Real-Time Packet Loss Concealment With Mixed Generative and Predictive Model

Jean-Marc Valin[♮], *Ahmed Mustafa*[♮], *Christopher Montgomery*[♮], *Timothy B. Terriberry*[♮], *Michael Klingbeil*[♮], *Paris Smaragdis*[♮♭], *Arvindh Krishnaswamy*[♮]

[♮]Amazon Web Services [♭]University of Illinois at Urbana-Champaign

{jmvalin, ahdmust, chrimonh, territim, klingm, parsmara, arvindhk}@amazon.com

Abstract

As deep speech enhancement algorithms have recently demonstrated capabilities greatly surpassing their traditional counterparts for suppressing noise, reverberation and echo, attention is turning to the problem of packet loss concealment (PLC). PLC is a challenging task because it not only involves real-time speech synthesis, but also frequent transitions between the received audio and the synthesized concealment. We propose a hybrid neural PLC architecture where the missing speech is synthesized using a generative model conditioned using a predictive model. The resulting algorithm achieves natural concealment that surpasses the quality of existing conventional PLC algorithms and ranked second in the Interspeech 2022 PLC Challenge. We show that our solution not only works for uncompressed audio, but is also applicable to a modern speech codec.
Index Terms: packet loss concealment, LPCNet, Opus

1. Introduction

Real-time voice communication over the Internet typically relies on "best-effort" unreliable transmission protocols (RTP/UDP) to minimize latency. When voice packets are lost (or arrive too late to be played back), the receiver attempts to *conceal* the loss in such a way as to limit the quality degradation caused by the loss. Traditional packet loss concealment (PLC) techniques often involve repeating pitch periods [1], which improves quality over filling the missing packets with silence, but often also causes noticeable artifacts.

Recent techniques based on deep neural networks (DNN) have been shown to significantly improve quality for both noise suppression [2] and echo cancellation [3]. DNN techniques for neural PLC have also recently been investigated [4, 5, 6, 7, 8, 9] and in this work, we propose a new PLC algorithm that ranked second in the Interspeech 2022 Audio Deep Packet Loss Concealment Challenge [10].

We propose a hybrid architecture that combines both a generative vocoder and a predictive conditioning model (Section 2). During losses, the predictive model estimates the acoustic features to be used by the generative LPCNet [11] neural vocoder (Section 3). Synthesis is performed in the time domain and in Section 4 we proposed both causal and non-causal approaches to post-loss resynchronization. The evaluation in Section 5 demonstrates that the proposed real-time PLC algorithm significantly out-performs conventional PLC algorithms.

2. Neural Concealment Architecture

To conceal missing packets, we want to use some form of neural vocoder, with many options available, including GAN-based [12], flow-based [13], and autoregressive [14] techniques. Although not strictly necessary, it is desirable for the vocoder to be able to synthesize speech that perfectly aligns with the ground-truth speech before a loss, thereby avoiding the need for a cross-fade at the beginning of the loss. Also, considering that the challenge rules target CPU inference, highly-parallel synthesis is not required, making autoregressive vocoders a good choice. In this work, we use LPCNet [11].

Purely autoregressive models have been shown to successfully conceal packet loss, with various measures to counter the model's natural tendency to drift and babble over time [7]. That behavior is expected since speech is being synthesized with no conditioning beyond the start of the packet loss.

In this work, we propose a hybrid approach combining generative and predictive models. A generative autoregressive model is used to conceal the missing audio samples, but the acoustic features used for generation are determined using a predictive model. This allows precise long-term control of the spectral trajectory being reconstructed, while generating natural-sounding speech.

This can be viewed as different generation criteria for different time scales. While we want our algorithm to be "creative" in extending missing segments of a phoneme using plausible-sounding audio, we do *not* want an algorithm that generates more phonemes or words that have not been spoken – no matter how plausible.

2.1. LPCNet Model

LPCNet is an autoregressive neural vocoder that improves on WaveRNN [15] by using linear prediction. LPCNet is divided into a frame rate network and a sample rate network. The frame rate network, operating at 100 Hz on acoustic features, outputs conditioning parameters for the sample rate network that autoregressively generates the 16 kHz speech samples. In this work, we use the improved version of LPCNet with significantly reduced complexity [16].

LPCNet uses acoustic feature vectors based on 20-ms overlapping windows with a 10-ms interval. Each consists of 18 Bark frequency cepstral coefficients (BFCC), a pitch period, and a pitch correlation. Each 10-ms synthesized speech segment corresponds to the center of the analysis window.

