Evaluating the Extrapolation Capabilities of Neural Vocoders to Extreme Pitch Values

Olivier Perrotin, Hussein El Amouri, Gérard Bailly, Thomas Hueber

INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Evaluating the Extrapolation Capabilities of Neural Vocoders to Extreme Pitch Values

Olivier Perrotin, Hussein El Amouri, Gérard Bailly, Thomas Hueber

[1]Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, F-38000 Grenoble, France
olivier.perrotin@grenoble-inp.fr

Abstract

Neural vocoders are systematically evaluated on homogeneous train and test databases. This kind of evaluation is efficient to compare neural vocoders in their "comfort zone", yet it hardly reveals their limits towards unseen data during training. To compare their extrapolation capabilities, we introduce a methodology that aims at quantifying the robustness of neural vocoders in synthesising unseen data, by precisely controlling the ranges of seen/unseen data in the training database. By focusing in this study on the pitch (F_0) parameter, our methodology involves a careful splitting of a dataset to control which F_0 values are seen/unseen during training, followed by both global (utterance) and local (frame) evaluation of vocoders. Comparison of four types of vocoders (autoregressive, source-filter, flows, GAN) displays a wide range of behaviour towards unseen input pitch values, including excellent extrapolation (WaveGlow); widely-spread F_0 errors (WaveRNN); and systematic generation of the training set median F_0 (LPCNet, Parallel WaveGAN). In contrast, fewer differences between vocoders were observed when using homogeneous train and test sets, thus demonstrating the potential and need for such evaluation to better discriminate the neural vocoders abilities to generate out-of-training-range data.

Index Terms: neural vocoders, pitch, unseen data, synthesis

1. Introduction

Most speech synthesis processing pipelines end with a vocoder that generates audio waveforms from a set of acoustic features. Non-neural vocoders classically use a speech signal model to reconstruct speech from a small set of descriptors [1, 2], and their robustness to a wide range of input parameters has led to their extensive use in speech or singing generation for the past decades [3, 4]. Yet, the hand-craft choice of parameters along with the absence of phase information in the acoustic features have been a strong limitation to the generated speech quality. Recent introduction of neural vocoders has leveraged the issue of quality by learning from data to predict audio waveforms. However, this tremendous rise of quality has been at the expense of a strong data dependency – neural vocoders poorly generalise to unseen data [5], a recurrent issue in deep learning applications. As a consequence, most newly proposed neural vocoders are evaluated on a test set that is homogeneous with the train set to demonstrate the vocoders' ability on the most favourable conditions. Along the same line, recent comparisons of neural vocoders performances also used homogeneous train and test sets [6, 7], without giving insights on the vocoders abilities to extrapolate to new data.

The extension of neural vocoders capacity to synthesise wider ranges of data has been tackled first by increasing the size of the training set, using for instance massive multi-speaker databases [8, 9]. Nevertheless, while pushing the boundary between seen and unseen data as far as possible, these studies do not address the issue of extrapolation to unseen data. Alternatively attempts have been made to propose network architectures that allow neural vocoders to be less sensitive to unseen F_0 [10, 11]. For evaluation, they artificially created unseen F_0 by shifting the original F_0 [10] or randomly generated new F_0 trajectories [11], but at the risk that the interaction with other input features might cause a degradation of quality. Moreover, these models were assessed at utterance scale, i.e. obtaining average performances over sequences of various input values. Yet, we believe that quantification of neural vocoders behaviours at a frame scale, i.e. for each input value, would bring valuable information on their capability to tackle various range of input and allow to better target relevant improvements.

Therefore we propose a methodology to evaluate the performance of neural vocoders outside of their "comfort zone" by synthesising unseen data, with a focus on F_0. This methodology relies on training models on a carefully segmented database that excludes ranges of F_0 values from training while leaving them in the test set. Then, we evaluated four classes of neural vocoders to provide insights on their behaviour towards synthesising unknown F_0. After reviewing the different classes of neural vocoders in section 2, section 3 details our methodology including database segmentation and model training. Section 4 reports the results of the study before concluding in section 5.

2. Neural Vocoders

Neural vocoders generate audio samples given acoustic features that carry speech characteristics at each time step. Most systems use non-parametric acoustic features such as 80-band mel-spectrograms that implicitly contain both pitch and spectral envelope information. Few uses explicit F_0 values and smoothed spectral envelopes [11, 12, 13], and fall within one of the four categories of neural vocoders studied here and described below.

