Very Low Complexity Speech Synthesis Using Framewise Autoregressive GAN (FARGAN) with Pitch Prediction

Jean-Marc Valin, *Member, IEEE*, Ahmed Mustafa, Jan Büthe *Member, IEEE*

***Abstract*—Neural vocoders are now being used in a wide range of speech processing applications. In many of those applications, the vocoder can be the most complex component, so finding lower complexity algorithms can lead to significant practical benefits. In this work, we propose FARGAN, an autoregressive vocoder that takes advantage of long-term pitch prediction to synthesize high-quality speech in small subframes, without the need for teacher-forcing. Experimental results show that the proposed 600 MFLOPS FARGAN vocoder can achieve both higher quality and lower complexity than existing low-complexity vocoders. The quality even matches that of existing higher-complexity vocoders.**

***Index Terms*—neural vocoder, speech synthesis, GAN**

I. Introduction

SINCE the publication of the original WaveNet [1] and SampleRNN [2] vocoders, neural vocoders have found their way into a wide range of modern audio processing applications. These including text-to-speech (TTS) synthesis [3], low-bitrate speech coding [4], super resolution [5], noise suppression [6], speech codec enhancement [7], and speed-adjustment [8]. That brings neural vocoders among the core building blocks in modern speech processing. While the original vocoders had a complexity prohibiting most real-time uses, further improvements such as WaveRNN [9] and LPCNet [10] made such applications possible.

The aforementioned autoregressive vocoders all rely on explicit density estimation to synthesize the speech waveform through conditional sampling, leading to two limitations. First, their structure requires the use of teacher-forcing [11], which leads to a domain gap between training and inference that sometimes limits quality. Second, it prevents direct signal generation and the use of more advanced loss functions, as done by GAN [12] vocoders such as MelGAN [13] and HiFi-GAN [14]. On the other hand, according to [15], "autoregressive models possess an inductive bias towards learning pitch and phase". The authors argue that the phase evolution of a periodic signal is analogous to the cumulative sum problem which is easier to learn for autoregressive models than for e.g. CNNs which are limited by their finite receptive field. The authors' proposed CARGAN model uses past context as implicit pitch conditioning and is shown to more accurately represent the pitch compared to the non-autoregressive HiFi-GAN. Although CARGAN relies on teacher-forcing with respect to its autoregressive component, it can still be trained adversarially provided that the chunk size is sufficiently large.

Jean-Marc Valin is with the Xiph.Org Foundation (e-mail: jmvalin@jmvalin.ca).

Ahmed Mustafa and Jan Büthe are with Amazon Web Services (e-mail: ahdmust@amazon.com, jbuethe@amazon.com).

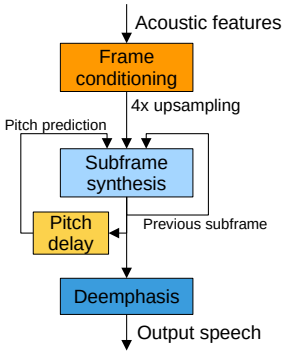

Fig. 1. Overview of FARGAN. The frame conditioning network operates on acoustic features at a 10-ms interval and outputs a conditioning latent representation at 2.5-ms interval for the autoregressive subframe synthesis network.

This paper attempts to further improve the efficiency of autoregressive GAN vocoders by expanding on both the CARGAN model and our previous Framewise WaveGAN (FWGAN) [16] vocoder. We propose (Sec. II) a framewise autoregressive GAN (FARGAN) that explicitly uses pitch-based long-term prediction as a second autoregressive feedback to improve quality and reduce complexity. Synthesizing speech based on 2.5 ms subframes to make optimal use of pitch prediction, we avoid teacher-forcing while still training on sufficiently long sequences by *unrolling* the model at training time (Sec. III). The resulting FARGAN model has a size of 820k parameters and a complexity of 600 MFLOPS. We show in Sec. IV that it provides significantly higher quality than the low-complexity vocoders like LPCNet and Framewise WaveGAN. The quality of FARGAN is even comparable to that of CARGAN and HiFi-GAN v1 whose complexity is more than 50 times higher.

