Neural Networks 141 (2021) 306–314

Contents lists available at ScienceDirect

Neural Networks

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

2021 Special Issue

FastTalker: A neural text-to-speech architecture with shallow and group autoregression

Rui Liu [a,b,*], Berrak Sisman [a], Yixing Lin [b,c], Haizhou Li [a,d]

[a] Singapore University of Technology and Design (SUTD), Singapore
[b] National University of Singapore, Singapore
[c] National Central University, Taoyuan City, Taiwan, China
[d] University of Bremen, Germany

ARTICLE INFO

Article history:
Available online 21 April 2021

Keywords:
Neural text-to-speech synthesis
Shallow autoregressive
Group autoregressive
Temporal dependency

ABSTRACT

Non-autoregressive architecture for neural text-to-speech (TTS) allows for parallel implementation, thus reduces inference time over its autoregressive counterpart. However, such system architecture does not explicitly model temporal dependency of acoustic signal as it generates individual acoustic frames independently. The lack of temporal modeling often adversely impacts speech continuity, thus voice quality. In this paper, we propose a novel neural TTS model that is denoted as *FastTalker*. We study two strategies for high-quality speech synthesis at low computational cost. First, we add a shallow autoregressive acoustic decoder on top of the non-autoregressive context decoder to retrieve the temporal information of the acoustic signal. Second, we further implement group autoregression to accelerate the inference of the autoregressive acoustic decoder. The group-based autoregression acoustic decoder generates acoustic features as a sequence of groups instead of frames, each group having multiple consecutive frames. Within a group, the acoustic features are generated in parallel. With the shallow and group autoregression, *FastTalker* retrieves the temporal information of the acoustic signal, while keeping the fast-decoding property. The proposed *FastTalker* achieves a good balance between speech quality and inference speed. Experiments show that, in terms of voice quality and naturalness, *FastTalker* outperforms the non-autoregressive FastSpeech baseline significantly, and is on par with the autoregressive baselines. It also shows a considerable inference speedup over Tacotron2 and Transformer TTS.

1. Introduction

Text-to-speech (TTS) aims to generate intelligible and natural voice which is indistinguishable from human vocal production (Taylor, 2009; Zen, Tokuda, & Black, 2009). In the past decades, TTS has gained popularity as part of artificial intelligence and has been widely used in many applications (Sisman, King, Yamagishi, & Li, 2021) such as smart voice assistants, dubbing of movies and games, online education, and smart home.

In the recent past, concatenative (Hunt & Black, 1996; Merritt, Clark, Wu, Yamagishi, & King, 2016) and statistical parametric speech synthesis (Liu, Bao, Gao, & Wang, 2017; Tokuda et al., 2013; Wu & King, 2016; Zen & Sak, 2015; Zen, Senior, & Schuster, 2013) systems were the mainstream techniques. We note that both of these techniques have complex pipelines including front-end model, duration model and acoustic model. Moreover, designing a good linguistic feature is often laborious and language-specific, which requires substantial engineering efforts.

With the advent of deep learning, end-to-end generative TTS models simplify traditional speech synthesis pipeline with a single neural network. They include Tacotron-based neural TTS (Shen et al., 2018; Wang et al., 2017) and its variants (Habib et al., 2020; Hsu et al., 2019; Skerry-Ryan et al., 2018; Sun et al., 2020, 2020). In these techniques, the key idea is to integrate the conventional TTS pipeline into a unified encoder–decoder network (Bahdanau, Cho, & Bengio, 2015a) and to learn the mapping directly from the <text, wav> pair (Oord et al., 2016).

Tacotron is a successful encoder–decoder implementation based on recurrent neural networks (RNN), such as LSTM (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2016; Hochreiter & Schmidhuber, 1997) and GRU (Cho et al., 2014). Generally, a Tacotron system (Shen et al., 2018; Wang, Skerry-Ryan et al., 2017) takes the character embeddings of text sequence as input and predicts the acoustic features using an encoder–decoder

* Corresponding author at: National University of Singapore, Singapore.
E-mail addresses: liu_rui@sutd.edu.sg (R. Liu), berrak_sisman@sutd.edu.sg (B. Sisman), e0502450@u.nus.edu (Y. Lin), haizhou.li@nus.edu.sg (H. Li).

https://doi.org/10.1016/j.neunet.2021.04.016
0893-6080/

framework with attention mechanism. Together with a WaveNet-like neural vocoder (Hayashi, Tamamori, Kobayashi, Takeda, & Toda, 2017; Kobayashi, Hayashi, Tamamori, & Toda, 2017; Sisman, Zhang, & Li, 2018, 2019; Tamamori, Hayashi, Kobayashi, Takeda, & Toda, 2017), it produces impressive natural-sounding speech. However, the autoregressive decoding (Jordan, 1997; Williams & Zipser, 1989) in recurrent neural network limits the possibility of parallel computing in both training and inference.

Self-attention architecture represents another type of encoder–decoder implementation, that achieves efficient parallel training with decent performance in machine translation (Vaswani et al., 2017). It allows for parallel computation during teacher-forcing training as implemented in Transformer TTS (Li, Liu, Liu, Zhao, & Liu, 2019). Self-attention mechanism also brings us another benefit. It functions as intra-attention (Liu, Sisman and Li, 2020; Vaswani et al., 2017; Yang et al., 2020), which has a shorter path to model long distance context. Despite the progress (Karita et al., 2019; Li et al., 2019; Shen et al., 2018; Wang, Skerry-Ryan et al., 2017), self-attention architecture suffers from slow decoding due to its autoregressive nature (Jordan, 1997; Williams & Zipser, 1989). As the generation of an output frame depends on its decoding history, parallel computation is not possible during inference.

To reduce inference latency resulting from autoregressive decoding, recently the community has studied non-autoregressive solutions. Unlike autoregressive models, non-autoregressive models (Gu, Bradbury, Xiong, Li, & Socher, 2018) generate output sequence in parallel, without explicitly depending on the previous decoding history. Along this line of thought, several non-autoregressive TTS models are proposed that include FastSpeech (Ren et al., 2019), FastSpeech2 (Ren et al., 2020), AlignTTS (Li et al., 2020) and Moboaligner (Zeng, Wang, Cheng, Xia, & Xiao, 2020). While these models reduce the computational cost during run-time inference, they do not model temporal dependency explicitly, hence speech continuity often suffers.

The prior works all suggest that speech synthesis benefits from temporal modeling of acoustic signal (Wang, Takaki and Yamagishi, 2018; Watts, Henter, Fong, & Valentini-Botinhao, 2019). During training, a model learns the temporal dependency from the target-side natural sequence; and during inference, the system applies the learned model to ensure natural articulation and temporal dependency of speech. The question is how to find a solution that models temporal dependency as effective as autoregressive models, and as efficient as non-autoregressive models in terms of computation, which will be the focus of this paper.

The trade-off between computational efficiency and temporal modeling is a research problem in many sequence-to-sequence decoding tasks. In neural machine translation, shallow autoregressive mechanism is introduced to a non-autoregressive network (Guo et al., 2019; Shao et al., 2019) to reinforce temporal dependency. However, the use of shallow autoregression in neural TTS remains to be explored.