The first proposed neural vocoders are *autoregressive*, i.e., they generate one audio sample at a time given previously predicted samples. Historically derived from image processing, the first neural vocoder WaveNet [14] is built from stacks of dilated convolution layers that have been proven particularly efficient to generate high quality output, but also extremely costly. To reduce complexity, recurrent neural networks that are more suited to the processing of time series have been introduced [15, 16, 8, 9]. With the addition of optimisation techniques, WaveRNN manages to synthesise speech in real-time [16]. To go further in model simplification, hybrid signal-neural models include *source-filter* decomposition of speech [17] in their model so that the latter can focus on source prediction only, the filter being derived from the input acoustic features [12, 13, 18]. As mentioned above, part of these models like LPCNet explicitly provide F_0 values in the acoustic features that condition the system [12]. Autoregressive models are adapted to on-

http://dx.doi.org/10.21437/Interspeech.2021-1547

line prediction and have been successfully been simplified to work in real-time, but they'll always been limited in computation efficiency by the impossibility to parallelise calculations. *Flow-based* networks have tackled the issue of generating all samples at a time by removing the autoregression process in generation. Since WaveNet training process is not autoregressive, it is used as a basis in most flow-based models. One solution it to use this training as a teacher model and transfer its knowledge to a student non-autoregressive generation network with probability distillation [19, 20]. Alternatively, authors have built reversible non-autoregressive networks from WaveNet with the help of affine coupling layers [21, 22, 23]. Teacher/student networks have been shown difficult to train and WaveGlow is one the most used flow-based neural vocoder in spite of its important memory footprint [21]. Another solution to train non-autoregressive generators is to use *generative adversarial networks* (GAN) [24, 25, 26]. Generators are often non-autoregressive stacked dilated-convolution based networks again derived from WaveNet, and trained adversarially against a discriminator that attempts to distinguish synthesised from natural speech. Auxiliary losses are often used to minimise the distance between acoustic features extracted from the synthesised speech and the input features. The use of GANs has drastically decreased model sizes and training and generation times, compared to flow-based models [25].

In this study, we selected neural vocoders that are representative of each class: WaveRNN [16] (autoregressive), LPCNet [12] (source-filter with explicit F_0 input), WaveGlow [21] (flow-based) and Parallel WaveGAN [25] (GAN-based).

3. Experimental setup

3.1. Database preparation

To compare the neural vocoders ability to generate unseen F_0, we built a dataset that excludes specific ranges of F_0. We used for this sake the LJ Speech dataset [27], consisting of approximately 24h of reading from an English female speaker, recorded as part of the LibriVox project. F_0 was extracted from 10 msec-spaced frames on the full database with Praat [28], whose distribution in semitones (ST) is displayed on top of Fig. 1. It displays one main lobe and two small side lobes. The latter are a consequence of the restricted F_0 range given to Praat for extraction ($[75; 600]$ Hz), and relate to creaky and pseudo-harmonic hissing sounds for the low and high lobes, respectively. Overall, we defined three F_0 classes: the main lobe that contains 95% of the values, delimited by the middle green arrow; the tails that contain 4% of the values on each side of the main lobe, hence a total of 8% of the values, included in the two red rectangles. Finally the two extreme lobes each contain 1% of the values and are classified as outliers.

In our method, we aim at excluding all F_0 contained within the tails (red rectangles) from our training set, while keeping them in our test set. We first built the latter by selecting the 100 utterances that contains the most low-tail F_0 values and the 100 utterances that contains the most high-tail F_0 values. The resulting test set F_0 distribution (bottom of Fig. 1) shows that tail-F_0 are over-represented, to voluntary increase the synthesis difficulty. Second, we built a train set that excludes all tail-F_0 values. During training, all neural vocoders split utterances in small chunks that are processed independently and in random order. Thus, after excluding the test set, we split all remaining utterances into 800 msec chunks (larger than the ones used in the selected neural vocoders). End of utterance chunks that

