ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747419

ARCHITECTURE FOR VARIABLE BITRATE NEURAL SPEECH CODEC WITH CONFIGURABLE COMPUTATION COMPLEXITY

Tejas Jayashankar[1,*], *Thilo Koehler*[2], *Kaustubh Kalgaonkar*[2], *Zhiping Xiu*[2]
Jilong Wu[2], *Ju Lin*[2], *Prabhav Agrawal*[2], *Qing He*[2,†]

[1] Massachusetts Institute of Technology [2] Facebook AI

ABSTRACT

Low bitrate speech codecs have become an area of intense research. Traditional speech codecs, which use signal processing methods to encode and decode speech, often suffer from quality issues at low bitrates. A neural speech codec, which uses a deep neural network in the compression pipeline, can help alleviate this issue. In this paper we present a new neural speech codec that: 1) supports variable bitrates 2) supports packet losses of up to 120 ms and 3) can operate at low-compute and high-compute modes. Our codec uses a hierarchical VQ-VAE (HVQVAE) for encoding and decoding spectral features at different bitrates. The decoded features are fed to a vocoder for speech synthesis. Depending upon the end user's computing resources, the decoder either uses a powerful WaveRNN or a parametric vocoder for speech synthesis. Our experiments demonstrate that our HVQVAE + WaveRNN setup achieves high audio quality.

***Index Terms*—** Speech codec, VQ-VAE, WaveRNN, Packet loss, Variable Rate

1. INTRODUCTION

In recent years the area of low bitrate speech coding has reemerged as a popular field of research. Low bitrate speech, with data rates under 16 kbps, is often desired in low bandwidth settings where the sender and/or receiver are/is constrained by the amount of information that can be transmitted between one another. For example, with video calls becoming a standard mode of communication in recent years, low bitrate speech codecs will serve as an important component to decrease the communication overhead while also reducing latency.

Traditional speech codecs often include a provision to encode speech at low bitrates, but the resulting decoded speech sounds mechanical and degraded. Vocoders such as linear predictive coders (LPC) [1, 2] and harmonic sinudoidal coders [3, 4] often work better than waveform coders such as PCM in low bitrate settings. For example, Codec2 [4] uses harmonic sinusoidal coding to compress speech at bitrates between 450 bps and 3.2 kbps. Hybrid coders, such as Opus [5], encode speech using a mixed waveform-voice coding approach, but can only achieve high audio quality at bitrates above 6 kbps.

Recently, neural speech codecs which use deep learning techniques to encode and decode speech, have been introduced to address the limitations of traditional speech codecs. For example, [6] improves Opus low bitrate speech quality by using an LPCNet [7] and WaveNet [8] for speech synthesis. There are also methods that encode the waveform directly into a discrete latent space using a VQ-VAE [9]. For example, [10] trains a VQ-VAE encoder and a WaveNet decoder end-to-end to synthesize high quality speech. Lyra [11] follows a different approach and trains a WaveRNN [12] to synthesize speech from KLT compressed spectrogram features at 3 kbps. More recently, methods that use VQ-VAEs to directly encode the waveform have also been introduced. SoundStream [13] is one such model that uses a wav to wav architecture to decode speech at different bitrates. Other architectures such as [14, 15] use quantized features from different layers of an autoencoding network to code speech at different bitrates.

In this paper, we propose a variable bitrate neural speech codec which we choose to operate at 3.2, 6.4 and 12.8 kbps. Note that the specific rates we experiment with are not required and the same architecture can be used to achieve other bitrates too. The codec chooses a suitable bitrate based on the bandwidth that is allocated for transmission. We use a hierarchical VQ-VAE (HVQVAE) architecture that compresses a spectral representation of the audio at different bitrates followed by a vocoder that synthesizes audio from the decoded spectral representation. Furthermore, our model is trained to support packet loss of up to 120 ms during transmission and decoding. If the receiver has sufficient computing resources, the decoder uses a powerful WaveRNN vocoder for synthesis. If the receiver's device is resource constrained, a parametric vocoder is used instead.