In this paper, we propose a novel neural TTS model, denoted as *FastTalker*, which generates high-quality speech at low computational cost. *FastTalker* takes two strategies. First, we separate the decoder into a non-autoregressive context decoder and an autoregressive acoustic decoder to minimize the autoregressive computation; Second, the decoder generates acoustic features as a sequence of groups instead of frames, each group having multiple consecutive frames. Within a group, the acoustic features are generated in parallel. We believe that the combination of shallow and group autoregression serves as a trade-off between computational efficiency and temporal modeling in neural TTS. The main contributions of this paper include:

- We propose a novel neural TTS architecture, with shallow autoregression mechanism, which incorporates the autoregressive module into the non-autoregressive decoding process. We benefit from shallow autoregression for effective temporal modeling.
- We employ a group-based autoregression mechanism to further accelerate the shallow autoregression. The decoder of FastTalker benefits from non-autoregression and group-based autoregression strategy for computational efficiency.
- We show that the neural TTS with shallow and group autoregression outperforms the non-autoregressive counterpart significantly, and is on par with other autoregressive systems, such as Tacotron2 and Transformer TTS, in terms of voice quality.

This paper is organized as follows. We motivate the study and present the prior work in Section 2. We further formulate the proposed *FastTalker* in Section 3. We report the results of a systematic evaluation and comparison in Section 4. Finally, Section 5 concludes the study.

2. Background and motivation

Human voice is a continuous signal regulated by physiological constraints of human vocal production mechanism, which involves complex fluid–structure interaction within the glottis and its control by laryngeal muscle activation (Zhang, 2016). Simply speaking, the lungs produce airflow that vibrates vocal folds, which create audible pulses that form the laryngeal sound source. The voiced sound is amplified and modified by the vocal tract resonators, e.g. the throat, mouth cavity, and nasal passages, that presents one's voice identity. The vocal tract articulators, e.g. the tongue, soft palate, and lips modify the voice sound to produce recognizable words (Titze & Martin, 1998). Human voice is natural and stable because it follows the voice physiology and biomechanics, the physics of vocal fold vibration and sound production, and laryngeal muscular control (Zhang, 2016).

Both traditional and neural TTS computational models endeavor to generate speech by observing the physiological constraints of voice production. This can be achieved through temporal modeling. Next we first study temporal modeling of speech as far as TTS is concerned. We will also review typical autoregressive neural TTS, and non-autoregressive neural TTS model to motivate our research problem and to set the stage for our study.

2.1. Temporal modeling of speech

Temporal modeling of speech has been studied in traditional TTS frameworks. For example, unit selection based TTS techniques impose a unit concatenation cost (Hunt & Black, 1996; Merritt et al., 2016) that is an estimate of concatenation quality of pairs of units. The idea is to select the most suitable sequence of acoustic units from the available acoustic inventory (Black & Campbell, 1995; Ling & Zhou, 2018). The optimization process reflects an interaction among multiple objectives, that include target cost that measures the mismatch between the expected target and the candidate unit, i.e. phonetic adequacy, and concatenation cost that estimates the temporal dependency of the acoustic signal when concatenating selected units, i.e. temporal dependency.

HMM-based approach represents a successful parametric solution (Yoshimura, 1999) to TTS. As HMM-based approach aims to maximize the output probability which makes the parameter sequence a mean vector sequence, resulting in a step-wise function (Duda, Hart, & Stork, 2012). To produce continuous acoustic signal, a maximum likelihood parameter generation algorithm (Tokuda, Yoshimura, Masuko, Kobayashi, & Kitamura,

2000) is used to generate the trajectory of acoustic features by taking the dynamic features into account. Autoregressive HMM-based TTS (Shannon & Byrne, 2009; Zen, Tokuda, & Kitamura, 2007) represents another solution to the same problem. It explicitly models the dynamics of speech to account for the continuity and temporal dependency needed for good quality synthesis.

Neural TTS systems ensure the temporal dependency of acoustic signal in different ways. For example, dynamic features (Ling et al., 2015; Zen et al., 2013) are used in addition to static features. A novel loss function (Matsunaga, Ohtani, & Hirahara, 2019) is studied with long and short term acoustic features.

Others use recurrent neural network (Wang, Takaki et al., 2018; Wang, Takaki, Yamagishi, King, & Tokuda, 2019) as an autoregressive function to incorporate temporal dependency into the training process. All studies, such as Tacotron (Wang, Skerry-Ryan et al., 2017), Tacotron2 (Shen et al., 2018) and Transformer TTS (Li et al., 2019), have highlighted the importance of temporal modeling (Watts et al., 2019) in neural TTS.

2.2. Autoregressive neural TTS

The encoder–decoder neural network architecture has been widely used in machine translation (NMT) (Luong, Pham, & Manning, 2015; Tu, Liu, Shang, Liu, & Li, 2017; Zheng et al., 2018). Successful examples include recurrent neural networks (Bahdanau, Cho, & Bengio, 2015b; Siddhant et al., 2020; Sutskever, Vinyals, & Le, 2014), convolutional neural networks (Li, Wang, Xiao, Liu and Zhu, 2020), and self-attention (Vaswani et al., 2017). They perform translation in an autoregressive manner by generating an output sentence word by word from left to right. Autoregressive neural TTS follows a similar idea (Li, Zhang, Liu, Zhang, & Bao, 2018; Liu, Sisman, Bao, Gao, & Li, 2020a; Liu et al., 2021).

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ represent the input text sequence, and $y_{<t}$ represent the generated acoustic frames before y_t, $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ represent the target sequence of acoustic features. The decoding of the tth speech frame y_t follows a conditional distribution $p(y_t|y_{<t}, \mathbf{x}, \theta)$, where speech frame y_t depends on the entire input text sequence $\mathbf{x}$, partial decoding history $y_{<t}$, and the network parameters θ. The joint probability of an autoregressive decoder can be formulated as follows:

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_{t=1}^{n} p(y_t|y_{<t}, \mathbf{x}, \theta) \tag{1}$$

The condition on $\mathbf{x}$ ensures phonetic adequacy, and that on partial decoding history $y_{<t}$ enforces temporal dependency (Wang, Skerry-Ryan et al., 2017). As the decoder has a direct impact on the temporal dependency of output acoustic signal, autoregression is often implemented in the decoder (Li et al., 2019; Liu, Sisman, Bao, Gao, & Li, 2020b; Liu et al., 2020; Shen et al., 2018). While such autoregressive decoding ensures the temporal dependency of acoustic signal, it does not permit parallel processing and involves intensive recurrent computation (Liu, Sisman, Gao and Li, 2020).

2.3. Non-autoregressive neural TTS

Non-autoregressive neural TTS systems, such as FastSpeech (Ren et al., 2019) and FastSpeech2 (Ren et al., 2020), feature a novel Feed-Forward Transformer (FFT) architecture that allows for parallel processing with a self-attention mechanism. A FFT encoder encodes a sequence of words in the text input in parallel. A FFT decoder generates output acoustic features from context vectors also in parallel. Together with parallel neural vocoder (Prenger, Valle, & Catanzaro, 2019; Wu, Hayashi, Okamoto, Kawai, & Toda, 2020a, 2020b; Yang et al., 2020), a full parallel implementation is possible, that is attractive for low-latency deployment.

Without depending on the decoding history, non-autoregressive approach generates output acoustic features as a product of the conditionally independent probability of individual speech frames:

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_{t=1}^{n} p(y_t|\mathbf{x}, \theta). \tag{2}$$

The conditional independence allows the model to generate multiple target acoustic frames concurrently, thus leading to significant reduction of computation over the autoregressive counterpart.

Non-autoregressive model solely relies on its attention mechanism to decode the acoustic features from the encoded textual features, without any autoregressive input. Positional encoding (Li et al., 2019) becomes crucial in guiding the attention to follow a monotonic progression over time at the beginning of training. To ensure the monotonicity and continuity, some non-autoregressive models use various monotonic alignment tricks (Kim, Kim, Kong, & Yoon, 2020; Li, Liu et al., 2020; Zeng et al., 2020). However, as they do not explicitly model the temporal dependency, speech continuity remains an issue to be addressed, that will be our focus in Section 3.