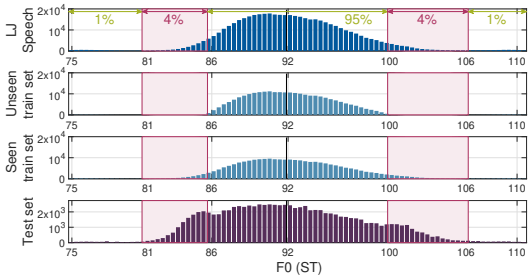

Figure 1: *Distribution of F_0 values per frame in the full LJ Speech data set (top); unseen training set (second row); seen training set (third row); test set (bottom).*

were shorter than 800 msec were discarded, and only chunks that did not contain any tail-F_0 values were kept for the training set, that finally contained 54.5% of the full dataset, i.e., about 12h of speech. We call *unseen* this train set, whose corresponding F_0 distribution (second row of Fig. 1) displays no tail-F_0 values. Finally, as a baseline, we created a second training set called *seen* that contains the same number of chunks, but selected randomly from LJ Speech after exclusion of the test set. The third row of Fig. 1 shows the corresponding F_0 distribution, that is representative of the full LJ speech corpus.

3.2. Training models

We used open-source implementations of neural vocoders [29, 30, 31, 32]. LPCNet and WaveGlow are provided by the original paper authors while others are re-implementations. Default training setups were used, or following their respective paper description when not specified in the code. LPCNet used audio sampled at 16 kHz and 20 acoustic features (pitch, pitch correlation, 18 Bark Frequency Cepstrum Coefficients). The three other models sample audio at 22.05 kHz and use 80-band mel-spectrograms as acoustic features. WaveRNN was trained with a batch size of 32 and a learning rate of $1e^{-4}$ for 1000K iterations. Important quality variations were observed between the last iterations, so we refined the model with additional 250K iterations at a learning rate of $1e^{-5}$. LPCNet was trained with a batch size of 64 and a learning rate of $5e^{-4}$ for 413K iterations (120 epochs, as in [12]). WaveGlow was trained with a batch size of 12 and a learning rate of $1e^{-4}$ for 580K iterations, and fine-tuned with additional 40K iterations with a learning rate of $5e^{-5}$ [21]. Parallel WaveGAN was trained for 400K iterations. Learning rates were $1e^{-4}$ and $5e^{-5}$ for the generator and discriminator, respectively, and were halved every 200K. The discriminator was frozen during the first 100K iterations [25]. Overall, the numbers of trained parameters are WaveRNN: 4.2M; LPCNet: 1.2M; WaveGlow: 87.9M; Parallel WaveGAN: 18.8M. All models were trained twice, on both the *unseen* and *seen* datasets. For each training, given their respective chunk size in batch, batch size and number of iterations, each model processed a total of WaveRNN: 689h; LPCNet: 1102h; WaveGlow: 1498h; Parallel WaveGAN: 711h of audio.

4. Comparison of Neural Vocoders

Assessment of the ability of neural vocoders to synthesise unseen F_0 was conducted both at utterance and at frame level. For this sake, all 200 test utterances were generated 8 times: with the 4 neural vocoders (*system* factor) trained on either the *unseen* or *seen* training set (*training* factor).

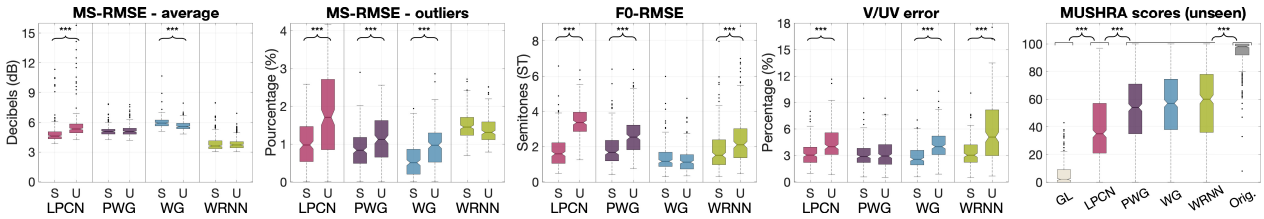

Figure 2: *Evaluation of neural vocoders. From left to right: average MS-RMSE; percentage of MS-RMSE outliers; F_0-RMSE; percentage of V/UV errors. First four panels display the measures computed by utterances for each neural vocoder (in colours), and each training set (left: seen (S); right: unseen (U)). Last panel: MUSHRA scores obtained on the unseen training set only.*