The original LPCNet model uses two 3x1 convolutional layers in its frame rate network, configured in such a way as to use two feature vectors ahead of the frame being synthesized. This extra context improves the synthesis quality at the cost of 25 ms of added latency. For PLC, when a packet is lost, we do not have the future features and trying to predict them would result in more unreliable information being used for the synthesis. For that reason, we use a model with causal features. That being said, due to the overlap in the analysis windows, we still have a 5 ms algorithmic delay, as we synthesize the middle 10 ms of the 20-ms analysis window.

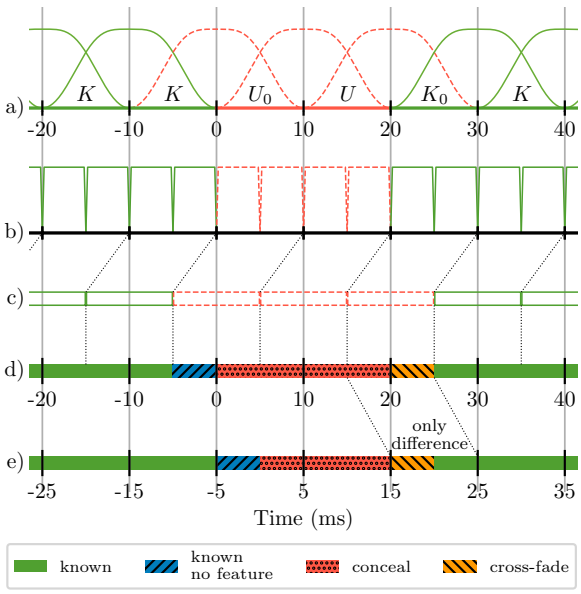

Figure 1: *Proposed PLC for a sequence where the samples at $t \in [0\,\text{ms}, 20\,\text{ms})$ are unknown (lost): a) feature analysis windows, with the dashed red windows denoting the features that cannot be computed due to missing data b) Burg analysis segments (Section 3.2), where the dashed red segments cannot be computed c) LPCNet input features, where the green features are a copy of the features in a) and the dashed red feature vectors are predicted by the DNN model d) output samples for causal processing (Section 4.1), where the green samples are known and used to update the LPCNet state with known features, the blue samples are known, but update the state using predicted features, the red samples are synthesized by LPCNet, and the orange samples are a cross-fade of synthesized and known samples e) non-causal processing (Section 4.2) illustrating the 5-ms delay and using the same colors, except that orange samples are a cross-fade of forward and backward synthesis. The tick marks still denote the processing intervals. Note the shorter total concealment needed for the non-causal case.*

3. Conditioning Prediction

LPCNet requires features at a 10-ms interval to condition the synthesis. When packets are lost, we cannot compute those features from the missing data, so we need to estimate them based on the received samples. The simplest approach is to just *freeze* the features during loss, i.e. use the last feature vector available. Unfortunately, when packet loss occurs during non-stationary periods, the concealment produces noticeable artifacts. For example, when a packet is lost at the end of a syllable, the last feature vector represents the energy centered 10 ms before the loss, and repeating it causes an audible burst in energy.

Instead, we use a DNN model to predict the unknown feature vectors from the known ones. To provide as much context for the prediction as possible, we use a recurrent neural network (RNN). For each frame, the input is either the known feature vector, or a vector of zeros. We include a "lost" binary flag to the input vector since the known features could theoretically be all zeros. The prediction network consists of one fully-connected layer, followed by two gated recurrent units (GRUs) [17], and a fully-connected output layer.

3.1. Perceptual Loss Functions

Because some aspects of the PLC task are completely unpredictable, we need a robust loss that treats unexpected/unpredictable events like label noise. For that reason, an L_1 loss function is more appropriate than an L_2 loss. We find that we can further improve quality by using a different loss for the different predicted features.

For the N_c cepstral features, we find that in addition to an L_1 loss in the cepstral domain we can benefit from considering the bands themselves. Moreover, we find that overestimating the energy of voiced frames hurts quality more than underestimating it, so we adaptively bias the spectral loss:

$$\mathcal{L}_s = \mathbb{E}\left[\sum_{k=0}^{N_c-1} \left(|\Delta c_k| + |\Delta b_k| + \alpha \max(\Delta b_k, 0)\right)\right], \quad (1)$$

where $\Delta c_k = \hat{c}_k - c_k$ is the difference between the predicted and ground truth cepstral coefficients, Δb_k is the difference in band energies computed from the inverse discrete cosine transform (IDCT) of Δc_k, and α is a bias coefficient equal to 1 for voiced frames and 0 for unvoiced frames.

