannot decide how to pronounce different words "uncle" and "unit" from the individual character "u". And even for the same words, different context leads to difference pronunciations. It is the local context that mainly decides the pronunciation. But when learning the latent representation of the character inputs, the self-attention itself does not know the importance of local relations among all symbols in the sequence. In this way, the self-attention blocks need to learn a vocabulary of words or phonemes "from scratch", which makes the model harder to converge (Al-Rfou, Choe, Constant, Guo, & Jones, 2018). Unlike machine translation or speech recognition that make use of a very large dataset for training, corpus for training speech synthesis models usually contains only about 10K short text sentences. Without any auxiliary loss like (Al-Rfou et al., 2018), it is quite challenging for the encoder to model the character-level semantic dependency with such amount of limited data. A pre-trained language representation model, such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018), may be an alternative way to mitigate this problem. Besides, self-attention measures the relevance between the neighbor symbols individually, which ignores the contextual syntactic information (Yang, Li et al., 2019). And since the different latent subspaces of different heads are simply concatenated to get the final latent representation, there are no interaction for multiple attention heads to benefit from each other (Yang, Wang, Wong, Chao and Tu, 2019).

To confirm the above arguments, we simply adopt phoneme inputs to add prior language-dependent knowledge like Ping et al. (2017) and Li et al. (2019) as an auxiliary experiment to help the encoder to learn semantic relations. MOS evaluation is conducted for the SELF-P system with character-level or phoneme-level inputs, as shown in Table 1. From Table 1 we found that system SELF-P with phoneme inputs significantly outperforms the same system with character inputs. Besides, the solely self-attention based model with phoneme-level inputs can generate mostly intelligible speech with less than 100K steps. But it suffers from the word repeating problem.

We also analyze the attention weights of the first self-attention layer in the decoder. Different from the encoder, the decoder self-attention layer can learn a good attention pattern, where

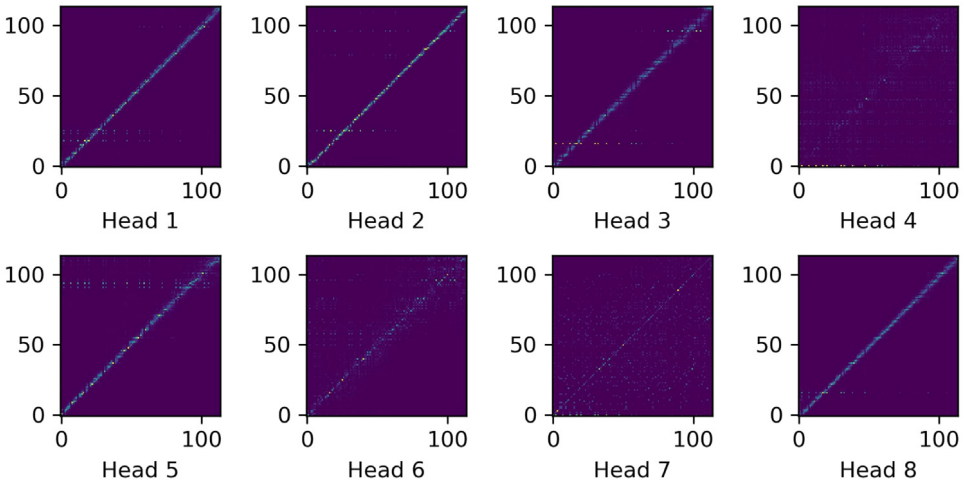

Fig. 4. The attention alignments of the first self-attention layer in the encoder.

each output frame pays most attention to the current input state. Because in the auto-regressive process, the masked self-attention only considers the leftward connections, e.g., the second output state only pays attention to the first and the second input frames. Such auto-regressive characteristic brings potential dependency in time series, so that the decoder self-attention can model the sequential information well. The self-attention alignments of the decoder can be found at our demo page.

6.4. On the effects of the proposed localness modeling

Considering the above shortcomings of the solely self-attention network, we propose to apply two localness modeling methods to build the relevance of local context dependency without any prior language information. Different from our previous hybrid architecture (Yang, Lu et al., 2019), the introduced methods maintain the advantage of parallel computation and the ability of self-attention to model global-range dependency. In this section, we use character-level inputs in all experiments to experimentally analyze the performance and the effects of the proposed localness modeling methods.