II. FARGAN Overview

Although it can be used in a wide range of applications, FARGAN is designed to meet the more stringent constraints of real-time speech communications applications. For those applications, a vocoder needs to produce high-quality speech with a low algorithmic delay ($<$ 20 ms) and with sufficiently

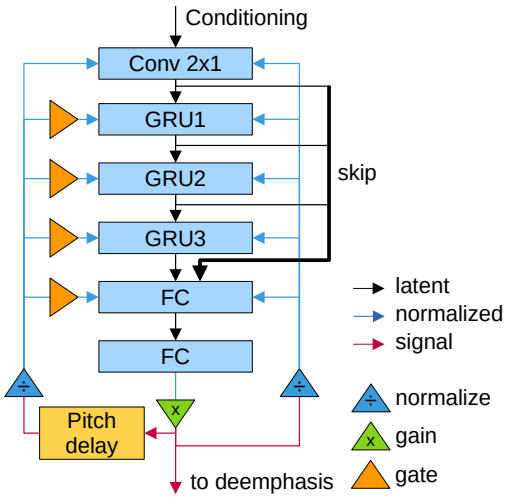

Fig. 2. Overview of the FARGAN subframe synthesis network. Multiple inputs to a layer denotes concatenation. All gains are computed from a small fully-connected layer using the conditioning as input. The normalization operations apply the inverse of the gain corresponding to the frame where the signal is used.

low complexity to avoid limiting battery life when used continuously on a mobile device CPU.

FARGAN operates on 20-dimensional acoustic features computed at a 10-ms interval on 16 kHz audio. Each frame is subdivided into 4 subframes of 2.5 ms each, or 40 samples. As for LPCNet, the acoustic features include 18 Bark-frequency cepstral coefficients (BFCC), a pitch period, and a voicing indicator. In each iteration, the model shown in Fig. 1 computes the output for an entire subframe based on the acoustic features, the previously synthesized subframe, as well as a long term (pitch) prediction based on the signal history. The frame conditioning network consists of one fully-connected layer, one 3x1 convolutional layer, and one transposed convolution layer for performing 4x up-sampling to the subframe rate. To make the pitch representation easier to learn, the conditioning network also receives a 12-dimensional embedding of the pitch similar to [10], for a total input dimension of 32.

The subframe network shown in Fig. 2 is responsible for the autoregressive property of FARGAN. The signal produced at the subframe network's output feeds back into its input in two ways. First, the last generated subframe is directly fed back to the input as a way of ensuring continuity of the signal. That autoregressive behavior is similar to that of CARGAN [15], except for the much smaller durations involved (40 samples instead of 512). The second and most important feedback path involves pitch. The input pitch period is not only used for conditioning, but is also directly used to look back in the synthesis history and extract the signal exactly one pitch period before the current subframe. For voiced speech, those samples tend to be an accurate prediction for the current subframe being synthesized. To handle cases where the pitch period T may be shorter than the subframe size N, the predicted signal $p(n)$ is obtained by

$$p(n) = \begin{cases} \hat{x}(n-T) & T \geq N \\ \hat{x}(n-2T) & \text{otherwise.} \end{cases} \quad (1)$$

The period T can never be shorter than $N/2$ since the highest pitch allowed is 500 Hz, corresponding to $T = 32$.

All layers of the subframe network use a $\tanh()$ activation. With the exception of the output layer, all layers of the subframe network also include a gated linear unit (GLU)

$$G(\mathbf{x}) = \mathbf{x} \odot \sigma(\mathbf{W}\mathbf{x}) \quad (2)$$

at their output, where $\odot$ denotes the Hadamard product, $\sigma(\cdot)$ denotes the sigmoid function, and $\mathbf{W}$ is the GLU weight matrix.

We use normalization to reduce the dynamic range of the synthesized signal. For each subframe, a single fully-connected neuron with exponential activation computes a gain from the subframe conditioning input. That gain is applied to the subframe output layer to scale the synthesized speech to its full dynamic range. In the autoregressive feedback, the previous subframe and pitch prediction are renormalized using the gain of the subframe where they are used rather than the one where the speech was generated.