3. FastTalker

The way autoregression works is to take the output of one neuron and return it as input to another neuron in the network. During decoding, an output depends on its previous decoding history. The computation complexity of such recurrent mechanism depends on its parameters, e.g. the depth, the size, of the deep neural network, denoted as θ.

To improve the temporal modeling, one idea is to apply autoregression only at a subnet near the output, denoted as *shallow autoregression* (Wang, Takaki and Yamagishi, 2017; Wang, Takaki et al., 2018). It means that the output feedback goes through a shallow subnet instead of entire deep neural network. To reduce the autoregressive computation, some studies attempt to generate the output acoustic features in groups instead of frames, each group having multiple consecutive frames. In other words, we have non-autoregressive decoding locally within a group, and autoregressive decoding globally across the groups. We embrace both ideas in this paper. In short, we would like to reinforce the temporal dependency of output speech, at the same time, we employ a group-based autoregressive mechanism for further rapid computation, and find a good balance between the speech quality and the inference speed.

The overall *FastTalker* framework is illustrated in Fig. 1, that has two main modules: (1) *Encoder* and (2) *FastDecoder*. Specifically, the *Encoder* takes the text sequence as input and models the global context using the multi-head self-attention mechanism. The *FastDecoder* contains a shallow autoregression decoder, which includes non-autoregressive context decoder and a group-based autoregressive acoustic decoder.

FastTalker features a novel neural TTS architecture that involves autoregression in the acoustic decoder near the output. With the shallow and group autoregression, it learns the temporal dependency of speech effectively at low computational cost, which is not considered previously by full non-autoregressive neural TTS, such as FastSpeech (Ren et al., 2019).

3.1. Encoder

The encoder consists of an encoder layer that converts input character sequence into context vector sequence, and a length regulator that resolves the alignment between input character sequence and output acoustic features of different length.

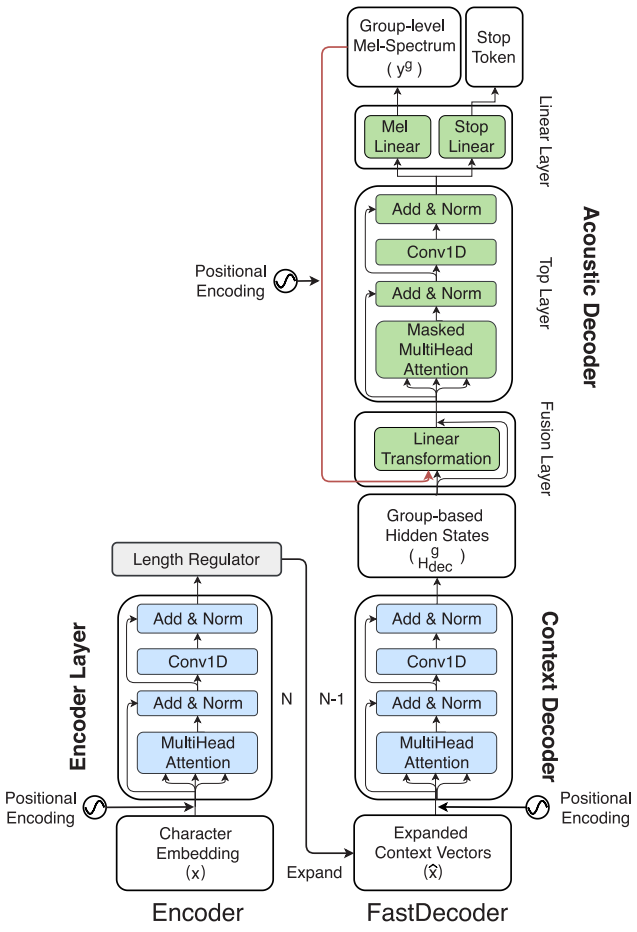

Fig. 1. The proposed *FastTalker* consists of Encoder (left panel) and FastDecoder (right panel). The FastDecoder contains a non-autoregressive context decoder (in blue) followed by a group autoregressive acoustic decoder layer (in green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.1.1. Encoder layer

The encoder layer contains N Feed-Forward Transformer (FFT) blocks on the character input side. Each FFT block consists of a self-attention and 1D convolutional network (Li et al., 2019). The self-attention network consists of a multi-head attention to extract the cross-position information. The 1D convolutional network consists of 2-layer 1D convolutional with ReLU activation. Furthermore, residual connections, layer normalization, and dropout are added after the self-attention network and 1D convolutional network respectively (Ren et al., 2019).

In practice, the input character sequence is first represented by a sequence of m character embeddings $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$. The encoder layer takes $\mathbf{x}$ as input and generates a hidden state sequence, also referred to as context vectors, $H_{enc} = \{h_1, h_2, \ldots, h_m\}$ as output with a self-attention mechanism.

$$H_{enc} = \text{Encoder}(\mathbf{x}) \tag{3}$$

As the entire input sequence is presented to the encoder at once, the encoding is a parallel process.

3.1.2. Length regulator

The length regulator is used to resolve the length mismatch between the input character sequence and output acoustic features (Ren et al., 2019), that refers to Mel-spectrum sequence in this paper.

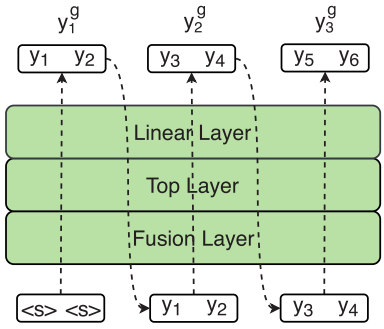

Fig. 2. An illustration of the group-based autoregression strategy for acoustic decoder with group size $K = 2$.

Specifically, the length regulator takes the hidden states sequence $H_{enc} = \{h_1, h_2, \ldots, h_m\}$ as input and outputs the estimated duration sequence $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$. According to the character duration d_i, where $i \in [1, m]$, the length regulator expands each hidden state h_i of the character sequence d_i times. In this way, we obtain an expanded sequence of context vectors, $\hat{\mathbf{x}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$, that is of n frames having the same length as the output sequence of acoustic features, typically, $n \gg m$. The length regulator is described as follows:

$$\hat{\mathbf{x}} = \text{Regulator}(H_{enc}, \mathcal{D}) \tag{4}$$

The encoder of *FastTalker* model shares a similar architecture with that of FastSpeech (Ren et al., 2019), which is proven effective. In this paper, we would like to focus on the study of a novel decoder mechanism, the *FastDecoder*.

3.2. Shallow and group autoregression: FastDecoder

FastDecoder takes the expanded sequence of context vectors $\hat{\mathbf{x}}$ as input to relate the context vectors to acoustic features. We propose a two-step shallow autoregressive decoding process that consists of (1) context decoding; and (2) acoustic decoding, as illustrated in Fig. 1. The context decoder operates in a non-autoregressive manner, while the acoustic decoder involves group autoregression, as illustrated in Fig. 2. The acoustic decoder consists of a fusion layer, a top layer, and a linear layer.

The *FastDecoder* benefits from the shallow autoregressive mechanism for speech quality; and group autoregression and largely non-autoregressive decoding mechanism for computational efficiency, which we will discuss in detail next.