Because the pitch period is always included in the features (even for unvoiced frames), it is particularly noisy and its loss function must be particularly robust to outliers. We use

$$\mathcal{L}_p = \mathbb{E}\left[|\Delta p| + \beta_1 \min\left(|\Delta p|, 50\right) + \beta_2 \min\left(|\Delta p|, 20\right)\right], \quad (2)$$

where $\Delta p = \hat{p} - p$ is the difference between the predicted and ground truth pitch period and $\beta_\times$ are tuning parameters (we use $\beta_1 = 20$ and $\beta_2 = 160$).

We find that overestimating the pitch correlation feature r leads to better pitch stability so we use the biased loss

$$\mathcal{L}_c = \mathbb{E}\left[|\Delta_r| + 2\max\left(-\Delta_r, 0\right)\right]. \quad (3)$$

3.2. Improved Temporal Resolution

One limitation with the LPCNet features in the context of PLC is that the 20-ms analysis window is centered 10 ms before the concealment begins (Fig. 1a), so it does not always capture the changes that occur just before a loss. To address the problem, we introduce additional features based on short-term spectral analysis. Burg spectral estimation [18] is well-suited for such analysis since it does not require windowing. We estimate the Burg spectrum independently on each 5-ms half of each new frame, as shown in Fig. 1b. The Burg-derived all-pole filters are then converted into cepstral coefficients that supplement the input to the feature prediction DNN (Fig. 1c). We can also see from Fig. 1 that the loss of N_L frames results in the loss of $(N_L + 1)$ LPCNet feature vectors, but only N_L Burg feature vectors, which is another benefit of the Burg analysis proposed here.

3.3. Handling Long Bursts

During long bursts of lost packets, it is meaningless to attempt concealment beyond around 100 ms. In that case, it is desirable to fade the concealment in the least objectionable way. Fading too slowly results in artifacts that sound similar to heavy breathing, whereas fading too quickly sounds very unnatural. We choose to fade out in such a way as to match the reverberation time of a small room with $\text{RT}_{60} = 120\,\text{ms}$. This can be accomplished, at inference time, by linearly decreasing the first predicted cepstral coefficient (c_0) over time after the initial 100 ms. As a result, long losses sound similar to a talker being naturally interrupted.

4. Framing, Transitions & Resynchronization

Unlike many other applications of neural vocoders, the PLC task involves using both known and synthesized audio in its output. It is thus important to make the transitions as seamless as possible. We consider four types of frames: known frames (K), transition unknown/lost frames at the beginning of a loss (U_0), unknown frames (U), and transition known frames after a loss (K_0). Each frame type requires different processing. Also, in this work, we propose three ways of performing the concealment: causal, non-causal, and inside a stateful codec.

4.1. Causal

The causal concealment case, with no look-ahead allowed, is depicted in Fig. 1a-1d. When no loss is occurring, the incoming K frame containing samples $[t-10\,\mathrm{ms}, t)$ is used to compute both the LPCNet features from the window covering $[t-20\,\mathrm{ms}, t)$, as well as the two Burg features in $[t-10\,\mathrm{ms}, t)$. The features are used to update the prediction DNN state (the output is discarded). The LPCNet features are also used alongside samples $[t-15\,\mathrm{ms}, t-5\,\mathrm{ms})$ to update the state of the LPCNet vocoder, meaning that the samples are given to the network as input, but the output probability is discarded.

Upon the first loss, we need to predict two LPCNet feature vectors. The first is used to continue updating the LPCNet state using known samples up to time t and then to synthesize the U_0 samples up to $t+5\,\mathrm{ms}$. The second predicted vector is used to synthesize samples $[t+5\,\mathrm{ms}, t+10\,\mathrm{ms})$. For any further loss only one new vector is predicted and the first half of the U samples are synthesized using the "old" feature vector and the second half is synthesized using the new vector.

On the first new packet after a loss, the samples at $[t+20\,\mathrm{ms}, t+30\,\mathrm{ms})$ (in our example) cannot be used to compute LPCNet features (since the first half of the window is lost), but can be used for Burg features. In turn, the Burg features can be used to update the latest prediction from the DNN. That prediction is used to continue the synthesis up to $t+25\,\mathrm{ms}$, cross-fading these generated samples with the K_0 samples to avoid any discontinuity.

