QUASI-FULLY CONVOLUTIONAL NEURAL NETWORK WITH VARIATIONAL INFERENCE FOR SPEECH SYNTHESIS

Mu Wang[1]*, Xixin Wu*[2]*, Zhiyong Wu*[12]*, Shiyin Kang*[3]*, Deyi Tuo*[3]*, Guangzhi Li*[3]*,*
Dan Su[3]*, Dong Yu*[3]*, Helen Meng*[12]

[1]Tsinghua-CUHK Joint Research Center for Media Science, Technologies and Systems,
Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[2]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China
[3]Tencent AI Lab, Shenzhen, China
wangmu16@mails.tsinghua.edu.cn, {wuxx, zywu, hmmeng}@se.cuhk.edu.hk,
{shiyinkang, deyituo, guangzhilei, dansu, dyu}@tencent.com

ABSTRACT

Recurrent neural networks, such as gated recurrent units (GRUs) and long short-term memory (LSTM), are widely used on acoustic modeling for speech synthesis. However, such sequential generating processes are not friendly to today's massively parallel computing devices. We introduce a fully convolutional neural network (CNN) model, which can effiently run on parallel processers, for speech synthesis. To improve the quality of the generated acoustic features, we strengthen our model with variational inference. We also use quasi-recurrent neural networks (QRNNs) to smoothen the generated acoustic features. Finally, a high-quality parallel WaveNet model is used to generate audio samples. Our contributions are twofold. First, we show that CNNs with variational inference can generate highly natural speech on a par with end-to-end models; the use of QRNNs further improves the synthetic quality by reducing trembling of generated acoustic features and introduces very little runtime overheads. Second, we show some techniques to further speed up the sampling process of the parallel WaveNet model.

Index Terms— convolutional neural network (CNN), quasi-fully recurrent neural network (QRNN), variational inference, parallel WaveNet, text-to-speech (TTS) synthesis

1. INTRODUCTION

The basic framework of statistical parametric speech synthesis (SPSS) [1] consists of two parts: a) an acoustic model used to predict acoustic features from given linguistic features; b) a vocoder used to generate waveforms from acoustic features.

Traditional acoustic models mainly refer to the hidden Markov model (HMM) [2]. Deep neural networks (DNNs), recurrent neural networks (RNNs), and related variants are popular as acoustic models in recent years [3, 4, 5]. Although the RNN-based acoustic models can achieve great results, they are not friendly to today's massively parallel computers. Tacotron [6], a complicated squence-to-sequence model utilizing attention mechanism, beats a production parametric system in terms of naturalness. Tacotron2 [7], using mel spectrograms as the intermediate feature to connect the acoustic feature generator and a neural vocoder, yields natural sounding speech that approaches the audio delity of real human speech. However, the end-to-end approach is known to be hard to control compared with the traditional multi-stage TTS pipeline.

Recent research shows that well-designed convolutional neural networks can outperform generic recurrent architectures, such as deep LSTM and GRUs, in several sequence modeling tasks [8]. Variational auto-encoders (VAEs) [9], as one kind of deep generative models, have shown promising results in generating many kinds of complicated data, including handwritten digits, faces, house numbers, CIFAR images, physical models of scenes, segmentation, and predicting the future from static images [10]. Although VAEs is known to be able to improve the synthetic samples, there is no successful result reported on using pure variational architectures to generate speech from linguistic features. WaveNet [11] is a high-quality neural vocoder, which auto-regressively generates audio samples from linguistic features. Parallel WaveNet [12], on the other hand, can generate audio samples in parallel.

