



Joint Modeling of Multi-Sample and Subband Signals for Fast Neural Vocoding on CPU

Hiroki Kanagawa, Yusuke Ijima, Hiroyuki Toda

NTT Corporation, Japan

hiroki.kanagawa.wk@hco.ntt.co.jp

Abstract

In this work, we propose a fast and high quality neural vocoder for CPU implementation. The main approaches to realize fast inference via an autoregressive model are 1) a subband-based vocoder and 2) multiple samples prediction. Our previous work demonstrated that the combination worked well up to two samples simultaneous generation without quality degradation. To further increase the number of simultaneous samples while maintaining quality, we focus on the existence of an association between subband signals and multiple samples. Our proposed vocoder jointly models these associations with a multivariate Gaussian. Experimentals show that our proposed four-sample vocoder is 1.47 times faster than the conventional two-sample equivalent. For both the acoustic features extracted from natural speech and those predicted by TTS, the proposed method realizes generation with up to four samples without any significant degradation in naturalness. This vocoder also matched the naturalness comparable of the two-sample conventional method.

Index Terms: speech synthesis, neural vocoder, multi-sample subband WaveRNN, joint modeling

1. Introduction

The introduction of neural vocoders has greatly improved the quality of text-to-speech (TTS) services. While WaveNet can generate high-quality speech waveforms directly from conditioning features via a large-scale convolution-based autoregressive model, its heavy computational cost and autoregressive (AR) architecture lead to very slow inference [1]. For faster neural vocoding, many approaches have been proposed such as parallel computation using a non-AR model [2, 3, 4] and a lightweight model [5, 6, 7, 8]. As an example of the former approach, Parallel WaveNet is trained via a distillation scheme using a teacher WaveNet. Recent studies also utilize a generative adversarial network [9] to obtain the true waveform distribution to compensate the quality degradations created by non-AR models [10, 11, 12]. Although these models are fast, parallel processors such as GPUs are typically required for fast inferring. The latter approach is mainly aimed at high-speed inference on CPU by reducing DNN's computational complexity. WaveRNN replaces WaveNet's huge convolution with an RNN and offers real-time operation on CPUs [5]. LPCNet introduced signal processing knowledge to match the quality of WaveRNN but with smaller model size [6]. As a method to develop vocoder targets, [13] generates two LPCNet excitation signals simultaneously, yielding 1.5 times higher speeds than the original LPCNet. Predicting subband signals obtained by pseudo quadrature mirror filter (PQMF) [14] can reduce the sequence length to be predicted. [15] and [16] adopted this concept in WaveRNN and LPCNet, respectively, and more than doubled the speed while maintaining quality.

Inspired by [13] for further speedup, we proposed subband

WaveRNN-based multiple sample generation via multivariate Gaussian [17]. Our vocoder achieved 1.81 times greater speed with no quality degradation when up to two samples were generated simultaneously. Especially efficient was the joint modeling of the association among subband signals. However, the quality degradation became more significant as the number of simultaneously generated samples was increased. This is due to failure to take account of the associations between neighboring samples; this resulted in poor reproducibility of mid- and high-spectrum components. Since reproducibility of these bands depends primarily on the accuracy of the variance predicted by the vocoder, it is necessary to model the relationships among simultaneously generated samples. A related study, [18], simultaneously models the relationship between samples in WaveNet with a Laplace distribution. Although this allowed more full-band signals to be processed simultaneously while maintaining naturalness, its effectiveness in predicting subband signals has not been confirmed.

In order to simultaneously generate more subband WaveRNNs while maintaining quality, we propose the joint modeling of subband signals and multiple samples. Our goal is to appropriately predict the continuity between samples and improve mid to high frequency reproducibility. To this end, multivariate Gaussian, which was employed in our previous work for joint modeling of subband signals, is also applied to elucidate associations among multiple subband signals. Assuming M as the number of simultaneous samples, speed comparisons show that the proposed vocoder with $M=4$ and 8 is 1.47 and 1.84 times faster, respectively, than the conventional subband WaveRNN with $M=2$. Our vocoder with $M=8$ did not achieve a dramatic speedup, because the dimension of multivariate Gaussian increases by the square order of the product of M and the number of subbands, B . A subjective evaluation demonstrates that, for the same number of samples, M , our vocoders offers better naturalness than the conventional method for both analytical synthesis and TTS tasks. Although the conventional method with $M=4$ suffers noticeable drops in quality, our vocoder with $M=4$ matched the naturalness of the conventional vocoder with $M=2$. Spectrograms confirm that our vocoder could avoid the problem with spectral reproducibility unlike the conventional method with $M > 2$.