4.2. Non-causal

In the non-causal PLC case (Fig. 1e), we improve the concealment quality by using 5 ms of look-ahead in the processing. When packets are continuously received, the processing is the same as the causal case, except that the algorithm delays the output signal by 5 ms to keep its look-ahead buffer. Similarly, there are no fundamental changes during loss. The improvement comes from the resynchronization after a loss. When the first packet arrives after a loss, we use the 10 ms of speech in the packet to extrapolate the speech backwards in time for 5 ms. Because of the delay, that 5 ms segment has not yet been played, so we can cross-fade the backward and forward signal extensions. By doing so, we are able to play the all received audio packets unmodified, unlike in the causal case.

4.3. Stateful codec

Both the causal and non-causal PLC cases apply to situations where audio is either uncompressed, or compressed with a simple stateless codec such as G.711 [19]. When speech is compressed using a stateful codec such as Opus [20] or AMR-WB [21], the concealed audio is needed to reconstruct the first packet after a loss, so the non-causal concealment proposed above is not possible. Moreover, because the linear prediction used in most modern codecs inherently avoids discontinuities, the cross-fade step from the causal case is not needed. Aside from that, the rest of Fig. 1d also holds for the stateful codec case. In this work, we demonstrate a neural PLC operating inside the voice coding mode of Opus [22]. In that context, our proposed PLC algorithm completely replaces the existing SILK PLC and its output is also used for prediction (long- and short-term) in the decoder state when a new packet arrives.

Table 1: *Computation time (in ms) to process 10 ms of speech for each type of frame, as measured on an Intel i7-10810U laptop CPU.*

Algorithm	K	U_0	U	K_0
Causal	1.35	2.12	1.34	1.53
Non-Causal	1.38*	1.33	1.34	2.58
Codec	1.38	2.18	1.37	0.84

*2.54 ms for the first frame following a K_0 frame

5. Evaluation & Results

The LPCNet vocoder model is trained on 205 hours of 16-kHz speech from a combination of TTS datasets [23, 24, 25, 26, 27, 28, 29, 30, 31] including more than 900 speakers in 34 languages and dialects. The training is performed as described in [16], except that we explicitly randomize the sign of each training sequence so that the algorithm works for any polarity of the speech signal. Due to recent improvements to the LPCNet efficiency, we are able to use a $\mathrm{GRU_A}$ size of 640 units, with 15% density, while meeting all real-time constraints.

The feature prediction model is trained on all 205 hours used for the vocoder, plus 64 hours of training speech published by the PLC challenge organizers. The size of the GRUs is set to 512 units, with a 256-unit input fully-connected layer. The source code for the proposed PLC is available under an open-source license at `https://github.com/xiph/LPCNet` in the `plc_challenge` branch, with the corresponding Opus neural PLC at `https://gitlab.xiph.org/xiph/opus` in the `neural_plc` branch.

Since the PLC challenge allows for up to 20 ms total latency and does not involve a codec, we submitted the non-causal PLC for consideration. Despite that, we still evaluate all use cases in this section.

5.1. Complexity

The complexity of the algorithm is dominated by the complexity of the LPCNet vocoder, with the feature prediction contributing less than 20% of the total complexity. The processing time for each 10-ms frame varies over time according to the number of samples processed by the vocoder based on the loss pattern and the use-case considered in Section 4. The measured processing times are shown in Table 1. In steady-state operation (either K or U), the algorithm requires between 1.3 and 1.4 ms to process 10 ms of speech, which amounts to between 13% and 14% of one CPU core required for real-time operation. The worst-case computation time for the non-causal case is 2.58 ms. When considering the 10-ms frame size and the 5-ms look-ahead, the total delay of the non-causal PLC is 17.58 ms, meeting the 20-ms maximum allowed for the challenge.

