FeatherWave: An efficient high-fidelity neural vocoder with multi-band linear prediction

Qiao Tian, Zewang Zhang, Heng Lu, Ling-Hui Chen, Shan Liu

Tencent, China

{briantian, zewangzhang, bearlu, nedchen, shiningliu}@tencent.com

Abstract

In this paper, we propose the FeatherWave, yet another variant of WaveRNN vocoder combining the multi-band signal processing and the linear predictive coding. The LPCNet, a recently proposed neural vocoder which utilized the linear predictive characteristic of speech signal in the WaveRNN architecture, can generate high quality speech with a speed faster than real-time on a single CPU core. However, LPCNet is still not efficient enough for online speech generation tasks. To address this issue, we adopt the multi-band linear predictive coding for WaveRNN vocoder. The multi-band method enables the model to generate several speech samples in parallel at one step. Therefore, it can significantly improve the efficiency of speech synthesis. The proposed model with 4 sub-bands needs less than 1.6 GFLOPS for speech generation. In our experiments, it can generate 24 kHz high-fidelity audio 9x faster than real-time on a single CPU, which is much faster than the LPCNet vocoder. Furthermore, our subjective listening test shows that the FeatherWave can generate speech with better quality than LPCNet.

Index Terms: WaveRNN, LPCNet, multi-band, linear prediction

1. Introduction

In recent years, the quality of text-to-speech (TTS) has been significantly improved by neural vocoders such as WaveNet [1], Parallel WaveNet [2], WaveRNN [3], LPCNet [4], etc. These neural vocoders are usually used in sequence-to-sequence acoustic models, e.g. Tacotron 2 [5] and DurIAN [6], to achieve generating human-like speech. The WaveNet vocoder, which is the state of the art model, can generate high-fidelity audio but is hard to deploy for real time services because of its huge computational complexity. The flow based neural vocoders, such as Parallel WaveNet [2], Clarinet [7], WaveGlow [8], are more practicable since they can perform parallel generation on GPU devices. However, these models often suffer from phase issues since the causality prior is ignored. Therefore the generated speech usually sounds muffled compared with the original auto-regressive WaveNet. Generative Adversarial Network (GAN) [9] has been adopted to address these issues in Parallel WaveNet [10, 11].

Recently, efficient RNN based sequential neural vocoders, such as WaveRNN, LPCNet and Multi-band WaveRNN [6], have been proposed for improving the performance of neural TTS system. The proposed LPCNet is the most lightweight neural vocoder currently, which integrates WaveRNN structured neural synthesis techniques with linear prediction. Meanwhile, an improved sampling strategy, as well as the pre-emphasis prior to μ-law quantization is introduced for achieving good quality under a small model size. Different from separately predicting the coarse and fine parts of the discretized speech signal in WaveRNN, LPCNet replaces the dual softmax output layer with a single softmax output layer on the 8-bit μ-law quantized signal with pre-emphasis. As a result, the LPCNet can produce 16 kHz high quality speech with a complexity less than 3 GFLOPS, which significantly improved the speed of speech synthesis system. On the other hand, the Multi-band WaveRNN is a variant of WaveRNN, which integrates multi-band strategy into WaveRNN based neural vocoder. Compared with WaveRNN, Multi-band WaveRNN can produce multi samples at one sequential step in parallel.

However, neural TTS systems with low computational complexity are very important for practical applications. As reported in [4] and [6], both LPCNet and Multi-band WaveRNN can not be 5x faster than real-time when producing 24 kHz high quality speech with a single CPU core, which means that the latency of synthesizing one second speech could be more than 200ms. Furthermore, there are many applications that require synthesizing speech on edge-devices, such as mobile phones with very limited computational capacity. For this purpose, we propose the FeatherWave vocoder, which merges multi-band processs to LPCNet framework. This makes it possible to match the quality of the state of the art neural vocoder WaveNet with significantly smaller computational load.