6.4.1. Experiments on relative-position-aware representation

To guide the self-attention to learn the neighboring relations, we firstly propose to use the relative-position-aware method to enhance relevance among local states. We firstly modify all self-attention layers of the SELF-P system with relative-position-aware based self-attention, named SELF-R, to see the affects of local relevances. Note that the enhanced local area is related to the maximum relative length m, which decides the enhanced context window size. So we evaluate the effect of different clipping value m, as shown in Table 2.

As discussed above, the inference process of SELF-P system is quite unstable. Table 2 clearly indicates that adding relative edges can significantly improve the performance of the generated speech. Unlike the SELF-P system, all the SELF-R systems can generate normal and intelligible speech. When the clipping value is 2, we find that there are some skip and mispronunciation problems in the generated speech, and these problems can be alleviated by increasing the context window length. The best MOS is achieved when $m = 10$. But when we further increase the value from 10, there is no significant improvement or even a little worse. Especially when m is greater than 20, there appears the repeating problem in the generated speech. Because the goal of the relative edges is to add extra connections in self-attention to enhance local context dependency, too wide window may disperse the attention of the localness modeling. Hence in the following architectures with relative-position-aware, the clipping value m is set to 10.

Table 2
MOS evaluation of SELF-R systems with different clipping value with 95% confidence interval.

Clipping value (m)	MOS
0 (SELF-P)	1.78 ± 0.21
2	2.94 ± 0.17
5	3.11 ± 0.14
10	**3.35 ± 0.12**
20	3.22 ± 0.17
40	2.72 ± 0.20

Table 3
MOS evaluation of CNN-P and CNN-R with 95% confidence interval.

Systems	MOS
CNN-P	4.01 ± 0.16
CNN-R	4.12 ± 0.11
CNN-R (encoder)	**4.15 ± 0.09**
CNN-R (decoder)	3.98 ± 0.14
Natural recording	4.36 ± 0.07

6.4.2. Auxiliary context injection in text representation

As discussed in Section 6.3, injecting contexts among input characters may improve the stability of the text encoder. In order to maintain the flexibility on parallel computation of the self-attention layer, a simple way to inject context information is to use CNNs to calculate local relations between neighboring input states, where long context can be gathered by stacked CNN layers. In this way, each input state of self-attention contains local information through a context window, whose size is decided by the kernel size of CNN. Like Tacotron2 (Shen et al., 2017), we replace the 3 feed forward pre-net with 3 CNN layers with kernel size 5 to modify the SELF-P system, where batch normalization is applied after each CNN layer. This system is named as CNN-P. Then we further add relative-position-aware in self-attention as system CNN-R. MOS evaluation results are summarized as in Table 3.

The result shows that CNN-P achieves a good MOS value 4.01,[3] which significantly outperforms SELF-P. Rather than generating

[3] Note that the MOS value is highly data-dependent. The Blizzard Challenge 2011 data is from audio book recordings ("found-data") and the quality is not as good as standard studio corpus from a professional anchor.

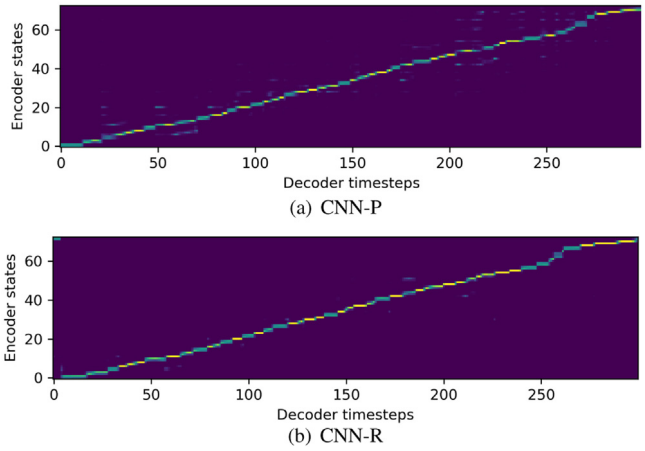

Fig. 5. An attention head alignments of the first bridge attention for system CNN-P and CNN-R.