2. Conventional multi-sample subband WaveRNN

Subband WaveRNN [15] reduces sequence length from T to T/B by predicting B -band subband signals instead of full band signals. Figure 1 overviews the extension of method to implement multiple sample generation [17]. The model consists of an encoder and a decoder, which are responsible for frame rate and sample rate, respectively. Encoders upsample frame-level acoustic features to corresponding samples. The decoder generates predictions of the next time $t + \tau \forall_m \in [1, M - 1]$ from

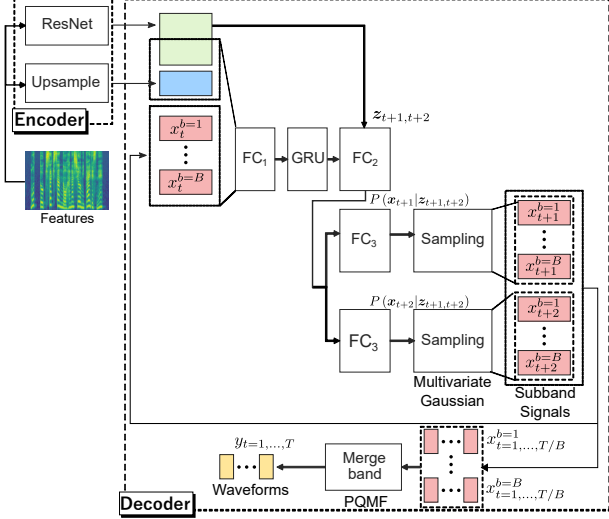


Figure 1: Overview of multi-sample subband WaveRNN via multivariate Gaussian [17]. This vocoder predicts M subband signals simultaneously in single forward propagation ($M=2$ is used in this figure).

the output of the encoder and the previous M subband signals, where m is the index of the number of subband signals to be generated simultaneously. To generate multiple samples in single forward propagation, linear layer FC3s are provided for subband signals of $t + \tau \forall m \in [1, M - 1]$. This module jointly predicts the associations among subband signals because PQMF has band overlaps. Assuming a multivariate Gaussian as FC3's target, this vocoder minimizes the negative log-likelihood given by:

$$\mathcal{L}(\theta) = - \sum_{t=1}^{T/B} \sum_{m=1}^M \ln \mathcal{N}(\mathbf{x}_{t+m}; \boldsymbol{\mu}(\mathbf{z}_{t+\tau \forall m}, \theta), \boldsymbol{\Sigma}(\mathbf{z}_{t+\tau \forall m}, \theta)) \quad (1)$$

where θ , \mathbf{z}_t , $\mathbf{x}_t \in \mathbb{R}^B$, $\boldsymbol{\mu} \in \mathbb{R}^B$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{B \times B}$ are the DNN model parameters, FC2's output, subband signals, the mean vector and covariance matrix of the multivariate Gaussian, respectively. In practice, to reduce the number of FC3's parameters, we predict \mathbf{L} instead of the covariance matrix $\boldsymbol{\Sigma}$.

$$\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top \quad (2)$$

where \mathbf{L} is a lower triangular matrix satisfying Cholesky decomposition¹. To guarantee spectral reproducibility, STFT loss [19] is calculated by generating subband signals from the multivariate Gaussian via a reparameterization trick. This is added to Eq. (1) without scaling to optimize the vocoder.

3. Proposed vocoder via joint modeling of multiple samples and subband signals

3.1. Model architecture and its training / vocoding

The conventional method described in the previous section can jointly model the association between subband signals. However, multiple samples are dealt with as being independent of each other, so continuity among samples is virtually lost completely when four or more samples are generated simultaneously. In particular, the mid- to high-frequency band, which has many random components, is hard to reproduce.