Table 1: *F-scores from the ANOVA performed on each objective measure against the system and training factors and their interaction. All effects are significant ($p < 0.01$).*

Factor (d.f.)	**MS-RMSE**		**F_0-RMSE**	**V/UV**
	Average	**Outliers**		**error**
System (3)	352	118	114	36
Training (1)	8	116	218	190
Interaction (3)	31	35	48	25

4.1. Global utterance-based evaluation

We evaluated global performance of neural vocoders with a series of objective and subjective measures. First, output quality is assessed using root-mean-square-error (RMSE) between 80-band mel-spectrograms (MS) extracted from original and generated signals, using frames of 92 msec and hop size of 10 msec. RMSE is computed on each frame to provide a MS-RMSE distribution for each utterance, from which two values are extracted: 1) the mean of the distribution as a measure of average quality; 2) the percentage of frames whose RMSE is higher than three standard deviations above the mean (outliers). F_0 of generated waveforms were also extracted with Praat, with the same parameters as the original waveforms (section 3.1), and RMSE between generated and original F_0 were computed for each utterance, with F_0 expressed in ST. Finally, the percentage of voiced-unvoiced (V/UV) errors per utterance is calculated, since unvoicing could be a mean for the neural vocoders to avoid synthesising unseen F_0 values. The four first panels of Fig. 2 display the distribution of these objective measures calculated on all test utterances, and organised by *system* (in colours) and *training* factors. For each measure, a 2-way ANOVA using *system* and *training* factors demonstrated the significance of both factors and their interaction. Corresponding effect sizes (F-scores) are summarised in Table 1. Post-hoc pair-wise comparison between *unseen* and *seen trainings* for each *system* assessed with Tukey HSD tests are superimposed on Fig. 2.

Average MS-RMSE (first panel of Fig. 2) is often considered as a measure of global quality. Overall, WaveGlow is the worse system, followed by Parallel WaveGAN and LPCNet, while WaveRNN outperforms all. Conversely, statistical analyses on MS-RMSE outliers (second panel of Fig. 2) show that overall *trainings*, LPCNet and WaveRNN have similar performance and the highest percentage of MS-RMSE outliers while WaveGlow has the smallest and Parallel WaveGAN is in-between. The percentage of MS-RMSE outliers is rarely studied but is well-related to salient local artefacts in the generated waveforms. Then, analysis of the *training* factor shows that synthesising unseen F_0 have a small effect on the average MS-RMSE, and only on LPCNet and WaveGlow. By contrast, all systems but WaveRNN trained on unseen F_0 have higher percentages of MS-RMSE outliers than the ones trained on seen F_0. The larger effect of the *training* factor on MS-RMSE outliers than on average MS-RMSE emphasises that synthesising unseen F_0 leads to local artefacts in the audio waveform rather than altering the global quality.

Regarding F_0 reconstruction (third panel of Fig. 2), overall *trainings* WaveGlow has the lowest F_0-RMSE, followed by Parallel WaveGAN and WaveRNN (not significant difference), and then LPCNet. A similar order is observed regarding the degradation of performance given unseen F_0 values: WaveGlow is not altered, followed by a small degradation for WaveRNN and Parallel WaveGAN ($\approx +1$ ST), and then LPCNet ($\approx$ +2 ST). When synthesising seen F_0, all models have a similar behaviour regarding V/UV errors. Nevertheless, the degradation given unseen F_0 is particularly pronounced for WaveRNN (almost double) but does not affect Parallel WaveGAN.

We assessed perceptive quality with a MUSHRA-based test [33]. We selected 20 utterances from the test set having the most high- (resp. low-) tail frames, respectively (10 each). For each, 6 versions were presented at a time: original; synthesis with the four vocoders trained on the unseen dataset; and with Griffin-Lim (GL) [34] as a low-anchor. All utterances were downsampled at 16 kHz to comply with LPCNet. Participants had to rank them by naturalness on a scale from 0 to 100. 30 native English speakers recruited on the Prolific Academic platform [35] participated in the experiment. As a control for listening ability, we discarded the 7 subjects who rated the reference (resp. low anchor) lower than 80 (resp. higher than 20) in more than 50% of the cases. Last panel of Fig. 2 summarises the results. Scores attributed to all vocoders are around 30% below those in [7], that used the full LJ Speech database for training and similar anchors in their MUSHRA. Although comparison between studies is not straightforward, this points out the severe degradation caused by synthesis of unseen data. A Kruskal-Wallis rank-sum test showed a significant effect of the *system* factor ($\chi^2 = 1327.5$, df = 5, $p < 0.01$) and a post-hoc Dunn test identified four groups of systems that significantly differed ($p < 0.01$): Griffin-Lim; LPCNet; Original; and the 3 remaining vocoders. The lower performance of LPCNet against the three other vocoders shows a stronger correlation of the subjective perception with MS-RMSE outliers and F_0-RMSE, where LPCNet had the lowest performance as well, than with average MS-RMSE and V/UV error. This suggests that subjects were more sensitive to local artefacts and F_0 reconstruction rather than overall MS quality. Moreover, the high percentage of V/UV error didn't prejudice WaveRNN. On the contrary, subjects might have been more tolerant to unvoicing than erroneous F_0 values.