In this paper, we propose a quasi-fully convolutional neural networks model with variational inference (QFCVI) for acoustic modeling. The QFCVI model gets linguistic features as input and outputs mel spectrograms. On GPUs, the QFCVI model runs much faster than RNN-based acoustic models. We utilize quasi-recurrent neural networks (QRNNs) [13] to smoothen the generated acoustic features. QRNNs actually consist of convolutional layers and a minimalist recurrent pooling function, so they are much faster than RNNs. The generated mel spectrograms are fed into a parallel WaveNet model to synthesize waveforms. Our model implicitly predicts fundamental frequencies (F0s). Compared with directly generating waveforms from linguistic features and F0s, our approach requires much less computing resources for model training. Subjective listening tests show that our QFCVI model can achieve a high degree of naturalness for standard Chinese speech synthesis on a par with an improved Tacotron model. We also come up with some techniques to further speed up the parallel WaveNet model.

2. APPROACH

In this section, we introduce in detail the proposed quasi-fully convolutional neural network with variants inference (QFCVI) model, which is based on the variational auto-encoder (VAE) architecture and consists of a quasi-fully convolutional encoder and decoder. The

978-1-5386-4658-8/18/$31.00 ©2019 IEEE

neural vocoder is also briefly described.

2.1. Variational Auto-Encoder Architecture

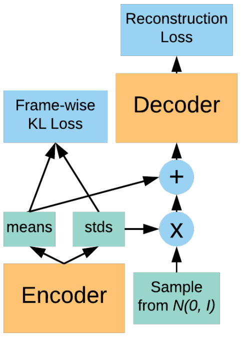


Fig. 1. VAE Architecture. We omit the inputs and outputs of the encoder and decoder for simplicity.

The basic optimization object of conditional VAEs is to minimize the following loss:

$$\mathcal{L} = \mathbb{E}_{z\sim Q}[-logP(Y|z,X)] + \mathcal{D}[Q(z|Y,X)||P(z|X)] \quad (1)$$

where X is linguistic features, and Y is acoustic features, in our model. For simplicity, we use another view of the above formula. We treat the KL distance loss term as a regularization to the model [10].

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda\mathcal{L}_{KL} \quad (2)$$

The total loss is the sum of two terms: a) acoustic feature reconstruction loss; b) KL distance as regularization, where λ is a scalar hyperparameter. Empirically, we set $\lambda = 0.2$.

2.1.1. Reconstruction Loss

We use a squared l2 norm distance (i.e. MSE loss) as the reconstruction loss.

$$\mathcal{L}_{recon} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{z_t\sim Q}||\hat{Y_t}(X_t,z_t) - Y_t||_2^2 \quad (3)$$

$$\approx \frac{1}{CT}\sum_{t=1}^{T}\sum_{c=1}^{C}||\hat{Y_{t,c}}(X_t,z_{t,c}) - Y_t||_2^2 \quad (4)$$

where T is the number of frames in a sequence, C is the number of sample times, $z_{t,c}$ is a sample from the output distribution (Q) of the encoder at the t-th frame step. Empirically, we set $C = 10$.

2.1.2. Frame-wise KL Loss

Instead of using a single latent vector for the whole sequence, the model infers latent vectors at each frame step.

$$\mathcal{L}_{KL} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{D}[Q(z_t|Y_t,X_t)||P(z_t|X_t)] \quad (5)$$

where $Q(z|Y,X)$ is the output distribution of the encoder, $P(z|X)$ is the true prior. In the context of VAEs, we assume $P(z|X)$ is $\mathcal{N}(0,I)$.

$$\mathcal{L}_{KL} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{D}[Q(z_t|Y_t,X_t)||\mathcal{N}(0,I)] \quad (6)$$