¹We took the logarithm of \mathbf{L} because each element is too small to model easily.

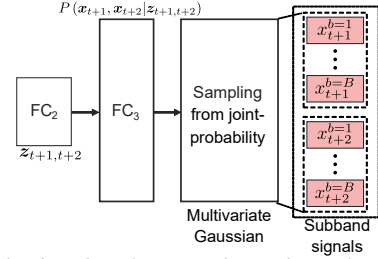


Figure 2: The decoder of proposed vocoder with joint modeling of multiple samples and subband signals. This vocoder aims to model the association between them.

To overcome this problem, we explicitly model the association between multiple samples and subband signals. Figure 2 shows the difference between the proposed decoder and the conventional equivalent (Fig. 1). Unlike the conventional vocoder, we extend the multivariate Gaussian's output by FC3 for not only subband signals but also multiple samples. Multiple subband signals are generated by single sampling operation from the joint probability $P(\mathbf{x}_{t+1}, \mathbf{x}_{t+2} | \mathbf{z}_{t+1,t+2}, \theta)$ obtained from FC3. The loss function during training is formulated by NLL as:

$$\mathcal{L}'(\theta) = - \sum_{t=1}^{T/BM} \ln \mathcal{N}(\mathbf{x}_{t+\tau \forall m}; \boldsymbol{\mu}'(\mathbf{z}_{t+\tau \forall m}, \theta), \boldsymbol{\Sigma}'(\mathbf{z}_{t+\tau \forall m}, \theta)) \quad (3)$$

where $\boldsymbol{\mu}' \in \mathbb{R}^{BM}$ and $\boldsymbol{\Sigma}' \in \mathbb{R}^{BM \times BM}$ are the mean vector and covariance matrix of multivariate Gaussian across multi-samples and subband signals, respectively. To pare parameters, FC3 predicts $\boldsymbol{\mu}'$ and a lower triangular matrix \mathbf{L}' that satisfies the Cholesky decomposition of $\boldsymbol{\Sigma}'$.

During vocoding, multiple subband signals are sampled from the multivariate Gaussian by:

$$[\mathbf{x}_t^\top, \dots, \mathbf{x}_{t+M-1}^\top]^\top = \boldsymbol{\mu}'(\mathbf{z}_{t+\tau \forall m}, \theta) + \mathbf{L}'(\mathbf{z}_{t+\tau \forall m}, \theta) \boldsymbol{\epsilon} \quad (4)$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^{BM}$ is a vector with variables that lie within $\mathcal{N}(0, 1)$. Note that, as in the conventional method, the elements of \mathbf{L}' are logarithmic except when sampling.

3.2. Computation complexity

The neural vocoder's computation complexity is mainly due to the decoder's DNN module. Its basic tasks are the operations of addition and multiplication, and floating-point operations (FLOPs) of conventional vocoder, see Section 2, as given by:

$$C = [d \{ (D_{in1} + D_{in2}) N_\alpha + 3N_\alpha^2 + (D_{in2} + N_\alpha) N_\beta \} + N_\beta \left(B + \frac{B}{2} (B + 1) \right) M] \times \frac{f_s}{BM} \quad (5)$$

where N_α and N_β are unit sizes of GRU and FC2, respectively. D_{in1} and D_{in2} are the Upsample's dimension, the half dimension of ResNet, and FC3's output dimension, respectively. d , f_s , and M are the density of the sparse DNN, the sampling frequency, and the number of simultaneous generated subband signals, respectively. The computational complexity of the vocoder proposed in Section 3.1 can likewise be reformulated as follows:

$$C' = [d \{ (D_{in1} + D_{in2}) N_\alpha + 3N_\alpha^2 + (D_{in2} + N_\alpha) N_\beta \} + N_\beta \left(BM + \frac{BM}{2} (BM + 1) \right)] \times \frac{f_s}{BM} \quad (6)$$

As described above, our method's FC3 incurs more than M times the computational cost of the conventional one. This increase in computational complexity yields only minor speed

Table 1: Average RTFs obtained from all evaluation data. “Speed enhancement” denotes the improvement over CONVENTIONAL ($M=2$).