To summarise, the *training* factor has little impact on global

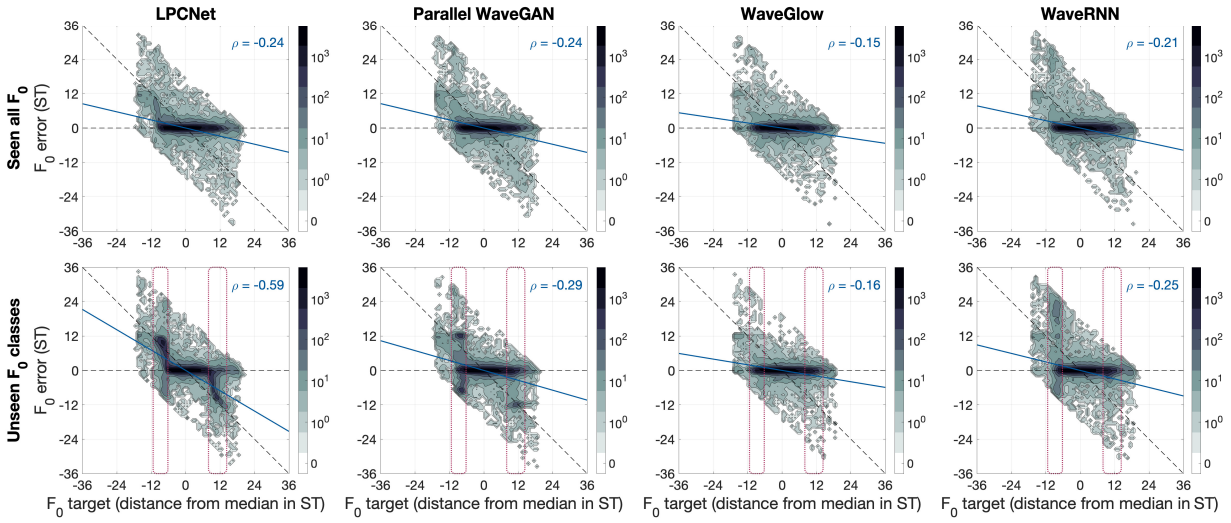

Figure 3: *Test frames distribution on the (F_0 target $\times$ F_0 error) plan, where F_0 target is centred around the median of LJ Speech corpus, and displayed on a colour log-scale. The red rectangles highlight the range of unseen F_0 target. Blue lines and corresponding values represent correlations across frames. Each column corresponds to a system, and rows correspond to training on seen and unseen data.*

quality (average MS-RMSE) but degrades performance locally (outliers MS-RMSE) as well as F_0-reconstruction, the two criteria that correlates the most with perceptive quality.