Fig. 6. Preference test result between CNN-P and CNN-R (encoder).

some nonsense speech in SELF-P, CNN-P can consistently produce intelligible speech with good quality. The generation process of CNN-P is much more stable than SELF-P, which shows the ability of CNN in modeling the context dependency. Since the only difference for the self-attention block in the two systems is the contexts that included in the inputs, the result indicates the importance of local informations. But there are still a few mispronunciation and repeating problems in CNN-P. Replacing input sequences from English characters to phonemes sequences may alleviate such mispronunciation problem (Li et al., 2019; Ping et al., 2017). Besides, by further applying relative-position-aware edges, the CNN-R system achieves better performance than CNN-P. From the listening feedbacks, the generated speech of CNN-R has better intonation and much fewer mispronunciation cases. We also analyze the attention weights of the bridge attention in CNN-P and CNN-R. We plot an attention head alignments of the first bridge attention in Fig. 5, which shows that CNN-R has much clearer attention path. Note that only some specific heads in the former bridge attention layers have nearly diagonal alignments.

We further analyze the effects of the relative-position-aware connection in the encoder and decoder individually. The subjective evaluation shows that only applying local connections in the encoder outperforms other systems. To confirm the benefits of the localness modeling, we also conduct A/B preference test between the system CNN-P and CNN-R (encoder), as shown in Fig. 6. The results indicate that adding local enhancement in self-attention significantly improve the model performance. And with relative-position-aware localness modeling, we do not need the position encoding anymore.

But when just applying the enhanced edges in the decoder self-attention, we find there may exist strange tones in the generated speech. Considering that we use fixed window length to enhance localness, the connection weights are shared for all utterances after model training, which may be not suitable for the decoding process of each individual utterance. A query-relative window may make the localness enhancement more adaptive to difference sequences, which we will analyze later.

6.5. Gaussian bias for localness modeling

In the relative-position-aware approach, the enhanced localness weights remain unchanged during inference. Since different characters or frames have different range of local dependencies, the fixed edge weights are not suitable for all time steps. We further propose a learnable Gaussian bias method to enhance the localness. Different from the fixed relative edges, we predict the local window from each query state to achieve dynamic and query-specific enhancements. We treat this system as CNN-G, where Gaussian bias is applied in each self-attention layer in CNN-P. Although the Gaussian window is defined by the central position and window length, we only predict the window length to enhance localness. That is because for current time step i, the latent output z_i is highly related to the input x_i. Compared with the relative-position-aware approach, the extra local weights become dynamic and are more relevant to different queries. For a sanity check, we also experiment a fix window size for Gaussian bias, where window length D is set to 20. MOS evaluation results are shown in Table 4.

Table 4
MOS evaluation of CNN-G systems with 95% confidence interval.

Systems	MOS
CNN-R (encoder)	4.15 ± 0.09
Fixed window ($D = 20$)	3.93 ± 0.18
Predicted window	$\mathbf{4.16 \pm 0.07}$

Table 5
Counted different types of errors for different models on the 100-sentence test set.

	CNN-P	CNN-R	CNN-G
Skipping error	12	6	**2**
Repeating error	5	2	**1**
Mispronunciation	6	3	**0**
Error sentences	17	8	**3**

With fixed Gaussian window, we find the result is worse than CNN-P and CNN-R. The generated speech of the fixed Gaussian window suffers a lot from the skipping problems though the quality and prosody are as good as CNN-R. When we change the variance of the Gaussian bias, we may get more stable generated speech but with worse prosody. When we adopt the query-specific window to enhance localness, we achieve the best performance among all the above systems.

Although the MOS value of the CNN-G system with learnable Gaussian is very similar to the CNN-R system, we find the CNN-G system is much more stable during inference. In order to evaluate the robustness of the proposed methods, we use different systems to generate the same 100 test sentences and count the generation errors. The results are summarized in Table 5, where CNN-R uses relative position only in encoder and CNN-G uses the predicted window. From the results, we find that the basic CNN-P system, which is similar to TransformerTTS (Li et al., 2019), suffers a lot from generation errors, especially skipping errors. This attributes to the fact that the original unrestricted version of self-attention may ignore local parts of texts, although it can model global contexts. When we adopt the relative-position-aware method to enhance localness, all types of errors are significantly reduced. After applying the proposed query-specific dynamic window to further enhance localness, we obtain the minimal generation errors in the CNN-G system. This result confirms the stability of the learnable Gaussian bias in enhancing localness and speech generation.