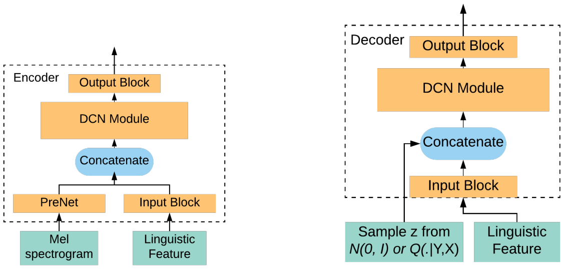
$$= \frac{1}{T}\sum_{t=1}^{T}\mathcal{D}[\mathcal{N}(\mu_t,s_t^2))||\mathcal{N}(0,I)] \quad (7)$$

$$= \frac{1}{2T}\sum_{t=1}^{T}[tr(s_t^2) + \mu_t^T\mu_t - K - log(det(s_t^2))] \quad (8)$$

$$= \frac{1}{2T}\sum_{t=1}^{T}[\sum_{k=1}^{K}s_{t,k}^2 + \sum_{k=1}^{K}\mu_{t,k}^2 - K - 2\sum_{k=1}^{K}log(s_{t,k})] \quad (9)$$

where K is the number of dimensions of the latent space. We set $K = 10$ for our model.

2.2. Quasi-Fully Convolutional Encoder & Decoder



(a) Encoder

(b) Decoder

Fig. 2. Architecture of the quasi-fully convolutional encoder and decoder.

The encoder and decoder of our model consist of almost the same submodules. The architectures of the encoder and decoder are depicted in Fig.2, while detailed setups of the submodules are shown in Fig.3.

The PreNet of the encoder consists of two dense layers, each followed by a dropout layer [6]. The Input Block is a simple 1D convolution layer with 1 reception field and 256 filters. The dropout rate is set as 0.1 during training. The Output Block is a little more complicated than the Input Block. We use a QRNN [13] layer to smoothen the generated acoustic features in frame axis. Although QRNN is recurrent-like, the calculation of its recurrent part is very cheap. A QRNN layer with fo-pooling can be formulated as following:

$$\mathbf{Z} = tanh(\mathbf{W}_z * \mathbf{X}) \quad (10)$$

$$\mathbf{F} = \sigma(\mathbf{W}_f * \mathbf{X}) \quad (11)$$

$$\mathbf{O} = \sigma(\mathbf{W}_o * \mathbf{X}) \quad (12)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot \mathbf{z}_t \quad (13)$$

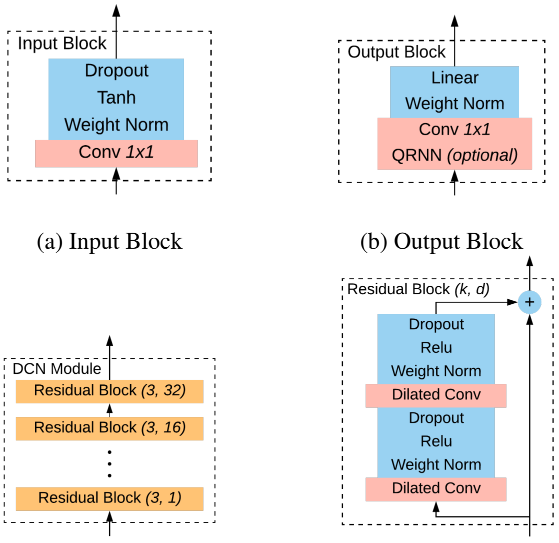
$$\mathbf{h}_t = \mathbf{o}_t \odot \mathbf{c}_t \quad (14)$$

where $*$ denotes a convolution operator, $\odot$ denotes an element-wise multiplication operator, and $\sigma(\cdot)$ is a sigmoid function. Instead of using a TensorFlow python implementation or writing a GPU kernel, we write a simple TensorFlow kernel to compute $\mathbf{c}_t$ and $\mathbf{h}_t$ using CPU. That fo-pooling computation in CPU is very fast, since the **time complexity is** $O(TN)$ [1], where T is the number of total time

[1] The time complexity of RNNs is $O(TN^2)$.

steps, and N is the number of feature channels. Note that we use QRNN in the decoder only. To speed up training, we also implement a simple CPU kernel for the backpropagation of fo-pooling. The convolution layer in the Output Block has 256 filters, the dropout rate is set as 0.1. Each convolution kernel in the QRNN layer has 128 filters and a reception field of 2.