The contributions of this paper are summarized as follows: (1) We propose the multi-band (MB) linear prediction (LP) based FeatherWave vocoder. Firstly, we adopt the multi-band signal processing into the LPCNet framework. Then, we combine the μ-law quantization with MB-LP for efficiently modeling the discretized speech signal. Benefiting from the MB-LP process, the complexity of the proposed model is significantly reduced compared with the conventional LPCNet. (2) We demonstrate that the proposed FeatherWave can be 10x faster than real-time on two CPU cores by using our engineered streaming inference kernel when generating 24 kHz high-fidelity speech, which achieved a mean opinion score (MOS) of 4.55 in our subjective listening test.

We organize the rest of the paper as follows: in Section 2, we will briefly review the lightweight RNN based neural vocoder, such as Multi-band WaveRNN and LPCNet. Then the proposed method will be given in Section 3. The evaluation of results will be presented in Section 4. Lastly in Section 5, conclusions and future work are presented.

2. Related work

2.1. Multi-band WaveRNN

Compared with Subscale WaveRNN [3], which can generate multi samples per step with a subscale dependency scheme, Multi-band WaveRNN exploits multi-band generation strategy

Preprint. Work in progress.

with the technique of subband [12, 13] to improve generation speed. It predicts all subband signal simultaneously through a multiple softmax output layer in a single recurrent step in WaveRNN. By using this variant of WaveRNN, the length of generated sequence can be down-sampled by a factor of N_b (the number of frequency bands). As a result, the total computational cost can be reduced to approximately 3.6 GFLOPS [6]. Before model training, the original waveform signal $x = \{x_1, \ldots, x_T\}$ should be down-sampled by N_b invertiable analysis filters into N_b subbands waveforms $g = \{g^b\}, b = 1, \ldots, N_b$, where $g^b = \{g^b_1, \ldots, g^b_{T/N_b}\}$. The joint probability of multi-band signal can be factorised as a product of conditional probabilities of subband signals as described as

$$p(g) = \prod_{n=1}^{T/N_b} p(g_n | g_1, g_2, \ldots, g_{n-1}), \tag{1}$$

where the conditional probability can be modeled by a recurrent neural network (RNN).

2.2. LPCNet

The LPCNet makes effort to reduce the computational load of each generation step benefiting from the classical technique of linear prediction. Similar to GlotNet [14] and ExcitNet [15] which use the WaveNet to capture the glottal excitation signal, the LPCNet models the discretized excitation signal of LPC filters with a WaveRNN for efficient generation. Instead of open-loop filtering approaches [16], LPCNet preforms as a closed-loop synthesis of predicting sample x_t by conditioning on the previously sampled excitation e_{t-1} and current prediction p_t, which can improve the quality of generated speech.

3. The proposed method

In this section, we present the proposed variant of WaveRNN vocoder, FeatherWave, which further improves the speed of audio generation with multi-band process and maintains the advantages of the LP-structure as LPCNet. Firstly, we introduce the MB-LP framework, which extends the process of the LP coding to multi-band signal. Then, we propose the FeatherWave vocoder which applies the MB-LP framework into the conventional neural vocoder.

3.1. Multi-band Linear Prediction

For the purpose of utilizing linear prediction to obtain good quality and multi-band to speed up synthesis, we introduce multi-band linear prediction (MB-LP) in the proposed model. By adopting LP analysis on multi-band waveform signal, M order linear prediction coefficients of each sub frequency band, α^b_k, can be extracted from the corresponding frequency bins of mel-spectrogram frame. The b-th subband signal g^b is down-sampled from the original signal x by invertible analysis filters. Under the LP assumption, the corresponding predicted signal p^b_n and excitation (prediction residual) e^b_n of b-th band can be computed as follows:

$$p^b_n = \sum_{k=1}^{M} \alpha^b_k \, g^b_{n-k}, \tag{2}$$

$$g^b_n = p^b_n + e^b_n. \tag{3}$$

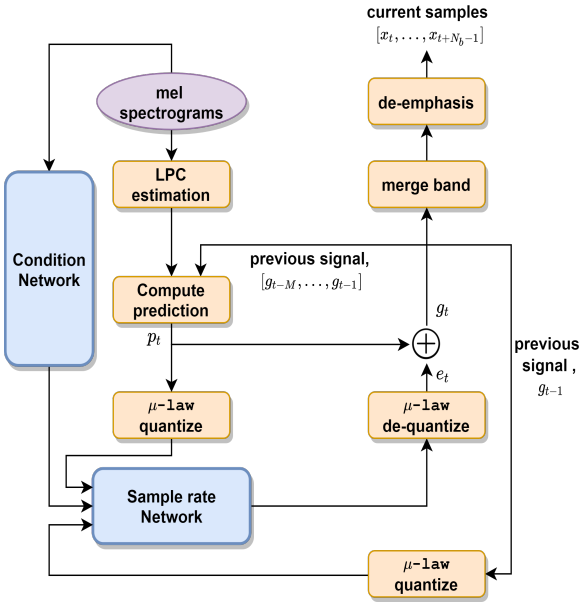

Figure 1: *Block diagram of the proposed FeatherWave vocoder.*

3.2. FeatherWave

In our proposed FeatherWave vocoder, MB-LP is introduced into the conventional WaveRNN vocoder as illustrated in Fig. 1. It consists of a condition network that operates on input frames of mel spectrograms and a sample rate network which produces N_b samples with a multi dual softmax output layer. Similar to the original WaveRNN, the sampling network firstly predicts coarse part of excitation signal and then computes fine part by conditioning on the predicted coarse signal. As indicated in Eq. 3, the subband signal is predicted from the network output excitation and linear predicted signal, which is linearly predicted from previous output signal as show in Eq. 2. As illustrated in Fig. 1, the merge band operation is applied, by using synthesis filters, to reconstruct original waveform signal from the predicted signal of subbands. In this paper, only mel spectrograms, which are widely used in neural TTS systems, are adopted as input conditional features.

3.2.1. Discretized Multi-band Linear Prediction

In LPCNet, a first-order pre-emphasis filter $E(z) = 1 - \alpha z^{-1}$ is applied to training data. This pre-emphasis makes it possible to model 8-bit μ-law discretized signal with high quality.

As an obvious extension of using this technique to help model learn and generate more efficiently, we also apply this pre-emphasis filter to training signal firstly, and then μ-law quantize all subbands signals after MB-LP process. Similar to LPCNet, we can model μ-law discretized signal using smaller model and achieve high-fidelity synthesis with the proposed MB-LP framework. For trading off quality against model size, we adopt 10-bit μ-law quantization for each subband signal in the FeatherWave.

3.2.2. Condition Network

For neural vocoder, the intelligibility of generated speech is much sensitive to the structure of condition network. In FeatherWave, instead of using bi-directional RNN, we adopted a stack

of convolutional layers as the condition network for the purpose of streaming inference. Specifically, the local acoustic features are firstly operated by five 1×3 convolution layers so the sample rate network can obtain enough receptive field. We adopt exponential linear unit (ELU) activation after every convolutional layer for more stable training. In order to match the sampling rate of target signal, the outputs of condition network are simply repeated by f times before passed into sample rate network. As h denotes hop size, the number of repetitions is $f = h/N_b$.

3.2.3. Sample Rate Network

In the sample rate network, predictions computed from linear prediction are conditioned for the manner of closed-loop synthesis by following the method in LPCNet. As a result, the predictions perform as reference signal to compute excitations. This can enhance the performance of model. Besides, the upsampled features from the output of condition network and the previous generated signal are used as well. All discretized signals are passed into a trainable embedding layer before fed into a GRU cell. Similar to the WaveRNN vocoder, we use dual softmax layer to predict coarse and fine parts of the discretized signal sequentially after a GRU and affine layers. A block sparse pruning [17] strategy is adopted to sparsify the parameters in the GRU layer for the purpose of speeding up inference. The output of the affine layer is passed into multiple softmax output layers to predict all subband excitations simultaneously. The parameters of model are optimized to minimize the negative log-likelihood (NLL) loss at the training phase.

3.2.4. Generation Method

In typical lightweight neural vocoder where small model is adopted, it is necessary to adjust the sharpness of the output distributions to avoid noise caused by the random sampling process and achieve better quality. In FFTNet [18] and iLPCNet [19], lowering temperature in the voiced region with a constant factor is exploited for such purpose. Rather than using voiced information, LPCNet adopts pitch correlation to adjust the temperature factor. Furthermore, the distribution is subtracted with a constant threshold T to prevent impulse noise caused by low probabilities.