6.6. Comparison with RNN-based models

Although the main purpose of this work is to show how the localness modeling affect the self-attention based speech synthesis model, we also compare the proposed model with RNN-based

Table 6
MOS evaluation of RNN-based Tacotron2 with 95% confidence interval.

Systems	MOS
Tacotron2 (character)	4.09 ± 0.11
Tacotron2 (phoneme)	4.15 ± 0.08
CNN-G (predicted window)	**4.16 ± 0.07**
Natural recording	4.36 ± 0.07

Table 7
Training speed of different models on a single Nvidia Tesla P40 GPU, where all models are trained with about 500k steps for fair comparison.

System	Time per step (s)	Total training time (day)
Tacotron2	1.25	7.2
CNN-G	0.37	2.2

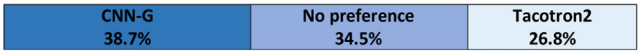

Fig. 7. Preference test result between Tacotron2 and CNN-G in character-level.

seq2seq models as a sanity check. We follow the Tacotron2 (Shen et al., 2017) architecture to build the RNN-based model using the same data as our proposed model. For Tacotron2, we experiment both character- and phoneme-level inputs, as shown in Table 6.

For the simple character-level inputs, the proposed model achieves a MOS value of 4.16, which outperforms Tacotron2 (4.09). We believe the performance gain comes from the proposed localness modeling scheme. Besides, the proposed model with character-level representation even achieves comparable performance with the phoneme-level Tacotron2 (MOS 4.15). This means our proposed approach can simplify the front-end module in TTS. We also conduct A/B preference test between Tacotron2 and the proposed self-attention with query-specific dynamic window (CNN-G), as shown in Fig. 7. The result indicates that listeners give more preferences on the proposed approach, showing that the localness enhanced self-attention version outperforms the RNN-based model by a large margin.

The model sizes for Tacotron2 and CNN-G are about 110M and 170M, respectively. Since we use 6 self-attention blocks for both encoder and decoder in the CNN-G system, its model size is bigger than the Tacotron2 model. We further analyze the training time of the two systems. For fair comparison, we trained both models with the same GPU and the same batch size. From Table 7, we can find that the training speed of CNN-G is much faster than Tacotron2 since the training process of CNN-G is parallel. And it is worth to mention that with output cache,[4] the speech generation time of CNN-G is mostly the same with Tacoton2.

7. Conclusion

In this paper, we propose two localness enhancement methods to improve the performance of self-attention based model, which maintains the advantages of parallel computation and global-range dependency modeling of self-attention while improving the generation stability. We systematically analyze the performance of the solely self-attention model for speech synthesis, and find the importance of local context especially using character-level input towards a fully end-to-end system. To enhance localness modeling, we firstly propose the relative-position-aware approach to model the local context. Different from generating mostly nonsense speech in the solely self-attention based model, the self-attention with relative edges can generate intelligible speech effectively. Experimental results indicate that the window size of the localness affects the model performance. Although injecting prior language-dependent or context information in the text representation is helpful in modeling localness, the proposed enhanced local connections in self-attention can also further improve the model performance.

Since the local connection weights stay unchanged during inference and the window size directly affects the model performance for different sentences, we further propose a learnable Gaussian bias to continue to enhance the localness in self-attention. Different from the fixed window in relative edges, with the Gaussian bias approach, the window size is dynamic and highly related to the current query. Hence we can achieve a query-specific enhancement to model different local weights for different state. The experimental result shows that the Gaussian based model outperforms all other systems, and it is much more stable than the relative-position-aware based localness modeling.

In future work, we will focus on improving the performance of the self-attention based speech synthesis framework. Since there are no interaction among all heads in the multi-head attention to get output representation, we also tend to find a proper way to make the attention head communicate with each other.