3.2.1. Context decoder

The context decoder of proposed FastTalker employs an FFT architecture that takes the sequence of context vectors $\hat{\mathbf{x}}$ as input. We note that $\hat{\mathbf{x}}$ and H_{dec} sequences are of the same length of n frames. The context decoding can be done in parallel as shown in Fig. 1, which allows for rapid computation. It is expected to relate the input context vectors to the acoustic features through an intermediate representation $H_{dec} = \{h_1, h_2, \ldots, h_n\}$.

$$H_{dec} = \text{ContextDecoder}(\hat{\mathbf{x}}) \tag{5}$$

The context decoder in FastTalker shares a similar architecture with the decoder of FastSpeech (Ren et al., 2019) as they both read and process data in a non-autoregressive manner. The non-autoregressive implementation improves the decoding speed and achieves acceptable performance as it does not model the temporal context (Ren et al., 2019).

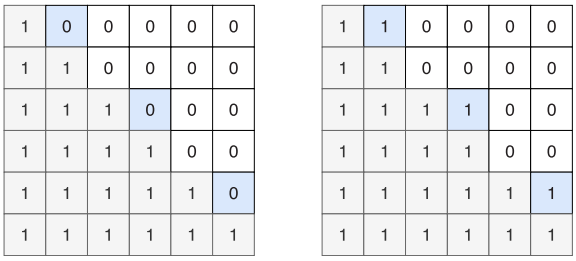

Fig. 3. Frame-based causal mask (left) and group-based causal mask (right) (take the target length n = 6 and the group size K = 2 for example). We mark their differences in blue color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2.2. Fusion layer in acoustic decoder

The fusion layer is a linear transformation layer with a ReLU activation, which fuses the group-level hidden states and group-level target acoustic features in each time step in a group autoregressive manner.

First of all, we divide the hidden states sequence in Eq. (5), $H_{dec} = \{h_1, h_2, \ldots, h_n\}$, into group-based hidden states sequence $H^g_{dec} = \{h^g_1, h^g_2, \ldots, h^g_l\}$ with a sequence grouping trick:

$$\begin{aligned} h^g_1 &= \{h_1, \ldots, h_K\} \\ h^g_2 &= \{h_{K+1}, \ldots, h_{2K}\} \\ &\ldots \\ h^g_l &= \{h_{[(n-1)/K]\times K+1}, \ldots, h_n\} \end{aligned} \tag{6}$$

where $l = [(n-1)/K] + 1$, $[\cdot]$ denotes floor operation. K denotes the number of frames in a group, or group size, as shown in Fig. 2. Thus K represents the extent of parallelization. A greater K values represents a higher level of parallelization. When $K = 1$, it is reduced to a frame-level autoregressive acoustic decoder. It is worth mentioning that the last group is padded with zero if it has less than K frames.

We then feed the fusion layer with K all-zero vectors of special symbols $\langle s \rangle$ to kick-start the regressive process, that generates group-level Mel-spectrum $\mathbf{y}^g = \{y^g_1, y^g_2, \ldots, y^g_l\}$ group by group.

A residual connection was applied after the linear transformation operation. As can be seen from Fig. 1, the output of linear transformation layer is combined with the group-level hidden states output h^g_t to form the final fusion output f^g_t at each time step t, as formulated next,

$$\begin{aligned} f^g_t &= \text{Fusion}(h^g_t, y^g_{t-1}) \\ &= \text{ReLU}(Wh^g_t + Uy^g_{t-1}) + h^g_t \end{aligned} \tag{7}$$

where W and U are weight matrices and $t \in [1, l]$. The final output are fed to the top layer.

A similar trick, named *reduction factor*, is also used in attention-based TTS framework, such as Tacotron (Wang, Skerry-Ryan et al., 2017) and Tacotron2 (Shen et al., 2018), to reduce the training and inference time. For a detailed comparison, we also report the computational efficiency of Tacotron2 with various reduction factors in Section 4.

3.2.3. Top layer in acoustic decoder

As shown in Fig. 1, The top layer contains 2 submodules: a masked multi-head attention, and a 2-layer 1D convolutional network. Residual connections and layer normalizations are applied to them.

In the original Transformer TTS decoder, the attention mask is a lower triangular matrix (Li et al., 2019), which strictly prevents earlier decoding steps from peeping information from later steps. However, in *FastTalker*, the autoregression takes place between groups instead of frames. Therefore, instead of a frame-based causal mask, we adopt a coarse-grained lower triangular matrix as a group-based causal mask (Wang, Zhang and Chen, 2018), that allows the information within the same group to be used concurrently. A group mask and its elements can be defined as follows:

$$M[i][j] = \begin{cases} 1, & \text{if } j < ([(i-1)/K] + 1) \times K \\ 0, & \text{other} \end{cases} \tag{8}$$

For ease of understanding, we compare frame-based and group-based causal masks in Fig. 3. With a group mask, the top layer takes $F^g = \{f^g_1, f^g_2, \ldots, f^g_l\}$ as input, and output hidden sequence $H^g_{top} = \{h^g_{top,1}, h^g_{top,2}, \ldots, h^g_{top,l}\}$ group by group, which reinforces the temporal dependency between adjacent frames. When $K > 1$, group-based autoregressive decoding involves less computation than frame-by-frame decoding. Finally the group-level hidden sequence H^g_{top} is taken by the linear layer to produce output acoustic features $\mathbf{y}^g = \{y^g_1, y^g_2, \ldots, y^g_l\}$.

3.2.4. Linear layer in acoustic decoder

Just like in Tacotron2 and Transformer TTS, we use a linear layer, which consists of a Mel linear and a stop linear to predict the group-level Mel-spectrum and the stop token respectively.

Specifically, the output of top layer H^g_{top} is projected through a linear transform Mel Linear to predict group-level output Mel-spectrum $\mathbf{y}^g$. At the same time, the stop linear layer is used to predict the end of utterance. With group-based autoregression, the model no longer produces Mel-spectrum acoustic features frame by frame in the same way as Tacotron2 and Transformer TTS do, but rather group by group. To be exact, Eq. (1) can be re-written as,

$$p(\mathbf{y}^g|\mathbf{x}, \theta) = \prod_{t=1}^{l} p(y^g_t|y^g_{<t}, \mathbf{x}, \theta) \tag{9}$$

Streaming the frames in the group sequence $\mathbf{y}^g = \{y^g_1, y^g_2, \ldots, y^g_l\}$, we obtain a final sequence of output acoustic features $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$. We note that we apply position encoding to x, $\hat{x}$ and y^g in the same way as Ren et al. (2019).

4. Experiments

To show the effectiveness of our proposed TTS model, FastTalker, we conduct experiments on LJSpeech database (Ito & Johnson, 2017) that consists of 13,100 short clips with a total of nearly 24 h of speech from one single speaker reading about 7 non-fiction books. We conduct both objective and subjective evaluations to assess the performance of our proposed model in a comparative study. We develop 3 competitive baselines that are: (1) Tacotron2, (2) Transformer TTS, and (3) FastSpeech. The implementation details of these baseline frameworks are provided as follows:

- Tacotron2: Tacotron2 is a typical autoregressive neural TTS system, that has a recurrent network as its encoder and decoder. We follow Shen et al. (2018) for the configuration of Tacotron2 implementation. The reduction factor (r) is set between 1 and 5.
- Transformer TTS: Transformer TTS employs a self-attention architecture that allows for parallel processing during teacher-forcing training. However, it has an autoregressive inference process at run-time. We follow the configuration in Li et al. (2019), that consists of a 6-layer encoder, a 6-layer decoder, and a mel-postnet.
- FastSpeech: FastSpeech is built with Feed-Forward Transformer (FFT) that is non-autoregressive. We follow Ren et al.

Table 1
Comparison of four encoder–decoder neural TTS systems in terms of regression strategy. During inference, only non-autoregressive network supports parallel computation.