Since only mel-spectrograms are used in condition network, we explore the technique of distribution subtraction carefully for better performance. We observed that a temperature $T = 0.02$ produced good results in the trade-off quality against artifact in generated speech. The subtraction is only performed on the distribution of fine part, which is given as follow:

$$P_f^{'}(e_t) = \mathcal{R}(\max[P_f(e_t) - T, 0]), \tag{4}$$

where $P_f(e_t)$ denotes the distribution of fine part, and $\mathcal{R}(\cdot)$ denotes the normalizing operator.

3.3. Two-stage Sparse Pruning

In [3], a GRU with block sparse weights is vital for achieving fast inference in neural vocoders. In this work, in order to improve the performance of block sparse strategy, we apply a novel two-stage sparse pruning (TSSP) method to achieve high sparsity ratio in GRU weights.

In the conventional block sparsity pruning methods, a high sparsity ratio (above 40%) usually degrades the model performance as mentioned in [20]. In practice, high sparsity ratio usually hurts the speech quality of neural vocoders, although it could speed up the inference. To address this problem, we adopt a two-stage sparse pruning strategy, which consists of warming-up stage and increasing stage. Firstly, we train sparse model with a warming-up sparsity ratio which is 50% in our configuration to avoid hurting performance of model in warming-up stage. In the increasing stage, we increase the sparsity ratio progressively by loops to reach the target sparsity ratio, e.g. increasing 10% sparsity ratio in a loop. We maintain the sparsity ratio with a constant iterations after the warming-up sparsity ratio or the target sparsity ratio of every loop in increasing stage is reached.

4. Experiments

4.1. Data Set

In our experiments, we used a Mandarin corpus of 20 hours of recordings, which were recorded by a professional broadcaster. The data we split into a training set and a test set. About 18 hours of recordings were used for model training and the rest were used for testing. All the recordings were down-sampled to 24 kHz sampling rate with 16-bit format. The 80 order mel-spectrograms were extracted as the conditions for all neural vocoders in our experiments with the method mentioned in [5].

4.2. Experimental Setup

To demonstrate that the proposed model accelerates speech synthesis without degrading the speech quality, we chose LPCNet, which is open-sourced and is known as the fastest high-quality neural vocoder, as the baseline. In the LPCNet baseline system, we used the open-sourced implementation [1] based on the commit $3a7ef33$ and the configuration was exactly the same as its original version. A 384-dimensional GRU layer with 90% sparsity ratio before the 16-dimensional dense GRU layer was used. Since the LPCNet open-sourced implementation can only generates 16 kHz audio, we down-sampled the generated speech of the proposed model for a fair comparison. The original 24 kHz speech of our model is included in the comparison as well. In order to observe the gap between the proposed model and the state-of-the-art neural vocoder, a WaveNet with mixture of logistic (MoL) output layer was also adopted for comparison. For robustness and stability, we chose the MoL WaveNet variant [21] and all the configurations were the same as mentioned in [21].

In the proposed FeatherWave vocoder, conv1d layers with kernel size 1×3 and channel size 256 were used in the condition network. In sample rate network, the final sparsity ratio is set as 90% in the GRU with 384 hidden units. The dimension of affine layer is 128. The embedding size for discretized signal is 16. In this work, we used 4 bands and 10-bit μ-law quantization for dual softmax layers, therefore the output dimension of the last FC layer before softmax layer was 128. For modeling and reconstructing on subband signal, we followed the design of analysis filters and synthesis filters in [22]. Instead of adopting cepstrums [4], the LP coefficients were estimated from the mel-spectrograms as in [23].