Method	RTF	Speed Improvement
CONVENTIONAL ($M=2$)	0.094	-
PROPOSED ($M=2$)	0.096	0.98x
CONVENTIONAL ($M=4$)	0.055	1.71x
PROPOSED ($M=4$)	0.064	1.47x
PROPOSED ($M=8$)	0.051	1.84x

down for small M (e.g. $M=2,4$), whereas large value (e.g. $M > 4$) may create unacceptably low speeds. For example, using $d = 0.4$, $N_\alpha = 256$, $N_\beta = 128$, $B = 4$, $f_s = 22050$, $D_{in1} = 80$, $D_{in2} = 64$, and $M = 2$, Eqs. (5) and (6) are 0.312 and 0.317 GFLOPs, respectively. When $M=4$, they are estimated 0.161 and 0.176 GFLOPs, respectively (9.0% increase). We can see that the computational cost does not significantly increase even if multi-sample joint modeling is employed. By contrast, our vocoder at $M=8$ suffers a dramatic cost increase from 0.086 to 0.122 GFLOPs (41.9% increase).

4. Experiments

4.1. Setup

We used speech data uttered by a Japanese professional female speaker. The sampling frequency was 22.05 kHz. 200 utterances were extracted as evaluation data (18.3 minutes), and remainder were used for training and validation (30.6 hours).

Eighty-dimensional logarithmic mel-spectrograms were used as the conditioning feature of the neural vocoder. The analysis frame shift was 5 ms². The number of training steps was 5000k. For fast vocoding, we performed pruning [20] in the same manner as WaveRNN using:

$$d_s = d \left[1 - \{1 - (s - s_0) / S\}^3 \right], \quad (7)$$

where $s_0 = 2000k$, $S = 2500k$ and s is the training step’s index. In order to utilize vector operations by block sparsification [21], the pruning block sizes were set to four for FC1 and 16 for GRU and FC2. The parameters used in determining the computational complexity are those described in Section 3.2. The ResNet of the encoder has ten residual blocks, each consisting of 1D-convolution with 128 units, batch normalization, and activation. ReLU was used for all activations. The vocoder’s optimization was performed using RAdam [22], with $\alpha = 1.0 \times 10^{-4}$, $\beta = (0.9, 0.999)$, and $\varepsilon = 1.0 \times 10^{-8}$. All multi-sample vocoders often failed to predict accurate variance parameters which yielded clicking sounds. To avoid this problem, we 1) eliminate variance outliers and 2) clip sampled results in a similar way to [13].

4.2. Vocoding speed comparison

The real-time factors (RTFs) were calculated to measure the inference speeds for all methods. RTF is defined by:

$$\text{RTF} := T_{\text{inference}} / T_{\text{data}}, \quad (8)$$

²Although we also investigated the commonly used frame shift of 12.5 ms in our preliminary experiments, we chose to set it to 5 ms because it better reproduced the pitch of synthetic speech. If a faster inference speed is preferred, the frame shift can be set to 12.5 ms like other studies for lower encoder computational complexity.

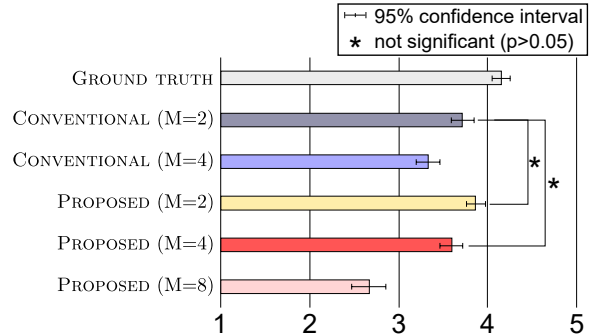


Figure 3: Mean opinion scores of naturalness. Acoustic features for vocoding were extracted from natural speech.

where T_{data} and $T_{\text{inference}}$ are speech length and single-thread inference time measured on an Intel Core i7-8750H CPU 2.20 GHz, respectively. Table 1 shows method, averaged RTFs from all evaluation data, and speed enhancement. A comparison of CONVENTIONAL ($M=2$) and PROPOSED ($M=2$) shows that joint modeling of multiple samples does not dramatically affect the RTF at $M=2$. This is a reasonable result as there is not no significant increase in decoder computational as discussed in Section 3.2. While vocoders for $M=4$ were faster than those for $M=2$, the speed enhancement was small due to joint modeling. This tendency was particularly noticeable for $M=8$, almost the same RTF (about 0.05) as that of PROPOSED ($M=4$). This was due to the more than M times increase in FC3’s computational complexity. Avoiding its pruning for the sake of quality also had an overly negative effect on inference speed.