The dilated convolutional network (DCN) module is a variant from the architecture proposed in [8]. The module consists of six Residual Blocks with increasing dilation rate from 1 to 32. The Residual Block has two 1D dilated convolution layers with 256 filters and a reception field of 3. Instead of using causal convolution layers as described in [8], we use standard dilated convolution operations. To speed up the convergence, weight normalization [14] is applied to all convolution layers in our model. The dropout rate in the Residual Block is set as 0.2.



(a) Input Block

(b) Output Block

(c) DCN Module

(d) Residual Block

Fig. 3. Detailed setup of the submodules of the encoder and decoder.

2.3. Neural Vocoder

A parallel WaveNet model, trained on ground-truth mel spectrograms, is used as the vocoder of our model. The vocoder consisted of 60 layers, with 64 filters for each residual and gating layers [12]. All parallel WaveNet models in this work are distilled from a teacher WaveNet model with 24 layers, grouped into 4 dilated residual block stacks of 6 layers [7].

3. EXPERIMENTS & RESULTS

3.1. Baselines and Conditions

A DNN system, a deep LSTM (DLSTM) system, and an end-to-end system are used as baselines. The DNN and DLSTM systems are used at Tencent AI Lab in production. The DNN system consists of a DNN-based duration model and a DNN-based acoustic model. The DLSTM system consists of a LSTM-based duration model and a LSTM-based acoustic model. The LSTM-based acoustic model consists of 3 unidirectional LSTM layers, each has 256 hidden units, followed by a linear output layer, which outputs 127-dimensional acoustic features, including 39-dimensional mel-cepstral coefficients (MCEPs) plus log energy, 1-dimensional band-aperiodicity parameter (BAP), logarithmic fundamental frequency (LF0), their first and second order deltas, and voice/unvoiced (V/UV) decision. The end-to-end system is an improved Tacotron model [6, 15, 16] with some modifications - we use 5 convolutional layers followed by a unidirectional GRU layer as the post-net. The GRU layer is intended to smoothen the generated mel spectrograms since the model predicts 3 frames at each decoding step. We also use a parallel WaveNet model, which is trained on ground-truth mel spectrograms for simplicity, as the neural vocoder of our improved Tacotron model. Because we only focus on the acoustic model, the aforementioned DNN-based duration model is also used in our QFCVI system. The same front-end model is shared for all four systems.

A corpus in standard Chinese from a *male* speaker, which contains about 17 hours of 16kHz speech data, and a corpus in standard Chinese from a *female* speaker, which contains about 50 hours of 16kHz speech data, are used for the subjective listening tests. We use a 50 ms frame size, 5 ms frame hop for the QFCVI and 12.5 ms frame hop for the improved Tacotron, and a Hann window function to extract 80-band log-scale mel-frequency spectrograms. Before taking the log compression, the mel spectrograms are stabilized to a floor of $1e - 5$. We use min-max normalization across each band of the mel spectrograms. The range of the normalized mel spectrograms is limited to $[-4, 4]$. Since we evaluate the Tacotron model on Chinese, input sequences consisted of pure phonemes are not a good choice. We also feed *word segmentation* information and *punctuations* to the Tacotron model.

3.2. Training Setup

The QFCVI model is trained for 400,000 global steps with the Adam optimizer [17] with a minibatch size of 2 utterances and a learning rate of 0.001, The gradients are clipped by a global norm of 10. The parallel WaveNet vocoder is trained for 300,000 global steps with the Adam optimizer with a minibatch size of 2 audio clips, each containing 8,000 timesteps. The parallel WaveNet is trained using the *Probability Density Distillation loss* and the *Power loss* [12]. Other training hyperparameters are the same as described in [18].