System	Decoder	No. of parameters (million)
Tacotron2 (Shen et al., 2018)	Autoregression	26.64
Transformer TTS (Li et al., 2019)	Autoregression	22.01
FastSpeech (Ren et al., 2019)	Non-autoregression	22.23
FastTalker	**Shallow and group autoregression**	24.18

(2019) for the configuration. For a fair comparison, we do not implement sequence-level knowledge distillation (Kim & Rush, 2016) in this model.

- FastTalker: The proposed neural TTS with shallow and group autoregression. The group size (K) is also set between 1 and 5.

Table 1 summarizes the regression strategies of the systems. All systems are implemented using the ESPnet-TTS toolkit (Hayashi et al., 2020).

We also compare these systems with the ground truth speech in listening tests. In the inference process, the output Mel-spectrums are transformed into audio samples using a pre-trained WaveNet vocoder (Oord et al., 2016). The speech samples are made available at this demo link: https://ttslr.github.io/FastTalker/.

4.1. Experiments setup

We use $N = 6$ FFT blocks in the encoder layer, and $N = 5$ FFT blocks in the decoder layer for *FastTalker*. We follow FastSpeech (Ren et al., 2019) and set the number of hidden units as 1536, the attention dimensions as 384. We design 4 heads for multi-head attention in the encoder layer, decoder layer and top layer. In the FFT block, we use 1-D convolution with kernel size 3.

The input text is first converted into a sequence of 256-dimensional character embeddings before presenting as input to the encoder. The decoder generates a sequence of 80-channel Mel-spectrum acoustic features as output. Preparing the training data, we extract Mel-spectrum acoustic features with a frame size of 50 ms and 12.5 ms frame shift, that are further normalized to zero-mean and unit-variance, to serve as the reference target. We pre-train an autoregressive Transformer TTS model as the teacher model to extract the character duration to supervise the training of length regulator.

We train the models using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$. We follow Vaswani et al. (2017) and vary the learning rate R over the course of training as follows,

$$R = \frac{\min\left(step^{-0.5}, step \cdot warmup\_steps^{-1.5}\right)}{384^{-0.5}} \tag{10}$$

where $warmup\_steps = 4000$. All models are trained with 150k steps to ensure complete convergence.

4.2. Objective evaluation

We employ Mel-spectrum distortion (MCD) (Kubichek, 1993) to measure the spectral distance between the synthesized and reference Mel-spectrum features. The MCD between a reference target and a synthesized utterance of n frames is calculated as:

$$\mathrm{MCD} = \frac{10\sqrt{2}}{\ln 10}\frac{1}{n}\sum_{t=1}^{n}\left(\frac{1}{\mathcal{N}}\sqrt{\sum_{k=1}^{\mathcal{N}}(y_{t,k}-\hat{y}_{t,k})^2}\right) \tag{11}$$

where $\mathcal{N}$ represents the dimension of the Mel-spectrum, $y_{t,k}$ denotes the kth Mel-spectrum component in tth frame for the reference target Mel-spectrum, and $\hat{y}_{t,k}$ for the synthesized Mel-spectrum. We note that lower MCD value indicates smaller distortion, thus better performance.

We use Root Mean Squared Error (RMSE) as prosody evaluation metric (Sisman & Li, 2018) that is calculated as follows,

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\mathrm{F0_t} - \widehat{\mathrm{F0}}_t\right)^2} \tag{12}$$

where $\mathrm{F0_t}$ and $\widehat{\mathrm{F0}}_t$ denote the reference and synthesized F0 at tth frame for a n frames output. We note that lower RMSE value suggests that the two F0 contours are more similar. As there is no guarantee that synthesized speech and reference speech have the same length, we apply dynamic time warping (DTW) (Müller, 2007) to align speech pairs and calculate MCD and RMSE between the acoustic features.

In Table 2, we report the MCD and RMSE results in a comparative study. To start with, we observe that Tacotron2 with various reduction factors and Transformer TTS outperform the FastSpeech in terms of spectral and prosody modeling, which is consistent with the prior studies (Ren et al., 2019; Shen et al., 2018).

The experiments on the FastTalker with various group size show that it brings improvements over the FastSpeech in terms of MCD and RMSE, and even achieves comparable performance to the autoregressive TTS models, such as Tacotron2 ($r = \{1, 2, 3, 4, 5\}$) and Transformer TTS, which proves the capacity of proposed FastTalker. For example, we note that the proposed FastTalker ($K = \{1, 2, 3, 4, 5\}$) outperforms FastSpeech consistently in terms of both MCD and RMSE as reported in Table 2. FastTalker also achieves comparable results to that of Tacotron2 ($r = \{1, 2, 3, 4, 5\}$) and Transformer TTS, which we believe is encouraging.

4.3. Subjective evaluation

We conduct listening experiments for subjective evaluation. We first evaluate Tacotron2, Transformer TTS, FastSpeech, *FastTalker*, and ground truth natural speech in terms of mean opinion score (MOS).

The listeners rate of the quality is at a 5-point scale: "5" for excellent, "4" for good, "3" for fair, "2" for poor, and "1" for bad. The MOS values are calculated by taking the arithmetic average of all scores assigned the subjects who have passed the validation question test. We keep the linguistic content the same among different models so as to exclude other interference factors. 15 subjects participate in these experiments, each listening to 300 synthesized speech samples.

As shown in Table 2, the MOS scores of the *FastTalker* with various group size outperforms the FastSpeech by a large margin. Similar with the results of the previous objective evaluation, *FastTalker* ($K = \{1, 2, 3, 4, 5\}$) outperforms FastSpeech baseline, and achieves comparable results to that of Tacotron2 ($r = \{1, 2, 3, 4, 5\}$) and Transformer TTS in terms of voice naturalness. The results clearly show that the shallow autoregression strategy, as implemented in *FastDecoder*, effectively model the temporal dependency, that defeats the full non-autoregression counterpart.

Table 2
Summary of the MCD, RMSE, MOS results and inference speedup in subjective and objective evaluation. The MOS results are calculated with 95% confidence intervals computed from the *t-test* (Forsythe, 1977). (AR: autoregression; NAR: non-autoregression).

	System	MCD [dB]	RMSE [Hz]	MOS	Inference speedup
AR	Tacotron2 (r = 1) (Shen et al., 2018)	8.11	18.03	4.577 ± 0.067	1.00×
	Tacotron2 (r = 2)	8.14	18.05	4.574 ± 0.071	1.49×
	Tacotron2 (r = 3)	8.18	18.11	4.573 ± 0.053	1.89×
	Tacotron2 (r = 4)	8.16	18.09	4.570 ± 0.060	2.21×
	Tacotron2 (r = 5)	8.19	18.15	4.568 ± 0.066	2.75×
	Transformer TTS (Li et al., 2019)	8.20	18.09	4.580 ± 0.059	4.35×
NAR	FastSpeech (Ren et al., 2019)	8.74	20.95	4.285 ± 0.086	59.24×
Ours	FastTalker ($K = 1$)	8.23	18.22	4.559 ± 0.081	8.36×
	FastTalker ($K = 2$)	8.27	18.22	4.551 ± 0.073	9.04×
	FastTalker ($K = 3$)	8.20	18.27	4.547 ± 0.091	9.59×
	FastTalker ($K = 4$)	8.22	18.30	4.540 ± 0.087	10.14×
	FastTalker ($K = 5$)	8.24	18.23	4.533 ± 0.072	11.92×
	Ground truth	–	–	**4.583 ± 0.084**	–