4.2. Local frame-based evaluation

Local frame-based evaluation consists in agglomerating all frames of the 200 test utterances on which we derived two values: 1) the distance of the target (input) F_0 from the median of the full LJ Speech distribution, that is an indication of difficulty to synthesise the input; 2) The error between the synthesised F_0 and the input. For each *systems* and *training* set, we derived the distributions of all frames on the (target $\times$ error) plan, that are displayed in Fig. 3. On each panel, the colour (log-scale) represents the density of frames associated to all possible (F_0 target, F_0 error) pairs. Correlation coefficients across all frames are given in blue along with the associated regression line. We expect a low correlation between input and error. The red rectangles highlight the ranges of unseen input F_0 values. Systems trained on seen F_0 values (top row) display a high density on the horizontal axis: F_0 errors are small for any input. Correlation coefficients show that all systems behave similarly, except WaveGlow that is able to have a higher density along the horizontal axis ($\rho = -0.15$). Second row (unseen F_0) shows four different behaviours. First, WaveGlow behaves similarly than for seen F_0 values, in line with F_0-RMSE analysis. Looking at WaveRNN, we observe a spread of the density across the F_0 error in the left red rectangle, i.e. for unseen input F_0 only. It therefore shows a failure to render low-tail F_0 values, but with no specific error pattern. By contrast, the densities within red rectangles for Parallel WaveGAN and LPCNet are high around the diagonal, corresponding to a synthesised F_0 that equals the median of LJ Speech distribution. The effect is particularly pronounced for LPCNet, where the frontier between seen and unseen F_0 is striking, with an abrupt F_0 error increase at the borders of the red rectangles. Compared to all other systems, F_0 values are given explicitly to LPCNet which learns an embedding for each during training. Observation of pitch embedding weights for both *trainings* displays random (not-learnt) embeddings for out-of-training-range F_0 values in both cases, thus explaining high errors when those are used in generation. While explicit F_0 input might provide more control, it is extremely sensitive to unseen values, while non-parametric mel-spectrograms might contain enough information to compensate for unseen F_0 to a certain extent. A recent neural vocoder proposed in the continuity of LPCNet [36] got rid of explicit F_0 for more robustness, suggesting similar conclusions to this study.

5. Conclusion

In order to precisely quantify the extrapolating capabilities of neural vocoders, we proposed a methodology that consists in 1) constructing a database that excludes precise ranges of F_0 values from a training set and 2) applying a series of evaluation measures at a global and local scale to quantify neural vocoders behaviours towards seen and unseen F_0 values. The discrepancy between global assessment of quality (average MS-RMSE) and local assessment (MS-RMSE outliers; frame-based F_0 errors) has demonstrated the need for frame-based analysis to discriminate between neural vocoders regarding their extrapolation capacities to extreme pitch values. Indeed, large F_0 errors only happen on unseen F_0 input values, leading to abrupt and localised degradation of quality. Moreover, while WaveGlow was shown perfectly robust to unseen F_0, WaveRNN displays widely spread errors along with more unvoicing errors, while LPCNet and Parallel WaveGAN generate F_0 around the median F_0 of the training set. These striking differences of behaviour suggests that different and adapted solutions must be found for each type of neural vocoder architecture to increase their extrapolation capabilities. Finally, we hope that this methodology can be a useful tool for research on extrapolation capabilities of neural vocoders, that can be extended to various input features, from phonemes identity to voice quality.

6. Acknowledgements

This work was performed using HPC resources from GENCI-IDRIS (Grant 2021-AD011011542R1) and supported by the BPI project THERADIA and MIAI@Grenoble-Alpes (ANR-19-P3IA-0003).

7. References

[1] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Comm.*, vol. 27, no. 3-4, pp. 187–207, April 1999.

[2] M. Morise, F. Yokomori, and K. Ozawa, "World: A vocoder-based high-quality speech synthesis system for real-time applications," in *IEICE Trans. on Information and Systems*, vol. E99.D, no. 7, July 2016, pp. 1877–1884.

[3] S. King, "An introduction to statistical parametric speech synthesis," *Sadhana*, vol. 36, no. 5, pp. 837–852, October 2011.

[4] H. Doi, T. Toda, T. Nakano, M. Goto, and S. Nakamura, "Singing voice conversion method based on many-to-many eigenvoice conversion and training data generation using a singing-to-singing synthesis system," in *Proc. of the Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, Hollywood, CA, USA, December 3-6 2012, pp. 1–6.

[5] P.-c. Hsu, C.-h. Wang, A. T. Liu, and H.-y. Lee, "Towards robust neural vocoding for speech generation: A survey," *arXiv*, vol. abs/1912.02461, 2020.

[6] X. Wang, J. Lorenzo-Trueba, S. Takaki, L. Juvela, and J. Yamagishi, "A comparison of recent waveform generation and acoustic modeling methods for neural-network-based speech synthesis," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Calgary, AB, Canada, April 15-20 2018, pp. 4804–4808.