3.3. Subjective Evaluation

10 sentences are used to evaluate all four systems. Each sample generated by those systems is rated by 34 listeners in terms of *naturalness* on a scale from 1 to 5 with 1 point increament.[2]

Name	MOS (male)	MOS(female)
DNN	3.453±0.113	3.453±0.112
DLSTM	3.809±0.109	3.873±0.092
Imp. Tacotron	3.956±0.121	3.373±0.126
QFCVI	**3.926±0.099**	**4.020±0.096**

Table 1. 5-scale mean opinion score (MOS) evaluation in terms of naturalness with 95% confidence intervals.

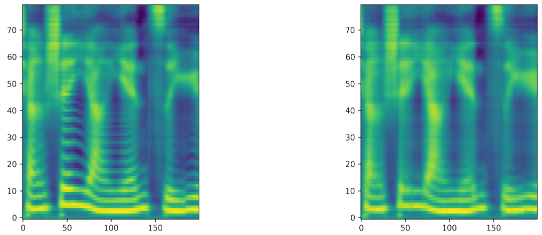
Tabel 1 shows the MOS results of 4 systems. Our proposed model QFCVI achieves a MOS of 3.926±0.099 for the male speaker, and a MOS of 4.020±0.096 for the female speaker, which surpasses the scores received by the DNN and LSTM systems. The improved Tacotron model gets a MOS of 3.956±0.121 for the male speaker, slightly higher than the score of our QFCVI model, and a MOS of

[2] Samples are available at https://mu94w.github.io/QFCVI/ .

3.373±0.126 for the female speaker, even worse than the DNN system. By analyzing the generated samples case by case, we found that the prosody prediction of the Tacotron model was suffered from occasional wrong word segmentation. And for the female speaker, although the Tacotron system can generate high-quality speech audios, which thanks to the parallel WaveNet vocoder, the speech rate was often too fast.

3.4. Ablation Studies by Case Analysis

3.4.1. Variational Auto-Encoder Architecture

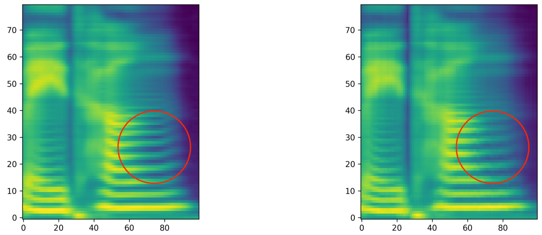


(a) With variational encoder. (b) Without variational encoder.

Fig. 4. Case study for variational auto-encoder architecture.

To show the importance of variational auto-encoder architecture, we trained a pair of models with and without a variational encoder. As shown in Fig.4, the mel spectrogram generated by the model without variational inference is much more blurred than the one with variational inference. Moreover, we found that the prosody of generated speech by our proposed system can be a little different even fed with the same sentence. And if we fed all zeros as the latent vector to the decoder, the generated speech would have poor prosody. So we conjecture that the latent vectors in our proposed model captured some prosody information.

3.4.2. QRNN



(a) With QRNN layer. (b) Without QRNN layer.

Fig. 5. Case study for QRNN layer in the Output Block of the decoder. Red circles are used to mark the differences between two mel spectrograms.

To show the importance of QRNN, we trained a pair of models with and without a QRNN layer in the Output Block of the decoder. As shown in Fig.5, the mel spectrogram generated by the model with QRNN in the decoder is smoother, more continuous and less trembling than the one without QRNN. Although generic RNNs can also tackle that problem, we choose QRNN for its high computational efficiency.

3.5. Further Speedup Parallel WaveNet

Although the parallel WaveNet vocoder is quite fast running on GPUs, we explored the possibilities for further speedup. We show two methods here.

Mixed Precision The parallel WaveNet vocoder with mixed precision is compressed from a trained one with single precision. All calculations and activations are in *16-bit floating point*, except the output layer of each inverse auto-regressive (IAF) flow.