Note that Kong *et al.*, 2020 claimed their RTF=0.075³ on an Intel Core i7 2.6 GHz CPU with the smallest HiFi-GAN, which is a fast non-AR model yet high quality [12]. Though direct comparisons are difficult because of different CPUs, our PROPOSED ($M \geq 4$) achieved extremely fast vocoding via AR-model even if a lower clock CPU than that of them were used.

4.3. Subjective evaluations

We subjectively evaluated naturalness of synthetic speech by using mean opinion score (MOS) on a five-point scale ranging from 5: very natural to 1: very unnatural. Sixty listeners participated in the test via crowdsourcing. They evaluated ten sentences for each method, randomly selected from all 200 evaluation data, for a total of sixty sentences.⁴

4.3.1. Vocoding for extracted acoustic features from natural speech

Figure 3 shows the subjective evaluation results of vocoding with acoustic features extracted from natural speech. CONVENTIONAL ($M=2$) and ($M=4$) showed scores of 3.72 and 3.33, respectively, revealing a drop in the naturalness with the increasing in M . This behavior is similar to that reported in our previous work [17]. Although no significant difference between CONVENTIONAL ($M=2$) and PROPOSED ($M=2$) was found, PROPOSED ($M=4$) obtained a naturalness of 3.59, which is higher than that of CONVENTIONAL ($M=4$). In particular, PROPOSED ($M=4$) matched the naturalness of CONVENTIONAL

³The smallest HiFi-GAN’s RTF is estimated 0.075 because it generates samples 13.4 times faster than real-time.

⁴These participants were different evaluators for analytic resynthesis and TTS.

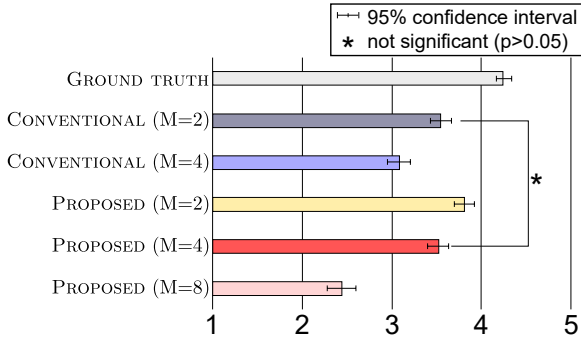


Figure 4: Mean opinion scores in terms of naturalness. Acoustic features for vocoding were predicted by TTS models.

(M=2). However, PROPOSED (M=8) showed a significant quality degradation. We discuss possible reasons for these results by observing the spectrograms of each method in Section 4.4.

4.3.2. Vocoding based on acoustic features predicted by TTS

To investigate robustness against degraded acoustic features, FastSpeech2 [23] as the TTS model was also trained with the same data as neural vocoders. We fed the 380 kinds of symbols including phoneme and prosodic information to FastSpeech2. It was optimized via a minimum mean absolute error criterion with 2000k steps by Adam [24] with $\beta = (0.9, 0.98)$, and $\epsilon = 1.0 \times 10^{-9}$. We also follow the same learning rate schedule in [25].

Figure 4 shows the subjective evaluation results. The overall difference in scores between ground truth and synthetic speech was greater than when using features extracted from natural speech since acoustic features predicted by TTS were degraded from the original one. Comparing CONVENTIONAL (M=2) and (M=4), the degradation of latter was noticeable, which might be due to this data mismatch. On the other hand, PROPOSED (M=2) and (M=4) outperformed CONVENTIONAL (M=2) and (M=4), respectively. As PROPOSED (M=4) is comparable to CONVENTIONAL (M=2), we found the multi-sample joint modeling was also effective in the TTS task. PROPOSED (M=8) obviously was degraded agreeing with the results of the previous section. These results demonstrated that our methods can robustly vocode acoustic features predicted by TTS without degrading naturalness unless the number of simultaneous generated samples is increased to $M=8$.