We find that simply feeding the autoregressive components back to the input of the subframe network does not result in optimal use of that information – likely due to vanishing gradient. For that reason we also feed them to all the other subframe layers. In the case of the pitch prediction, we also add a gate that avoids the prediction being used for unvoiced speech. The gate values are computed from the conditioning in the same way as the gain described above.

The output of the subframe network is de-emphasized with the first-order IIR filter $H(z) = 1/(1-\alpha z^{-1})$, with $\alpha = 0.85$. Operating in the pre-emphasized domain reduces the precision required in the internal representation and, combined with the gain normalization described above, makes it possible to use 8-bit quantization throughout the model for both the weights and the activations.

A. Computational Considerations

To make FARGAN useful on a wide range of devices, we need to reduce both the number of operations (multiplications and additions) required, but also the size of the model. Fewer operations has obvious benefits in terms of speed and power consumption, but model size is also very important. A smaller model will not only reduce the cache/memory bandwidth required, but also allow the model to be held in a smaller and faster cache. Moreover, reducing the size of each weight – in our case from 32-bit floating-point to 8-bit integers – makes it possible to compute 4 times more operations at a time for the same SIMD (single instruction, multiple data) vector length.

The choice of $\tanh()$ and sigmoid activation above is motivated by the fact that their ± 1 bounds make it easy to quantize to 8 bits (unlike the unbounded ReLU). Similarly, the use of 2.5 ms subframes compared to the 10-ms subframes in our previous Framewise WaveGAN work further reduces the model size for a given complexity. The proposed model weights can thus fit in less than 1 MB, corresponding to the L2 cache of newer CPUs, or the L3 cache of older CPUs.

III. Training

Unlike many other autoregressive vocoders, FARGAN training does not (and cannot due to its structure) involve teacher forcing [11]. Instead, the subframe network is *unrolled* in time in such a way that the autoregressive components used at training time are based on the synthesized speech, rather than the ground truth speech. The authors are aware of several unsuccessful attempts (including their own) at adding direct pitch prediction to enhance the efficiency of LPCNet. One of the likely culprits for those failures is the use of teacher forcing. That is another reason for seeking to avoid teacher forcing in FARGAN. Due to the small model size and the framewise generation, training the unrolled model is still fast enough.

A. Pretraining

Let $X_L(\ell,k)$ denote the short-time Fourier transform (STFT) of signal x with window size L for frame ℓ at frequency k and 75% overlap. We define the spectral loss $\mathcal{L}_L$ between the ground truth signal x and the synthesized signal $\hat{x}$ as

$$\mathcal{L}_L = \sum_\ell \sum_k \left| |\hat{X}_L(\ell,k)|^\gamma - |X_L(\ell,k)|^\gamma \right| , \quad (3)$$

where $\gamma = 0.5$ approximates the perceived loudness [17].

For pre-training, we use a multi-resolution spectral loss

$$\mathcal{L}^{(S)} = \mathcal{L}_{80} + \mathcal{L}_{160} + \mathcal{L}_{320} + \mathcal{L}_{640} + \mathcal{L}_{1280} + \mathcal{L}_{2560} . \quad (4)$$

In the pretraining phase, we use sequences of 15 frames, with 10% of the sequences being 30-frame long. Pre-training for 470k updates with sequences of 15 frames and a batch size of 4096 takes approximately 2.5 days on one Nvidia A100 GPU.

B. Adversarial Training

For adversarial training we use spectrogram discriminators at multiple resolutions as proposed in [18]. We follow the architecture from [16] and use 6 STFT discriminators D_k with the modifications described in [19]: Each discriminator takes as input a log-magnitude spectrogram computed from size-2^{k+5} STFTs with 75% overlap. To simplify notation, we use $D_k(x)$ and $D_k(\hat{x})$, treating the log-magnitude STFT transform of x and $\hat{x}$ as part of the discriminator. We furthermore apply strides along the frequency axis to keep the frequency range of the receptive fields constant. This has been found to increase the ability of discriminators with high frequency resolution to detect inter-harmonic noise. Finally, we concatenate a two-dimensional frequency positional sine-cosine embedding to the input channels of every 2d-convolutional layer.