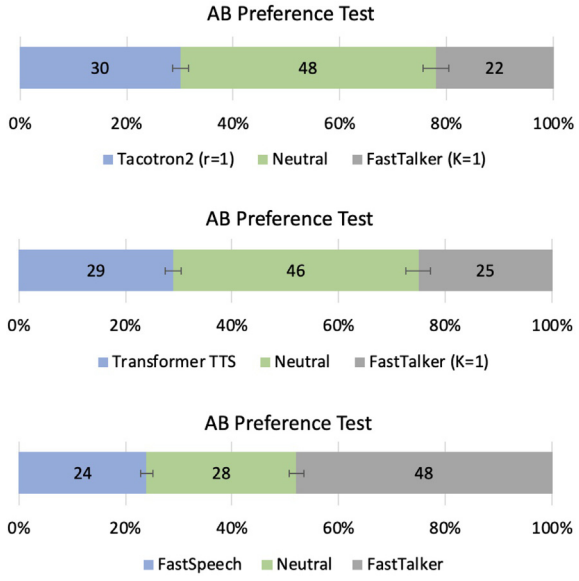

Fig. 4. The AB preference test for naturalness evaluation by 15 listeners, with 95% confidence interval computed from the *t-test* (Forsythe, 1977). The *p*-values are $3.1023e^{-21}$, $1.5022e^{-21}$ and $2.1002e^{-20}$ respectively.

Note that we evaluate *FastTalker* with K values ranging from 1 to 5. The *group size*, or K, in group-based shallow autoregression represents the number of frames within one group. In particular, $K = 1$ means the shallow autoregression module in the *FastDecoder* runs with complete autoregressive manner. If K is equal to the length of the target sequence, then the decoding strategy of *FastDecoder* is equivalent to complete non-autoregressive. The results in Table 2 show that *FastTalker* achieves the best result when $K = 1$, with an MOS of 4.559.

With larger ($K \geq 2$) group sizes, there are more frames in a group, therefore, fewer groups. The shallow regression implementation aims to reinforce the temporal dependency while the group regression strategy seeks to accelerate the inference. We note that it is more difficult to model the temporal dependency over large groups than small groups. Increasing group size may adversely impact the ability of the temporal modeling. According to the results, the MOS scores of the FastTalker gradually decreases as K increases, which is also consistent with this intuition.

We further conduct AB preference tests which adopt Tacotron2 ($r = 1$), Transformer TTS, FastSpeech and FastTalker ($K = 1$), where the listeners are asked to compare the quality and naturalness of the synthesized speech samples between a pair of systems, and select the better one. 15 listeners were invited to participate in all the tests, each listening to 100 pairs synthesized speech samples. In Fig. 4, the AB preference tests show that most listeners choose neutral (or no preference) when they access the speech pairs generated from *FastTalker* ($K = 1$) and autoregressive TTS models, that are Tacotron2 ($r = 1$) and Transformer TTS. More importantly, it is encouraging to see that *FastTalker* outperforms the FastSpeech model. Specifically, the listeners prefer *FastTalker* ($K = 1$) model over FastSpeech consistently that further validates the proposed regression strategy.

4.4. Inference acceleration

We follow Peng, Ping, Song, and Zhao (2020) and test inference speed of all models on a single Tesla P40 GPU. 15 sentences were randomly selected from the test set as the test samples in this section. The average speech duration of the utterances is 7.38 s. We run inference for 50 runs on each of the 15 sentences (batch size is set to 1) and summarize synthesis speedup for all systems in the last column of Table 2.

We can found that FastTalker ($K = \{1, 2, 3, 4, 5\}$) runs significantly faster than Tacotron2 ($r = \{1, 2, 3, 4, 5\}$) and Transformer TTS. For example, our FastTalker ($K = 1$) brings about 8.36 times speedup over Tacotron2 ($r = 1$). Besides, there is no doubt that FastTalker with $K = \{2, 3, 4, 5\}$ brings even more significantly speedup than $K = 1$. In a nutshell, the proposed FastTalker with $K = \{1, 2, 3, 4, 5\}$ consistently outperforms the autoregressive models by achieving average inference speedup of 9.81 times, which is much higher than that of the Tacotron2 with $r \geq 2$ (2.09 times), and Transformer TTS (4.35 times). The reason is that the FastTalker benefit from both non-autoregression and group-based shallow autoregression strategy for computational efficiency. Different from the complete autoregressive mechanism, it just involves a few autoregressive computations brings from the shallow autoregression.

It is noted that FastSpeech brings about 59.24 times speedup over Tacotron2 ($r = 1$) and achieves the fastest synthesis speed rely on the complete non-autoregressive mechanism.

All the above observations validate that the *FastTalker* achieves a good balance between audio quality and decoding speed. With the shallow and group autoregression, the *FastTalker* can learn the temporal dependency information and generate more natural acoustic features with fast-decoding property.

5. Conclusion

In this work, we propose a novel neural TTS architecture denoted as *FastTalker*. The proposed *FastTalker* with shallow and

group autoregression effectively models the temporal dependency with fast decoding property. Experimental results show that FastTalker, on par with autoregressive TTS, outperforms the non-autoregressive counterpart significantly in terms of voice quality and naturalness. At the same time, the group-based shallow autoregression brings out encouraging inference speedup. The FastTalker achieves a good balance between audio quality and decoding speed. We note that the proposed idea is also applicable to other encoder–decoder speech synthesis systems, which will be our future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the associate editor and the reviewers for their useful feedback that improved this paper.

The research by Rui Liu and Berrak Sisman is funded by SUTD Start-up Grant Artificial Intelligence for Human Voice Conversion (SRG ISTD 2020 158) and SUTD AI Grant, titled 'The Understanding and Synthesis of Expressive Speech by AI'.

This research by Haizhou Li is supported by the National Research Foundation, Singapore under its AI Singapore Programme (Award No: AISG-GC-2019-002) and (Award No: AISG-100E-2018-006), and its National Robotics Programme (Grant No. 192 25 00054), and by RIE2020 Advanced Manufacturing and Engineering Programmatic Grants A1687b0033, and A18A2b0046.

References

Bahdanau, Dzmitry, Cho, Kyunghyun, & Bengio, Yoshua (2015a). Neural machine translation by jointly learning to align and translate. In *3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference track proceedings*.

Bahdanau, Dzmitry, Cho, Kyunghyun, & Bengio, Yoshua (2015b). Neural machine translation by jointly learning to align and translate. In *International conference on learning representations, ICLR 2015*.

Black, Alan W., & Campbell, Nick (1995). Optimising selection of units from speech databases for concatenative synthesis. In *4th European conference on speech communication and technology* (pp. 581–584).

Cho, Kyunghyun, Merriënboer, Bartvan, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, et al. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1724–1734).

Duda, Richard O., Hart, Peter E., & Stork, David G. (2012). *Pattern classification*. John Wiley & Sons.

Forsythe, George Elmer (1977). *Prentice-Hall series in automatic computation: vol. 259, Computer methods for mathematical computations*.

Greff, Klaus, Srivastava, Rupesh K., Koutník, Jan, Steunebrink, Bas R., & Schmidhuber, Jürgen (2016). Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232.

Gu, Jiatao, Bradbury, James, Xiong, Caiming, Li, Victor O. K., & Socher, Richard (2018). Non-autoregressive neural machine translation. In *International conference on learning representations, ICLR 2018*.

Guo, Junliang, Tan, Xu, He, Di, Qin, Tao, Xu, Linli, & Liu, Tie-Yan (2019). Non-autoregressive neural machine translation with enhanced decoder input. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 3723–3730).