4.4. Discussion

By observing the spectrograms of each method used for the evaluation, we verified if the proposed vocoders improved the reproducibility at mid-high range, a task that the conventional method had great difficulty in performing at $M > 2$. Figure 5 shows mel-spectrograms of each method obtained by vocoding acoustic features extracted from natural speech. A comparison of CONVENTIONAL (M=2) yielded a spectrogram close to that of ground truth, while CONVENTIONAL failed to well reproduction of mid-high frequency components (e.g. “Detail 1” in Fig. 5). Both PROPOSED (M=2) and (M=4) achieved natural spectrograms similar to CONVENTIONAL (M=2). These results reveal that the proposed method contributes to overcoming the spectral reproducibility issue that has been a drawback of the conventional method. Furthermore, the results of Sections 4.2 and 4.3 also showed that PROPOSED (M=4) achieved 1.47 times faster vocoding while outputting synthesized speech of compa-

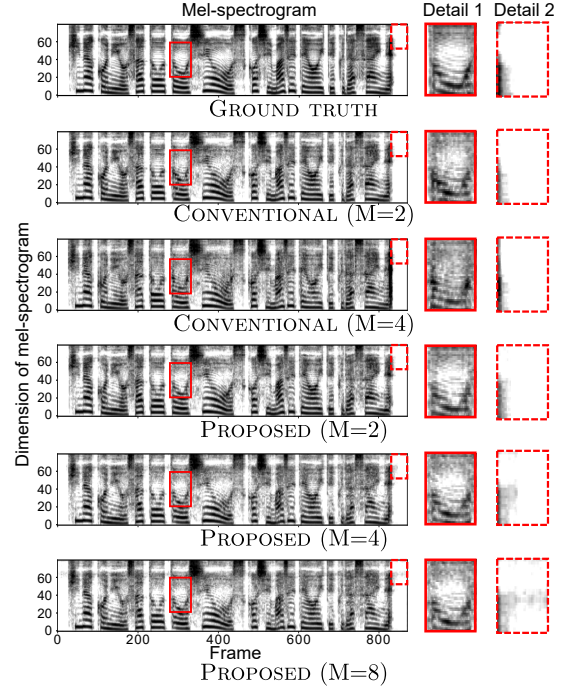


Figure 5: The mel-spectrogram comparison obtained by all methods. All synthetic speech samples were obtained by using features extracted from natural speech. From top to bottom: GROUND TRUTH, CONVENTIONAL (M=2), (M=4), PROPOSED (M=2), (M=4), and (M=8). CONVENTIONAL (M=4) had difficulty reproducing the mid and high-frequencies due to sample discontinuities, whereas CONVENTIONAL (M=2) and PROPOSED (both of M=2 and 4) still yielded spectrograms relatively close to that of GROUND TRUTH (e.g. “Detail 1”). PROPOSED (M=8) struggled with undesirable spectral components via noisy sound even in silence (e.g. “Detail 2”).

parable quality to CONVENTIONAL (M=2). However, PROPOSED (M=8) produced undesired spectrum components in silence and sometimes also sounded clicking noise (e.g. “Detail 2” in Fig. 5). We suspect that this is attributed to the complexity of embedding previous eight samples’ information in the FC3’s input $z_{t+\tau \vee m}$. In order to overcome this problem, the dimension of $z_{t+\tau \vee m}$ should be increased for further model representational performance. Since there is a trade-off between increasing the number of dimensions and inference speed, we need to compensate the inference speed by developing a model with higher sparsity.

5. Conclusions

In this work, we proposed the subband WaveRNN-based fast neural vocoder; it jointly models multiple simultaneously generated samples and subband signals. Taking M as the number of simultaneous sample predictions, the proposed vocoder with $M=4$ achieved the comparable quality as the conventional method with $M=2$ for both acoustic features extracted from natural speech and those predicted by TTS. Its speed enhancement was 1.47 times, and the proposed vocoder with $M=4$ yielded a measured RTF value of 0.064 on a single-threaded CPU. Mel-spectrogram observations of each method confirmed that the proposed method offers improved reproducibility of the mid- to high-frequency spectrum components, which was an issue with the conventional method.