We train FARGAN as a least-squares GAN [20]. First we note that the generated signal $\hat{x}$ depends deterministically on ground-truth signal x. With this we define the adversarial part of the training loss for FARGAN as

$$\mathcal{L}_{\text{adv}}(x,\hat{x}) = \frac{1}{6}\sum_{k=1}^{6} E_x[(1 - D_k(\hat{x}))^2] + \mathcal{L}_{\text{feat}}(D_k, x, \hat{x}), \quad (5)$$

TABLE I
Objective evaluation of the different vocoders using PESQ, WARP-Q and mean pitch error (MPE)

Condition	PESQ	WARP-Q	MPE
FARGAN	**3.298**	0.587	**4.108**
small	3.241	0.615	4.172
no-pitch	3.174	0.608	4.239
no-AR	2.859	0.655	4.457
CARGAN	3.127	0.559	4.322
HiFi-GAN v1	3.024	**0.495**	5.501
HiFi-GAN v3	2.373	0.651	6.715
LPCNet	2.539	0.694	5.303
FWGAN	2.833	0.648	5.063

where $\mathcal{L}_{\text{feat}}$ denotes the standard feature matching loss, i.e. the mean of the L_1 losses of hidden layer outputs for x and $\hat{x}$.

The complete training loss for FARGAN includes the pre-training spectral loss, such that

$$\mathcal{L}_{\text{tot}}(x,\hat{x}) = \mathcal{L}_{\text{adv}}(x,\hat{x}) + \mathcal{L}^{(S)}(x,\hat{x}). \quad (6)$$

Simultaneously, the discriminators are trained to minimize the loss

$$\mathcal{L}_{D_k}(x,\hat{x}) = E_x[D_k(\hat{x})^2 + (1 - D_k(x))^2]. \quad (7)$$

Adversarial training is carried out on 60-frame sequences for another 50 epochs with a fixed learning rate of $2 \cdot 10^{-6}$ and a batch size of 160, which corresponds to about 380k training steps. We used the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for both FARGAN and the discriminators.

IV. Experiments & Results

We train speaker-independent FARGAN models on 205 hours of 16-kHz speech from a combination of TTS datasets [21], [22], [23], [24], [25], [26], [27], [28], [29] including more than 900 speakers in 34 languages and dialects.

We evaluate two versions of FARGAN: a proposed 820k-weight model and an even smaller 500k-weight model. As an ablation study, we evaluate the effect of removing (from the larger proposed model) the pitch prediction (replacing it with a larger history to maintain the same number of weights) and removing all autoregressive behavior.

We compare FARGAN to three other low-complexity vocoders: LPCNet [10], Framewise WaveGAN [16], and HiFi-GAN [14] v3. As a reference, we also include CARGAN and HiFi-GAN v1, which have a much higher-complexity than the proposed vocoder. All evaluations are conducted at 16 kHz and all vocoders are trained using the same datasets as FARGAN. The complete implementation of FARGAN is available under an open-source license[1].

We evaluated the algorithms using 192 clean English speech clips from the NTT Multi-Lingual Speech Database for Telephonometry and 192 clean English clips from the PTDB-TUG [30] database[2]. No items from these databases were included in the training data.

[1] https://gitlab.xiph.org/xiph/opus/-/tree/spl_fargan/dnn/torch/fargan

[2] Audio samples are available at https://ahmed-fau.github.io/fargan_demo/

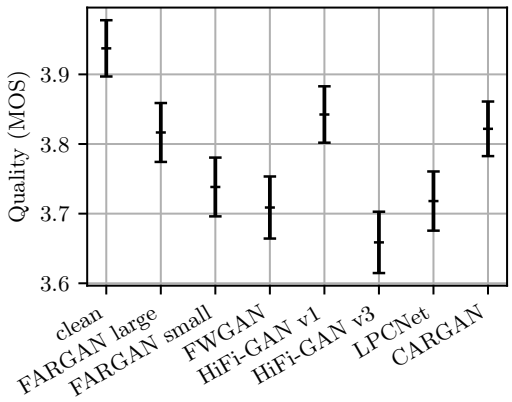

Fig. 3. P.808 mean opinion score (MOS) results including the 95% confidence intervals.