Habib, Raza, Mariooryad, Soroosh, Shannon, Matt, Battenberg, Eric, Skerry-Ryan, R. J., Stanton, Daisy, et al. (2020). Semi-supervised generative modeling for controllable speech synthesis. In *8th International conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Hayashi, Tomoki, Tamamori, Akira, Kobayashi, Kazuhiro, Takeda, Kazuya, & Toda, Tomoki (2017). An investigation of multi-speaker training for wavenet vocoder. In *2017 IEEE automatic speech recognition and understanding workshop (ASRU)* (pp. 712–718). IEEE.

Hayashi, Tomoki, Yamamoto, Ryuichi, Inoue, Katsuki, Yoshimura, Takenori, Watanabe, Shinji, Toda, Tomoki, et al. (2020). Espnet-tts: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 7654–7658). IEEE.

Hochreiter, Sepp, & Schmidhuber, Jürgen (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hsu, Wei-Ning, Zhang, Yu, Weiss, Ron J., Zen, Heiga, Wu, Yonghui, Wang, Yuxuan, et al. (2019). Hierarchical generative modeling for controllable speech synthesis. In *7th International conference on learning representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019*.

Hunt, Andrew J., & Black, Alan W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *ICASSP 1996-1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings (ICASSP)* (pp. 373–376). IEEE.

Ito, Keith, & Johnson, Linda (2017). The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/.

Jordan, Michael I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology, Vol. 121* (pp. 471–495). Elsevier.

Karita, Shigeki, Chen, Nanxin, Hayashi, Tomoki, Hori, Takaaki, Inaguma, Hirofumi, Jiang, Ziyan, et al. (2019). A comparative study on transformer vs RNN in speech applications. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)* (pp. 449–456). IEEE.

Kim, Jaehyeon, Kim, Sungwon, Kong, Jungil, & Yoon, Sungroh (2020). Glow-tts: A generative flow for text-to-speech via monotonic alignment search. In *Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*.

Kim, Yoon, & Rush, Alexander M. (2016). Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016* (pp. 1317–1327). The Association for Computational Linguistics.

Kobayashi, Kazuhiro, Hayashi, Tomoki, Tamamori, Akira, & Toda, Tomoki (2017). Statistical voice conversion with wavenet-based waveform generation. In *Proc. Interspeech 2017* (pp. 1138–1142).

Kubichek, Robert F. (1993). Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE pacific rim conference on communications computers and signal processing, Vol. 1* (pp. 125–128). IEEE.

Li, Naihan, Liu, Shujie, Liu, Yanqing, Zhao, Sheng, & Liu, Ming (2019). Neural speech synthesis with transformer network. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 6706–6713).

Li, Naihan, Liu, Shujie, Liu, Yanqing, Zhao, Sheng, Liu, Ming, & Zhou, Ming (2020). Moboaligner: A neural alignment model for non-autoregressive TTS with monotonic boundary search. In *Interspeech 2020, 21st annual conference of the international speech communication association, virtual event, Shanghai, China, 25-29 October 2020* (pp. 3999–4003). ISCA.

Li, Yanyang, Wang, Qiang, Xiao, Tong, Liu, Tongran, & Zhu, Jingbo (2020). Neural machine translation with joint representation. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8285–8292).

Li, Jingdong, Zhang, Hui, Liu, Rui, Zhang, Xueliang, & Bao, Feilong (2018). End-to-end mongolian text-to-speech system. In *2018 11th international symposium on chinese spoken language processing (ISCSLP)* (pp. 483–487). IEEE.

Ling, Zhen-Hua, Kang, Shi-Yin, Zen, Heiga, Senior, Andrew, Schuster, Mike, Qian, Xiao-Jun, 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.

Ling, Zhen-Hua, & Zhou, Zhi-Ping (2018). Unit selection speech synthesis using frame-sized speech segments and neural network based acoustic models. *Journal of Signal Processing Systems*, *90*(7), 1053–1062.

Liu, Rui, Bao, Feilong, Gao, Guanglai, & Wang, Yonghe (2017). Mongolian text-to-speech system based on deep neural network. In *National conference on man-machine speech communication* (pp. 99–108). Springer.

Liu, Rui, Sisman, Berrak, Bao, Feilong, Gao, Guanglai, & Li, Haizhou (2020a). Modeling prosodic phrasing with multi-task learning in tacotron-based tts. *IEEE Signal Processing Letters*, *27*, 1470–1474.

Liu, Rui, Sisman, Berrak, Bao, Feilong, Gao, Guanglai, & Li, Haizhou (2020b). WaveTTS: Tacotron-based TTS with joint time-frequency domain loss. In *Proc. Odyssey 2020 the speaker and language recognition workshop* (pp. 245–251).

Liu, Rui, Sisman, Berrak, Bao, Feilong, Yang, Jichen, Gao, Guanglai, & Li, Haizhou (2021). Exploiting morphological and phonological features to improve prosodic phrasing for mongolian speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 274–285. http://dx.doi.org/10.1109/TASLP.2020.3040523.

Liu, Rui, Sisman, Berrak, Gao, Guanglai, & Li, Haizhou (2020). Expressive tts training with frame and style reconstruction loss. arXiv preprint arXiv:2008.01490.

Liu, Rui, Sisman, Berrak, & Li, Haizhou (2020). Graphspeech: Syntax-aware graph attention network for neural speech synthesis. arXiv preprint arXiv:2010.12423.

Liu, Rui, Sisman, Berrak, Li, Jingdong, Bao, Feilong, Gao, Guanglai, & Li, Haizhou (2020). Teacher-student training for robust tacotron-based TTS. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6274–6278). IEEE.

Luong, Thang, Pham, Hieu, & Manning, Christopher D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP)*, Lisbon, Portugal (pp. 1412–1421).

Matsunaga, Noriyuki, Ohtani, Yamato, & Hirahara, Tatsuya (2019). Loss function considering temporal sequence for feed-forward neural network–fundamental frequency case. In *Proc. 10th ISCA speech synthesis workshop* (pp. 143–148).

Merritt, Thomas, Clark, Robert A. J., Wu, Zhizheng, Yamagishi, Junichi, & King, Simon (2016). Deep neural network-guided unit selection synthesis. In *ICASSP 2016-2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5145–5149). IEEE.

Müller, Meinard (2007). Dynamic time warping. Information retrieval for music and motion. (pp. 69–84).

Oord, Aaron van den, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, et al. (2016). Wavenet: A generative model for raw audio. In *2016 ISCA speech synthesis workshop* (pp. 125–133).

Peng, Kainan, Ping, Wei, Song, Zhao, & Zhao, Kexin (2020). Non-autoregressive neural text-to-speech. In *International conference on machine learning* (pp. 7586–7598). PMLR.

Prenger, Ryan, Valle, Rafael, & Catanzaro, Bryan (2019). Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 3617–3621).

Ren, Yi, Hu, Chenxu, Tan, Xu, Qin, Tao, Zhao, Sheng, Zhao, Zhou, et al. (2020). Fastspeech 2: Fast and high-quality end-to-end text to speech. arXiv preprint arXiv:2006.04558.

Ren, Yi, Ruan, Yangjun, Tan, Xu, Qin, Tao, Zhao, Sheng, Zhao, Zhou, et al. (2019). Fastspeech: Fast, robust and controllable text to speech. In *Advances in neural information processing systems* (pp. 3171–3180).

Shannon, S. M., & Byrne, William (2009). Autoregressive hmms for speech synthesis. In *Proc. Interspeech 2009*.