In the training phase, the Adam [24] optimizer was adopted with a learning rate of 0.001. The proposed model was trained on a single GPU with mini-batch size of 1536 samples. The weights of the neural vocoders were randomly initialized with fixed random seed and all the networks were trained with 1200k iterations. In the two-stage sparse pruning of FeatherWave, the target sparsity ratio of the warming-up stage was 50% with

[1] `https://github.com/mozilla/LPCNet/`

Table 1: *The synthesis speed over real-time of the baseline model LPCNet and the proposed FeatherWave for two sampling rate (*16 *kHz and* 24 *kHz) speech.*

syn. speed	single core	two cores
LPCNet	5.7x	-
FeatherWave (16k)	**12.1x**	**15.5x**
FeatherWave (24k)	**9.2x**	**10.8x**

300k sparse iterations and continued to the increasing stage after maintaining the current sparsity with 100k iterations. In every loop of increasing stage, the sparsity ratio was increased by 10% with 100k iterations and maintaining the current sparsity with another 100k iterations. After four loops in increasing stage, the total iterations reached 1200k and the final sparsity ratio was 90%, which is same as in the LPCNet. The blocks with size 16×1 were adopted in our pruning experiments.

4.3. Synthesis Speed

We estimated the computational complexity of different vocoders firstly for revealing the speedup of our proposed FeatherWave vocoder. The main complexity of FeatherWave comes from one sparse GRU and four fully-connected layers. We compute it following the method in [4], which is given by:

$$C = (3dN_G^2 + N_G \cdot N_F + 2N_F \cdot Q \cdot N_B) \cdot 2F_S/N_B, \quad (5)$$

where N_G is the size of the sparse GRU, d is the density of the sparse GRU, Q is the root of the number of μ-law levels, N_F is the width of affine layer connected with final fully-connected layer, N_B is the number of frequency bands, and F_S is the sampling rate. In our experiments, we set N_G = 384, d = 0.1, Q = 32, N_F = 128 and N_B = 4 for F_s = 16000. Therefore, a total complexity of FeatherWave is approximately 1.6 GFLOPS, which is much smaller than 2.8 GFLOPS in the conventional LPCNet.

The synthesis speeds over real-time of different vocoders are listed in Table 1. All the speed tests were performed on the Intel Xeon Platinum 8255C CPU. The results show that merging multi-band into LPCNet framework can bring about 2x speedup when generating 16 kHz speech. When producing high-fidelity 24 kHz speech, FeatherWave can be 10x faster than real-time using our engineered multi-thread inference kernel on two CPU cores. Additionally, our implementation of Parallel WaveNet [10] requires 8 cores to achieve the similar synthesis speed.

4.4. Evaluations

Firstly, subjective evaluation was conducted to evaluate the MOS of perceptual quality of the proposed FeatherWave vocoder. In order to perform fair comparison, we randomly selected 40 utterance from test set for MOS testing and 30 native Mandarin speakers participated in the listening test.

The results[2] of the subjective MOS evaluation is presented in Table 2. The results show that the proposed FeatherWave can generate high quality 16 kHz speech with a slightly better MOS than the LPCNet. And when producing high-fidelity speech at higher sampling rate (24 kHz), the proposed FeatherWave achieves a MOS with a small gap to the powerful MoL WaveNet, which consists of 24 dilated conv1d layers. Since

[2]A subset of generated samples can be found at the following URL: `https://wavecoder.github.io/FeatherWave/`

Table 2: *Mean Opinion Score (MOS) with* 95% *confidence intervals for different vocoders.*

Model	MOS on speech quality
LPCNet	4.48 ± 0.04
FeatherWave (16k)	4.51 ± 0.03
FeatherWave (24k)	$\mathbf{4.55 \pm 0.03}$
MoL WaveNet	4.58 ± 0.02

Table 3: *FeatherWave NLL results on different sparse strategies. All the experiments were conducted on the same sparsity ratio, 90%.*

Method	NLL
FeatherWave **w/o TSSP**	4.14
FeatherWave w/ TSSP	**4.07**

we use mel-spectrograms to extract the LP filters, the proposed model doesn't depend on pitch extraction. The model has less artifact in the generated speech and is easy to build a neural TTS system instead of LPCNet. Furthermore, our model can produce less quantization noise and fidelity loss than LPCNet as 10-bit μ-law quantization with dual softmax layer is used instead of 8-bit one.