TABLE II
COMPLEXITY OF THE DIFFERENT VOCODERS. THE NUMBER OF OPERATIONS, EXPRESSED IN GFLOPS COUNTS ONE MULTIPLY-ADD OPERATION AS TWO FLOPS. WHEN AVAILABLE, WE ALSO INCLUDE THE PERCENTAGE OF ONE I7-8565 CPU CORE REQUIRED FOR REAL-TIME OPERATION (INVERSE OF "REAL-TIME FACTOR").

Condition	GFLOPS	CPU (%)
FARGAN	0.6	0.8
small	0.35	0.5
CARGAN	65.9	-
HiFi-GAN v1	38.1	-
HiFi-GAN v3	2.8	-
LPCNet	2.8	4.5
FWGAN	1.2	-

We first evaluated all the vocoders objectively using PESQ [31], WARP-Q [32], as well as mean pitch accuracy (MPE), as measured in [16]. Table I shows the objective evaluation results, demonstrating that FARGAN achieves better pitch accuracy than all other vocoders. They also demonstrate the effectiveness of the autoregressive components and show that explicit pitch prediction can help achieve both higher quality and better pitch accuracy. Although PESQ and WARP-Q results are provided for all algorithms, these metrics tend to be less reliable when comparing different families of algorithms (as opposed for variants of the same base algorithm).

For subjective evaluation, we used the crowd-sourcing methodolgy from ITU-R P.808[33]. Each sample was subjectively evaluated by 9 randomly-selected naive listeners. Results in Fig. 3 show that the larger FARGAN model is statistically tied with CARGAN and HiFi-GAN v1, and significantly better ($p < 0.05$) than LPCNet, FWGAN, and HiFi-GAN v3. Moreover, the smaller FARGAN model is statistically tied with LPCNet and FWGAN, and out-performs ($p < 0.05$) HiFi-GAN v3.

A. Complexity

Table II compares the complexity of the different vocoders, both in number of operations and speed on actual hardware (when available). The proposed FARGAN model has a complexity of 0.6 GFLOPS, which about 5 times less complex than LPCNet and HiFi-GAN v3, despite providing a higher quality. Compared to the high-quality CARGAN and HiFi-GAN v1, FARGAN achieves a complexity reduction of 110x and 64x repectively, with equivalent quality. Using an optimized C implementation, FARGAN can synthesize speech in real time using less than 1% of a modern laptop or phone CPU core.

V. CONCLUSION

We have demonstrated a very-low-complexity GAN vocoder that uses pitch-prediction-based autoregression to achieve high-quality speech synthesis using only 600 MFLOPS. The proposed FARGAN vocoder achieves both higher quality and lower complexity when compared to other existing low-complexity vocoders. Moreover, it matches the quality of state-of-the-art high-complexity vocoders. We believe the demonstrated reduction in vocoder complexity opens the way for neural vocoders being used in new applications, including low-power embedded systems.

REFERENCES

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[2] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *arXiv:1612.07837*, 2016.

[3] J. Shen, R. Pang, R.J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[4] W. B. Kleijn, F. SC Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, "WaveNet based low rate speech coding," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 676–680.

[5] H. Liu, W. Choi, X. Liu, Q. Kong, Q. Tian, and D. Wang, "Neural vocoder is all you need for speech super-resolution," *arXiv:2203.14941*, 2022.

[6] S. Maiti and M. I. Mandel, "Parametric resynthesis with neural vocoders," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 303–307.

[7] J. Skoglund and J.-M. Valin, "Improving opus low bit rate quality with neural speech synthesis," in *Proc. INTERSPEECH*, 2019.

[8] E. Cohen, F. Kreuk, and J. Keshet, "Speech time-scale modification with gans," *IEEE Signal Processing Letters*, vol. 29, pp. 1067–1071, 2022.