Softsign We replace both the tanh and sigmoid in all gating layers of the WaveNet model with the softsign function, which is efficient for mobile CPUs [19] and can also offer speedup for GPUs.

The parallel WaveNet models are trained on the corpora aforementioned at section 3.1. 20 sentences are randomly selected from the test set as the evaluation set. We use the ground-truth mel spectrograms to generate speech audios. Each sample is rated by more than 10 listeners.

Name	MOS (male)	MOS (female)
Single Precision	3.985±0.081	4.176±0.099
Mixed Precision	4.023±0.079	4.161±0.098
Softsign	3.823±0.101	-
Natural	4.169±0.071	4.176±0.099

Table 2. 5-scale mean opinion score (MOS) evaluation in terms of quality with 95% confidence intervals.

The MOS results are shown in Table 2. Although the softsign approach receives a lower MOS, the quality of its generated speech is acceptably good. Due to the efficiency running on mobile CPUs and speedup for GPUs, such performance degradation is actually acceptable for mobile applications. Interestingly, the mixed-precision model for the male speaker achieves a slightly higher score than the single-precision one, although it is somewhat counterintuitive, which indicates that a lower precision, such as 8-bit integer, is worth exploring.

4. CONCLUSION

In this paper, we propose a quasi-fully convolutional neural networks with variational inference (QFCVI) model to generate mel spectrograms from given linguistic features. A parallel WaveNet model is used as vocoder to convert mel spectrograms to waveforms. Results of subjective listening tests show that our model can generate speech with a high degree of naturalness even on a par with an end-to-end model. Comparing to generate waveforms directly from linguistic [12], our model can be trained with much less computational resources. Moreover, we found that the mixed precision approach is a feasible method for further speedup. For future work, we plan to analyze the latent space of our proposed model for better prosody generation and control.

5. ACKNOWLEDGEMENT

This work is supported by joint research fund of National Natural Science Foundation of China Research Grant Council of Hong Kong (NSFC-RGC) (61531166002, N CUHK40415), National Natural Science Foundation of China (61433018, 61375027) and National Social Science Foundation of China (13&ZD189). We would also like to thank Tencent AI Lab Rhino-Bird Joint Research Program (No.JR201803) and Tsinghua University Tencent Joint Laboratory for the support.

6. REFERENCES

[1] Heiga Zen, Keiichi Tokuda, and Alan W Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W Black, and Keiichi Tokuda, "The hmm-based speech synthesis system (hts) version 2.0.," in *SSW*. Citeseer, 2007, pp. 294–299.

[3] Zhen-Hua Ling, Shi-Yin Kang, Heiga Zen, Andrew Senior, Mike Schuster, Xiao-Jun Qian, Helen M Meng, and Li Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[4] Heiga Ze, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7962–7966.

[5] Heiga Zen and Haşim Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4470–4474.

[6] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proceedings of Interspeech*, Aug. 2017.

[7] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[8] Shaojie Bai, J Zico Kolter, and Vladlen Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[9] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," in *Proceedings of the Neural Information Processing Systems (NIPS)*, 2013.

[10] Carl Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[11] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.

[12] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis, "Parallel wavenet: Fast high-fidelity speech synthesis," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[13] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher, "Quasi-Recurrent Neural Networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[14] Tim Salimans and Diederik P Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proceedings of the Neural Information Processing Systems (NIPS)*, 2016, pp. 901–909.

[15] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning (ICML)*, 2018.

[16] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J Weiss, Rob Clark, and Rif A Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *International Conference on Machine Learning (ICML)*, 2018.

[17] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] Mu Wang, Zhiyong Wu, Shiyin Kang, Xixin Wu, Jia Jia, Dan Su, Dong Yu, and Helen Meng, "Speech super resolution using parallel wavenet," in *Proceedings of the Chinese Spoken Language Processing (ISCSLP)*, 2018.

[19] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu, "Efficient neural audio synthesis," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.