[4] https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/layers/transformer_layers.py.

Acknowledgments

The research work is supported by the National Key Research and Development Program of China (No. 2017YFB1002102) and Tencent AI Lab Rhino-Bird Joint Research Program, China (No. JR201853).

References

Al-Rfou, R., Choe, D., Constant, N., Guo, M., & Jones, L. (2018). Character-level language modeling with deeper self-attention. arXiv preprint arXiv:1808.04444.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Black, A. W., Zen, H., & Tokuda, K. (2007). Statistical parametric speech synthesis. In *Proc. ICASSP* (pp. 1229–1232).

Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., et al. (2018). The best of both worlds: Combining recent advances in neural machine translation. arXiv preprint arXiv:1804.09849.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In *Proc. NPIS* (pp. 577–585).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Fan, Y., Qian, Y., Xie, F.-L., & Soong, F. K. (2014). TTS synthesis with bidirectional LSTM based recurrent neural networks. In *Proc. INTERSPEECH*.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. arXiv preprint arXiv:1705.03122.

Li, N., Liu, S., Liu, Y., Zhao, S., Liu, M., & Zhou, M. (2019). Close to human quality TTS with transformer. In *Proc. AAAI*.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., et al. (2017). A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130.

Ling, Z.-H., Kang, S.-Y., Zen, H., Senior, A., Schuster, M., Qian, X.-J., et al. (2015). Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine*, *32*(3), 35–52.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., et al. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

Parikh, A., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In *Proc. EMNLP*.

Pascual, S., Bonafonte, A., & Serrà, J. (2018). Self-attention linguistic-acoustic decoder. arXiv preprint arXiv:1808.10678.

Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., et al. (2017). Deep voice 3: 2000-speaker neural text-to-speech. arXiv preprint arXiv:1710.07654.

Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. arXiv preprint arXiv:1803.02155.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., et al. (2017). Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. arXiv preprint arXiv:1712.05884.

Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., et al. (2017). Char2wav: End-to-end speech synthesis. In *Proc. ICLR Workshop*.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proc. NIPS* (pp. 3104–3112).

Tachibana, H., Uenoyama, K., & Aihara, S. (2018). Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *Proc. ICASSP* (pp. 4784–4788).

Tamamori, A., Hayashi, T., Kobayashi, K., Takeda, K., & Toda, T. (2017). Speaker-dependent WaveNet vocoder. In *Proc. INTERSPEECH* (pp. 1118–1122).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Proc. NIPS* (pp. 5998–6008).

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., et al. (2017). Tacotron: Towards end-to-end speech synthesis. In *Proc. INTERSPEECH* (pp. 4006–4010).

Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R., Battenberg, E., Shor, J., et al. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. arXiv preprint arXiv:1803.09017.

Wang, W., Xu, S., & Xu, B. (2016). First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention. In *Proc. INTERSPEECH* (pp. 2243–2247).

Watts, O., Henter, G. E., Merritt, T., Wu, Z., & King, S. (2016). From HMMs to DNNs: where do the improvements come from?. In *Proc. ICASSP* (pp. 5505–5509).

Yang, B., Li, J., Wong, D., Chao, L. S., Wang, X., & Tu, Z. (2019). Context-aware self-attention networks. In *Proc. AAAI*.

Yang, S., Lu, H., Kang, S., Xie, L., & Yu, D. (2019). Enhancing hybrid self-attention structure with relative-position-aware bias for speech synthesis. In *Proc. ICASSP* (pp. 6910–6914).

Yang, B., Tu, Z., Wong, D. F., Meng, F., et al. (2018). Modeling localness for self-attention networks. In *Proc. EMNLP*.

Yang, B., Wang, L., Wong, D. F., Chao, L. S., & Tu, Z. (2019). Convolutional self-attention network. In *Proc. NAACL*.

Yasuda, Y., Wang, X., Takaki, S., & Yamagishi, J. (2018). Investigation of enhanced tacotron text-to-speech synthesis systems with self-attention for pitch accent language. arXiv preprint arXiv:1810.11960.

Ze, H., Senior, A., & Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. In *Proc. ICASSP* (pp. 7962–7966).