6. 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,” *CoRR abs/1609.03499*, 2016.
- [2] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Van Den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, “Parallel WaveNet: Fast high-fidelity speech synthesis,” *Proc. ICML*, vol. 9, 2018.
- [3] W. Ping, K. Peng, and J. Chen, “Clarinet: Parallel wave generation in end-to-end text-to-speech,” *Proc. ICLR*, 2019.
- [4] B. C. Ryan Prenger, Rafael Valle, “WaveGlow: A flow-based generative network for speech synthesis,” *Proc. ICASSP*, pp. 3617–3621, 2019.
- [5] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Van Den Oord, S. Dieleman, and K. Kavukcuoglu, “Efficient Neural Audio Synthesis,” *Proc. PMLR*, pp. 2410–2419, 2018.
- [6] J.-M. Valin and J. Skoglund, “LPCNet: Improving neural speech synthesis through linear prediction,” *Proc. ICASSP*, pp. 5891–5895, 2019.
- [7] M.-J. Hwang, F. Soong, E. Song, X. Wang, H. Kang, and H.-G. Kang, “LP-WaveNet: Linear prediction-based wavenet speech synthesis,” *Proc. APSIPA*, pp. 810–814, 2020.
- [8] E. Song, K. Byun, and H.-G. Kang, “ExcitNet vocoder: A neural excitation model for parametric speech synthesis systems,” *Proc. EUSIPCO*, pp. 1–5, 2019.
- [9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Proc. NIPS*, pp. 2672–2680, 2014.
- [10] R. Yamamoto, E. Song, and J.-M. Kim, “Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” *Proc. ICASSP*, pp. 6199–6203, 2020.
- [11] K. Kumar, R. Kumar, T. de Boissiere, L. Gestein, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, “MelGAN: Generative adversarial networks for conditional waveform synthesis,” *Proc. NeurIPS*, 2019.
- [12] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” *Proc. NeurIPS*, 2020.
- [13] V. Popov, M. Kudinov, and T. Sadekova, “Gaussian LPCNet for multisample speech synthesis,” *Proc. ICASSP*, pp. 6204–6208, 2020.
- [14] T. Q. Nguyen, “Near-perfect-reconstruction pseudo-QMF banks,” *IEEE Trans. Speech and Audio Processing*, vol. 42, no. 1, pp. 65–76, 1994.
- [15] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei, D. Su, and D. Yu, “DurIAN: Duration informed attention network for speech synthesis,” *Proc. INTERSPEECH*, pp. 2027–2031, 2020.
- [16] B. Zhai, T. Gao, F. Xue, D. Rothchild, B. Wu, J. Gonzalez, and K. Keutzer, “SqueezeWave: Extremely lightweight vocoders for on-device speech synthesis,” *arXiv:2001.05685*, 2020.
- [17] H. Kanagawa and Y. Ijima, “Multi-sample subband WaveRNN via multivariate Gaussian,” *Proc. ICASSP*, pp. 8427–8431, 2022.
- [18] P. L. Tobing, Y.-C. Wum, T. Hayashi, K. Kobayashi, and T. Toda, “Efficient shallow wavenet vocoder using multiple samples output based on laplacian distribution and linear prediction,” *Proc. ICASSP*, pp. 7204–7208, 2020.
- [19] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, “Multi-band melgan: Faster waveform generation for high-quality text-to-speech,” *Proc. SLT*, pp. 492–498, 2021.
- [20] M. Zhu and S. Gupta, “To prune, or not to prune: exploring the efficacy of pruning for model compression,” *Proc. ICLR*, 2018.
- [21] S. Narang, E. Undersander, and G. Diamos, “Block-sparse recurrent neural networks,” *Proc. ICLR*, 2018.
- [22] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, “On the variance of the adaptive learning rate and beyond,” *Proc. ICLR*, 2020.
- [23] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Fastspeech 2: Fast and high-quality end-to-end text to speech,” *Proc. ICLR*, 2021.
- [24] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *Proc. ICLR*, 2015.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Proc. NIPS*, 2017.