[9] 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," *arXiv:1802.08435*, 2018.

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

[11] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[13] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[14] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17022–17033, 2020.

[15] M. Morrison, R. Kumar, K. Kumar, P. Seetharaman, A. Courville, and Y. Bengio, "Chunked autoregressive gan for conditional waveform synthesis," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[16] A. Mustafa, J.-M. Valin, J. Büthe, P. Smaragdis, and M. M. Goodwin, "Framewise wavegan: High speed adversarial vocoder in time domain with very low computational complexity," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
[17] B.C.J. Moore, *An introduction to the psychology of hearing*, Brill, 2012.
[18] W. Jang, D. Lim, J. Yoon, B. Kim, and J. Kim, "Univnet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation," in *Proc. INTERSPEECH*, 2021.
[19] J. Büthe, A. Mustafa, J.-M. Valin, K. Helwani, and M. M. Goodwin, "NoLACE:Improving Low-Complexity Speech Codec Enhancement Through Adaptive Temporal Shaping," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
[20] X. Mao, Q. Li, H. Xie, R. YK Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.
[21] I. Demirsahin, O. Kjartansson, A. Gutkin, and C. Rivera, "Open-source Multi-speaker Corpora of the English Accents in the British Isles," in *Proc. LREC*, 2020.
[22] O. Kjartansson, A. Gutkin, A. Butryna, I. Demirsahin, and C. Rivera, "Open-Source High Quality Speech Datasets for Basque, Catalan and Galician," in *Proc. SLTU and CCURL*, 2020.
[23] K. Sodimana, K. Pipatsrisawat, L. Ha, M. Jansche, O. Kjartansson, P. De Silva, and S. Sarin, "A Step-by-Step Process for Building TTS Voices Using Open Source Data and Framework for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese," in *Proc. SLTU*, 2018.
[24] A. Guevara-Rukoz, I. Demirsahin, F. He, S.-H. C. Chu, S. Sarin, K. Pipatsrisawat, A. Gutkin, A. Butryna, and O. Kjartansson, "Crowdsourcing Latin American Spanish for Low-Resource Text-to-Speech," in *Proc. LREC*, 2020.
[25] F. He, S.-H. C. Chu, O. Kjartansson, C. Rivera, A. Katanova, A. Gutkin, I. Demirsahin, C. Johny, M. Jansche, S. Sarin, and K. Pipatsrisawat, "Open-source Multi-speaker Speech Corpora for Building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu Speech Synthesis Systems," in *Proc. LREC*, 2020.
[26] Y. M. Oo, T. Wattanavekin, C. Li, P. De Silva, S. Sarin, K. Pipatsrisawat, M. Jansche, O. Kjartansson, and A. Gutkin, "Burmese Speech Corpus, Finite-State Text Normalization and Pronunciation Grammars with an Application to Text-to-Speech," in *Proc. LREC*, 2020.
[27] D. van Niekerk, C. van Heerden, M. Davel, N. Kleynhans, O. Kjartansson, M. Jansche, and L. Ha, "Rapid development of TTS corpora for four South African languages," in *Proc. INTERSPEECH*, 2017.
[28] A. Gutkin, I. Demirşahin, O. Kjartansson, C. Rivera, and K. Túbòsún, "Developing an Open-Source Corpus of Yoruba Speech," in *Proc. INTERSPEECH*, 2020.
[29] E. Bakhturina, V. Lavrukhin, B. Ginsburg, and Y. Zhang, "Hi-Fi Multi-Speaker English TTS Dataset," *arXiv preprint arXiv:2104.01497*, 2021.
[30] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario.," in *Proc. Interspeech*. 2011, pp. 1509–1512, ISCA.
[31] ITU-T, *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, 2001.
[32] Wissam A Jassim, Jan Skoglund, Michael Chinen, and Andrew Hines, "Warp-q: Quality prediction for generative neural speech codecs," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 401–405.
[33] ITU-T, *Recommendation P.808: Subjective evaluation of speech quality with a crowdsourcing approach*, 2018.