Table 2: *PLC Challenge official results. The overall score only considers the subjective quality evaluation (CMOS) and the speech recognition word accuracy (WAcc). The proposed algorithm ranks first for word accuracy and second for both CMOS and overall score. We provide the scores for the first overall submission and the average of the two submissions tied for third place. The PLCMOS [10] and DNSMOS [35] objective metrics are included only for reference and are not considered in the rankings.*

Algorithm	PLC-MOS	DNS-MOS	**CMOS**	**WAcc**	**Score**
1^{st} place	4.282	3.797	**-0.552**	0.875	**0.845**
proposed	3.744	3.788	-0.638	**0.882**	0.835
3^{rd} avg.	3.903	3.686	-0.825	0.864	0.794
Zero-fill	2.904	3.444	-1.231	0.861	0.725

Table 3: *Ablation study results. Compared to feature repetition, we show the improvement from adding a feature prediction DNN, adding perceptual loss functions (Section 3.1), adding Burg spectral features (Section 3.2), and adding 5 ms non-causal processing (Section 4.2).*

Algorithm	PESQ-WB	PLC-MOS
Zero-fill	2.185	2.874
Baseline	2.059	2.786
Repetition	2.517	3.642
+DNN (causal)	2.647	3.688
+loss (1)-(3)	2.652	3.660
+Burg	2.705	3.739
+non-causal	**2.766**	3.790

5.2. Quality

The PLC challenge organizers [10] evaluated *blind* test utterances processed with the non-causal PLC algorithm. The comparison category rating (CCR) mean opinion score (CMOS) [32] results in Table 2 were obtained using the crowdsourcing methodology described in P.808 [33, 34], where 5 randomly-selected listeners were asked to evaluate each of the 966 test utterances. The 95% confidence intervals on the CMOS values are around 0.035. Among the 7 submissions to the challenge, the proposed algorithm ranked second in the CMOS evaluation and first on the speech recognition word accuracy evaluation (over the same test set), taking the second place overall.

To assess the contribution of different components of the algorithm, we perform an ablation study using the PESQ-WB [36] and PLCMOS [10] objective metrics. Although PESQ-WB has been shown in the past not to be suitable for vocoder evaluation, we have found that for neural PLC, it correlates sufficiently well with subjective evaluation. Table 3 shows an ablation study demonstrating the main improvements described in this work. We note that both metrics agree on all improvements, except the (1)-(3) losses, but we have found through informal listening that the losses are indeed useful for improving quality.

In addition to the official challenge experiments, we conducted further experiments on the challenge development set (because the clean samples were needed). Those experiments used the same models used for our challenge submission. In that experiment, we evaluate the causal, non-causal, and Opus codec cases. To evaluate the Opus PLC, we encoded the clean utterances at 24 kb/s using the speech mode with 20-ms frames, with the encoder robustness settings optimized for 20% loss (see Section 3.8 of [22]). We compare the proposed Opus neural PLC to the default classical PLC. Similarly, we include the NetEQ[1] classical PLC algorithm operating on (uncompressed) PCM audio.

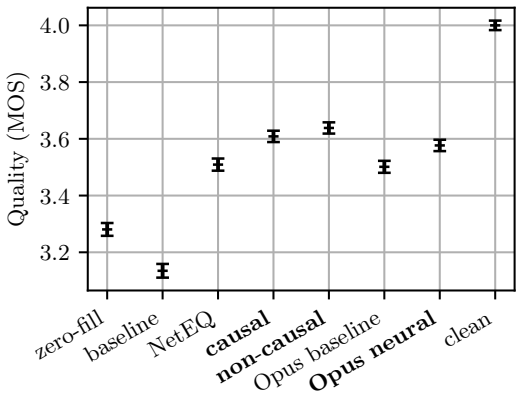

Figure 2: *Internal MOS results, including the 95% confidence intervals.*

Each of the 966 development utterances was evaluated by 15 randomly-selected listeners on a MOS absolute category rating (ACR) scale. The results in Fig. 2 show that for PCM, the causal and non-causal versions both out-perform the popular NetEQ PLC and the baseline PLC. While the difference between the non-causal and causal algorithm is not statistically significant, the objective results in Table 3 suggest that non-causal concealment improves slightly on the causal case. In the context of Opus-coded speech, the proposed PLC significantly out-performs the existing Opus PLC.

6. Conclusions

We have demonstrated that by splitting packet loss concealment into a feature-wise prediction model and a sample-wise generative model, we can achieve high-quality concealment that surpasses existing conventional PLC algorithms. The proposed solution operates in real-time on 10-ms frames, with or without look-ahead, using less than 14% of a laptop CPU core. We have also demonstrated that our work can be applied in the context of a modern speech compression algorithm and also out-perform conventional concealment.

7. References

[1] H. Sanneck, A. Stenger, K. Younes, and B. Girod, "A new technique for audio packet loss concealment," in *Proc. GLOBECOM*, 1996, pp. 48–52.