[7] P. Govalkar, J. Fischer, F. Zalkow, and C. Dittmar, "A comparison of recent neural vocoders for speech signal reconstruction," in *Speech Synthesis Workshop*, Vienna, Austria, September 20-22 2019, pp. 7–12.

[8] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards Achieving Robust Universal Neural Vocoding," in *Proc. of Interspeech*, Graz, Austria, September 15-19 2019, pp. 181–185.

[9] D. Paul, Y. Pantazis, and Y. Stylianou, "Speaker Conditional WaveRNN: Towards Universal Neural Vocoder for Unseen Speaker and Recording Conditions," in *Proc. of Interspeech*, Shanghai, China, October 25-29 2020, pp. 235–239.

[10] Y.-C. Wu, T. Hayashi, P. L. Tobing, K. Kobayashi, and T. Toda, "Quasi-periodic wavenet vocoder: A pitch dependent dilated convolution model for parametric speech generation," in *Proc. of Interspeech*, Graz, Austria, September 15-19 2019, pp. 196–200.

[11] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2020.

[12] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Brighton, UK, May 12-17 2019, pp. 5891–5895.

[13] Y. Ai and Z.-H. Ling, "Knowledge-and-Data-Driven Amplitude Spectrum Prediction for Hierarchical Neural Vocoders," in *Proc. of Interspeech*, Graz, Austria, October 25-29 2020, pp. 190–194.

[14] 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," *arXiv*, vol. abs/1609.03499, 2016.

[15] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio, "Samplernn: An unconditional end-to-end neural audio generation model," in *Int. Conf. on Learning Representations (ICLR)*, Toulon, France, April 24-26 2016.

[16] 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," in *Proc. of Machine Learning Research*, vol. 80, Stockholm, Sweden, July 10-15 2018, pp. 2410–2419.

[17] G. Fant, *Acoustic Theory of Speech Production*. Mouton, 1960.

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

[19] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. 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," in *Proc. of Machine Learning Research*, vol. 80, Stockholmsmässan, Stockholm, Sweden, July 10-15 2018, pp. 3918–3926.

[20] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," in *International Conference on Learning Representations (ICLR)*, ser. ICLR 2019, New Orleans, LA, USA, May 6-9 2019.

[21] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Brighton, UK, May 12-17 2019, pp. 3617–3621.

[22] S. Kim, S.-G. Lee, J. Song, J. Kim, and S. Yoon, "FloWaveNet : A generative flow for raw audio," in *Proc. of Machine Learning Research*, vol. 97, Long Beach, California, USA, June 9-15 2019, pp. 3370–3378.

[23] W. Ping, K. Peng, K. Zhao, and Z. Song, "Waveflow: A compact flow-based model for raw audio," in *Proc. of the International Conference on Machine Learning*, 2020.

[24] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*. Vancouver, Canada: Curran Associates, Inc., 2019, pp. 14 910–14 921.

[25] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 4-8 2020, pp. 6199–6203.

[26] L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, "GELP: GAN-Excited Linear Prediction for Speech Synthesis from Mel-Spectrogram," in *Proc. of Interspeech*, Graz, Austria, September 15-19 2019, pp. 694–698.

[27] K. Ito and L. Johnson, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[28] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9/10, pp. 341–347, November 2001.

[29] O. McCarthy, "WaveRNN implementation." [Online]. Available: https://github.com/fatchord/WaveRNN

[30] NVIDIA, "WaveGlow implementation." [Online]. Available: https://github.com/NVIDIA/waveglow

[31] Mozilla, "LPCNet implementation." [Online]. Available: https://github.com/mozilla/LPCNet

[32] T. Hayashi, "Parallel WaveGAN implementation." [Online]. Available: https://github.com/kan-bayashi/ParallelWaveGAN

[33] I. T. Union, "Method for the subjective assessment of intermediate quality level of audio systems," International Telecommunication Union, Tech. Rep. ITU-R BS.1534-3, October 2015.

[34] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, April 1984.

[35] S. Palan and C. Schitter, "Prolific.ac—a subject pool for online experiments," *Journal of Behavioral and Experimental Finance*, vol. 17, pp. 22–27, 2018.

[36] W. B. Kleijn, A. Storus, M. Chinen, T. Denton, F. S. C. Lim, A. Luebs, J. Skoglund, and H. Yeh, "Generative speech coding with predictive variance regularization," *arXiv*, vol. abs/2102.09660, 2021.