We also investigated the effectiveness of two-stage sparse pruning method by objective NLL results. Lower NLL usually indicates better quality of the neural vocoder generated speech [3]. It is obviously observed from the results in Table 3 that the model got lower NLL compared with the baseline model after using the proposed two-stage sparse pruning method, which could lower the probability of bad choice in sparse pruning compared with the conventional pruning methods. Finally, we got better speech quality in FeatherWave with this improvement.

5. Conclusions and future work

In this work, we proposed the FeatherWave vocoder which applies the MB-LP method to the conventional RNN based neural vocoder, such as WaveRNN. For faster generation and utilizing the linearity of the LP filters, we merged multi-band into LPCNet framework which only conditioned on mel spectrograms. Furthermore, we also make other contributions, such as the discretized multi-band linear prediction and two-stage sparse pruning. Our experimental results indicated that the proposed FeatherWave can further reduce the computational cost at speech generation and get higher speech quality compared with the conventional neural vocoders.

In future work, we will investigate FeatherWave with low bit and balanced sparsity [20] pruning training method for deploying on edge-devices.

6. Acknowledgments

The authors would like to thank Yi Xie and Ciyong Chen in IAGS, Intel Asia-Pacific Research & Development Co Ltd.. These two members in Intel not only provided the guidance on how to get good performance on the Intel(R) Xeon(R) Scalable Processors, but also helped to optimize/validate our algorithm with Intel(R) Deep Learning Boost using bfloat16 (BF16) format on the upcoming hardware.

7. References

[1] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio." in *SSW*, 2016, p. 125.

[2] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[3] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[4] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

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

[6] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei *et al.*, "Durian: Duration informed attention network for multimodal synthesis," *arXiv preprint arXiv:1909.01700*, 2019.

[7] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

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

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[10] Q. Tian, X. Wan, and S. Liu, "Generative adversarial network based speaker adaptation for high fidelity wavenet vocoder," in *Proc. 10th ISCA Speech Synthesis Workshop*, pp. 19–23.

[11] R. Yamamoto, E. Song, and J.-M. Kim, "Probability density distillation with generative adversarial networks for high-quality parallel waveform generation," *arXiv preprint arXiv:1904.04472*, 2019.

[12] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, "Improving fftnet vocoder with noise shaping and subband approaches," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 304–311.

[13] T. Okamoto, K. Tachibana, T. Toda, Y. Shiga, and H. Kawai, "An investigation of subband wavenet vocoder covering entire audible frequency range with limited acoustic features," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5654–5658.

[14] L. Juvela, B. Bollepalli, V. Tsiaras, and P. Alku, "Glotneta raw waveform model for the glottal excitation in statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 1019–1030, 2019.

[15] E. Song, K. Byun, and H.-G. Kang, "Excitnet vocoder: A neural excitation model for parametric speech synthesis systems," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[16] L. Juvela, V. Tsiaras, B. Bollepalli, M. Airaksinen, J. Yamagishi, and P. Alku, "Speaker-independent raw waveform model for glottal excitation," *arXiv preprint arXiv:1804.09593*, 2018.

[17] S. Narang, E. Undersander, and G. Diamos, "Block-sparse recurrent neural networks," *arXiv preprint arXiv:1711.02782*, 2017.

[18] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "Fftnet: A real-time speaker-dependent neural vocoder," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2251–2255.

[19] M.-J. Hwang, E. Song, R. Yamamoto, F. Soong, and H.-G. Kang, "Improving lpcnet-based text-to-speech with linear prediction-structured mixture density network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7219–7223.

[20] Z. Yao, S. Cao, W. Xiao, C. Zhang, and L. Nie, "Balanced sparsity for efficient dnn inference on gpu," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5676–5683.

[21] Q. Tian, J. Chen, and S. Liu, "The tencent speech synthesis system for blizzard challenge 2019."

[22] T. Q. Nguyen, "Near-perfect-reconstruction pseudo-qmf banks," *IEEE Transactions on signal processing*, vol. 42, no. 1, pp. 65–76, 1994.

[23] R. Korostik, A. Chirkovskiy, A. Svischev, I. Kalinovskiy, and A. Talanov, "The stc text-to-speech system for blizzard challenge 2019."

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.