[2] C. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun, P. Rana, S. Srinivasan, and J. Gehrke, "The INTERSPEECH 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," *arXiv preprint arXiv:2005.13981*, 2020.

[3] K. Sridhar, R. Cutler, A. Saabas, T. Parnamaa, M. Loide, H. Gamper, S. Braun, R. Aichner, and S. Srinivasan, "ICASSP 2021 acoustic echo cancellation challenge: Datasets, testing framework, and results," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[1] Based on the WebRTC implementation at https://webrtc.org/

[4] B.-K. Lee and J.-H. Chang, "Packet loss concealment based on deep neural networks for digital speech transmission," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 378–387, 2015.

[5] R. Lotfidereshgi and P. Gournay, "Speech prediction using an adaptive recurrent neural network with application to packet loss concealment," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5394–5398.

[6] M. Mohamed and B. Schuller, "Concealnet: An end-to-end neural network for packet loss concealment in deep speech emotion recognition," *arXiv preprint arXiv:2005.07777*, 2020.

[7] F. Stimberg, A. Narest, A. Bazzica, L. Kolmodin, P. González, O. Sharonova, H. Lundin, and T. Walters, "Waveneteq – packet loss concealment with wavernn," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 672–676.

[8] J. Lin, Y. Wang, K. Kalgaonkar, G. Keren, D. Zhang, and C. Fuegen, "A time-domain convolutional recurrent network for packet loss concealment," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[9] S. Pascual, J. Serrà, and J. Pons, "Adversarial auto-encoding for packet loss concealment," *arXiv preprint arXiv:2107.03100*, 2021.

[10] L. Diener, S. Sootla, S. Branets, A. Saabas, R. Aichner, and R. Cutler, "INTERSPEECH 2022 audio deep packet loss concealment challenge."

[11] 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.

[12] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *Proc. ICLR*, 2019.

[13] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[14] 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.

[15] 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.

[16] J.-M. Valin, U. Isik, P. Smaragdis, and A. Krishnaswamy, "Neural speech synthesis on a shoestring: Improving the efficiency of lpcnet," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[17] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proceedings of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

[18] J. P. Burg, "Maximum entropy spectral analysis," Ph.D. dissertation, Stanford University, 1975. [Online]. Available: http://sepwww.stanford.edu/data/media/public/oldreports/sep06/

[19] ITU-T, *Recommendation G.711 : Pulse code modulation (PCM) of voice frequencies*, 1988.

[20] J.-M. Valin, K. Vos, and T. B. Terriberry, "Definition of the Opus Audio Codec," RFC 6716, Internet Engineering Task Force, Sep. 2012, https://tools.ietf.org/html/rfc6716.

[21] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen, "The adaptive multirate wideband speech codec (AMR-WB)," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 8, pp. 620–636, 2002.

[22] K. Vos, K. V. Sorensen, S. S. Jensen, and J.-M. Valin, "Voice coding with Opus," in *Proceedings of the 135th AES Convention*, 2013.

[23] 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. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.804

[24] 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. [Online]. Available: https://www.aclweb.org/anthology/2020.sltu-1.3

[25] K. Sodimana, K. Pipatsrisawat, L. Ha, M. Jansche, O. Kjartansson, P. D. 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. [Online]. Available: http://dx.doi.org/10.21437/SLTU.2018-14

[26] 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. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.801

[27] 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. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.800

[28] 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. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.777

[29] 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. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1139

[30] A. Gutkin, I. Demirşahin, O. Kjartansson, C. Rivera, and K. Túbòsún, "Developing an Open-Source Corpus of Yoruba Speech," in *Proc. INTERSPEECH*, 2020. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1096

[31] E. Bakhturina, V. Lavrukhin, B. Ginsburg, and Y. Zhang, "Hi-Fi Multi-Speaker English TTS Dataset," *arXiv preprint arXiv:2104.01497*, 2021.

[32] ITU-T, *Recommendation P.800: Methods for subjective determination of transmission quality*, 1996.

[33] ——, *Recommendation P.808: Subjective evaluation of speech quality with a crowdsourcing approach*, 2018.

[34] B. Naderi and R. Cutler, "An open source implementation of ITU-T recommendation P.808 with validation," in *Proc. INTERSPEECH*, 2020.

[35] C. K. Reddy, V. Gopal, and R. Cutler, "DNSMOS P.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[36] ITU-T, "P.862.2: Wideband extension to recommendation P.862 for the assessment of wideband telephone networks and speech codecs (PESQ-WB)," 2005.