Shao, Chenze, Feng, Yang, Zhang, Jinchao, Meng, Fandong, Chen, Xilin, & Zhou, Jie (2019). Retrieving sequential information for non-autoregressive neural machine translation. In *Proceedings of the 57th annual meeting of the association for computational linguistics (ACL)*, Florence, Italy (pp. 3013–3024).

Shen, Jonathan, Pang, Ruoming, Weiss, Ron J., Schuster, Mike, Jaitly, Navdeep, Yang, Zongheng, et al. (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP 208-2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4779–4783). IEEE.

Siddhant, Aditya, Johnson, Melvin, Tsai, Henry, Ari, Naveen, Riesa, Jason, Bapna, Ankur, et al. (2020). Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8854–8861).

Sisman, Berrak, King, Simon, Yamagishi, Junichi, & Li, Haizhou (2021). An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 132–157.

Sisman, Berrak, & Li, Haizhou (2018). Wavelet analysis of speaker dependent and independent prosody for voice conversion. In *Interspeech* (pp. 52–56).

Sisman, Berrak, Zhang, Mingyang, & Li, Haizhou (2018). A voice conversion framework with tandem feature sparse representation and speaker-adapted wavenet vocoder. In *Proc. Interspeech 2018* (pp. 1978–1982).

Sisman, Berrak, Zhang, Mingyang, & Li, Haizhou (2019). Group sparse representation with wavenet vocoder adaptation for spectrum and prosody conversion. *IEEE/ACM Transactions on Audio, Speech and Language Processing*.

Skerry-Ryan, R. J., Battenberg, Eric, Xiao, Ying, Wang, Yuxuan, Stanton, Daisy, Shor, Joel, et al. (2018). Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *Proceedings of the 35th international conference on machine learning, Vol. 80* (pp. 4693–4702). PMLR, 2018.

Sun, Guangzhi, Zhang, Yu, Weiss, Ron J., Cao, Yuan, Zen, Heiga, Rosenberg, Andrew, et al. (2020). Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6699–6703). IEEE.

Sun, Guangzhi, Zhang, Yu, Weiss, Ron J., Cao, Yuan, Zen, Heiga, & Wu, Yonghui (2020). Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6264–6268). IEEE.

Sutskever, Ilya, Vinyals, Oriol, & Le, Quoc V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Tamamori, Akira, Hayashi, Tomoki, Kobayashi, Kazuhiro, Takeda, Kazuya, & Toda, Tomoki (2017). Speaker-dependent wavenet vocoder. In *Proc. Interspeech 2017* (pp. 1118–1122).

Taylor, Paul (2009). *Text-to-speech synthesis*. Cambridge university press.

Titze, Ingo R., & Martin, Daniel W. (1998). Principles of voice production.

Tokuda, Keiichi, Nankaku, Yoshihiko, Toda, Tomoki, Zen, Heiga, Yamagishi, Junichi, & Oura, Keiichiro (2013). Speech synthesis based on hidden markov models. *Proceedings of the IEEE*, *101*(5), 1234–1252.

Tokuda, Keiichi, Yoshimura, Takayoshi, Masuko, Takashi, Kobayashi, Takao, & Kitamura, Tadashi (2000). Speech parameter generation algorithms for hmm-based speech synthesis. In *2000 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 00CH37100), Vol. 3* (pp. 1315–1318). IEEE.

Tu, Zhaopeng, Liu, Yang, Shang, Lifeng, Liu, Xiaohua, & Li, Hang (2017). Neural machine translation with reconstruction. In *Proceedings of the thirty-first AAAI conference on artificial intelligence, AAAI'17* (pp. 3097–3103).

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wang, Yuxuan, Skerry-Ryan, R. J., Stanton, Daisy, Wu, Yonghui, Weiss, Ron J., Jaitly, Navdeep, et al. (2017). Tacotron: A fully end-to-end text-to-speech synthesis model. In *Proc. Interspeech 2017* (pp. 4006–4010).

Wang, Xin, Takaki, Shinji, & Yamagishi, Junichi (2017). An autoregressive recurrent mixture density network for parametric speech synthesis. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4895–4899). IEEE.

Wang, Xin, Takaki, Shinji, & Yamagishi, Junichi (2018). Autoregressive neural f0 model for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(8), 1406–1419.

Wang, Xin, Takaki, Shinji, Yamagishi, Junichi, King, Simon, & Tokuda, Keiichi (2019). A vector quantized variational autoencoder (vq-vae) autoregressive neural f_0 model for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 157–170.

Wang, Chunqi, Zhang, Ji, & Chen, Haiqing (2018). Semi-autoregressive neural machine translation. In *Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP)*, Brussels, Belgium, October-November (pp. 479–488).

Watts, Oliver, Henter, Gustav Eje, Fong, Jason, & Valentini-Botinhao, Cassia (2019). Where do the improvements come from in sequence-to-sequence neural tts?. In *2019 ISCA speech synthesis workshop (SSW), Vol. 10* (pp. 217–222).

Williams, Ronald J., & Zipser, David (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270–280.

Wu, Yi-Chiao, Hayashi, Tomoki, Okamoto, Takuma, Kawai, Hisashi, & Toda, Tomoki (2020a). Quasi-periodic parallel wavegan: A non-autoregressive raw waveform generative model with pitch-dependent dilated convolution neural network. arXiv preprint arXiv:2007.12955.

Wu, Yi-Chiao, Hayashi, Tomoki, Okamoto, Takuma, Kawai, Hisashi, & Toda, Tomoki (2020b). Quasi-periodic parallel wavegan vocoder: A non-autoregressive pitch-dependent dilated convolution model for parametric speech generation. arXiv preprint arXiv:2005.08654.

Wu, Zhizheng, & King, Simon (2016). Investigating gated recurrent networks for speech synthesis. In *ICASSP 2016-2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5140–5144). IEEE.

Yang, Shan, Lu, Heng, Kang, Shiyin, Xue, Liumeng, Xiao, Jinba, Su, Dan, et al. (2020). On the localness modeling for the self-attention based end-to-end speech synthesis. *Neural Networks*, *125*, 121–130.

Yang, Geng, Yang, Shan, Liu, Kai, Fang, Peng, Chen, Wei, & Xie, Lei (2020). Multi-band melgan: Faster waveform generation for high-quality text-to-speech. arXiv preprint arXiv:2005.05106.

Yoshimura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In *Proc. EuroSpeech 1999* (pp. 2347–2350).

Zen, Heiga, & Sak, Haşim (2015). Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *ICASSP 2015-2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4470–4474). IEEE.

Zen, Heiga, Senior, Andrew, & Schuster, Mike (2013). Statistical parametric speech synthesis using deep neural networks. In *ICASSP 2013-2013 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 7962–7966). IEEE.

Zen, Heiga, Tokuda, Keiichi, & Black, Alan W. (2009). Statistical parametric speech synthesis. *Speech Communication*, *51*(11), 1039–1064.

Zen, Heiga, Tokuda, Keiichi, & Kitamura, Tadashi (2007). Reformulating the hmm as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. *Computer Speech and Language*, *21*(1), 153–173.

Zeng, Zhen, Wang, Jianzong, Cheng, Ning, Xia, Tian, & Xiao, Jing (2020). Alignntts: Efficient feed-forward text-to-speech system without explicit alignment. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6714–6718). IEEE.

Zhang, Zhaoyan (2016). Mechanics of human voice production and control. *The Journal of the Acoustical Society of America*, *140*(4), 2614–2635.

Zheng, Zaixiang, Zhou, Hao, Huang, Shujian, Mou, Lili, Dai, Xinyu, Chen, Jiajun, et al. (2018). Modeling past and future for neural machine translation. *Transactions of the Association for Computational Linguistics*, *6*, 145–157.