The paper is organized as follows: In Section 2 we detail our proposed codec with an explanation of our hierarchical VQ-VAE and WaveRNN decoder. In Section 3 we provide architecture details and training details. We finally close with a results section that demonstrates that our codec is able to decode audio with high quality by comparing against various baselines through a Mean Opinion Scores (MOS) study.

2. PROPOSED CODEC

In this paper we study speech compression on 16 kHz audio. The main components of our codec are the variable bitrate hierarchical VQ-VAE and the configurable decoder that uses a WaveRNN or parametric vocoder for speech synthesis. In this section we provide more details about these components and also describe how our model is capable of handling packet loss during transmission.

2.1. System Overview

Our codec operates by first extracting spectral features from the input speech. These features are subsequently fed to a hierarchical VQ-VAE (HVQVAE) that encodes the features into a discrete latent space before being converted into a bitstream or sequence of packets. Our codec uses 40 ms packets during transmission and the HVQVAE operates on 5 consecutive packets, i.e., 200 ms of speech.

[*] Work performed during internship at Facebook AI
[†] Corresponding author

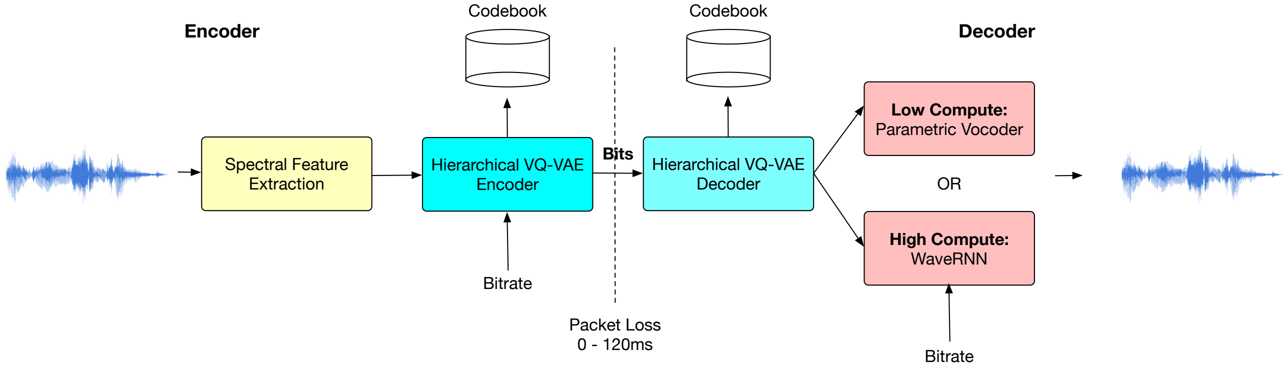

Encoder
Codebook
Codebook
Decoder
Spectral Feature Extraction
Hierarchical VQ-VAE Encoder
Bits
Hierarchical VQ-VAE Decoder
Low Compute: Parametric Vocoder
OR
High Compute: WaveRNN
Bitrate
Packet Loss 0 - 120ms
Bitrate

Fig. 1: Overall system architecture for our codec. A spectrogram representation of the input speech is fed to a HVQVAE in the encoder. The HVQVAE encoder sends a discrete latent representation of the spectrogram to the decoder via a bitstream. The codec decoder first reconstructs the spectrogram using the HVQVAE decoder before synthesizing the speech. If the user device has low-compute the decoder uses a parametric vocoder. If the user device has high-compute the decoder uses a WaveRNN.

The encoder is also provided with a bitrate at which the HVQVAE should operate. The packets are received by the HVQVAE decoder which reconstructs the spectral features. The latter reconstruction is finally sent to a WaveRNN or parametric vocoder along with bitrate information to synthesize the speech. To handle packet loss, the decoder of the HVQVAE is trained to reconstruct the spectral features by masking random sequence of packets during training.

2.2. Variable Bitrate Hierarchical VQ-VAE

The design of our HVQVAE, as seen in Figure 2, is based on the architecture in [16, 17]. The model operates on 200 ms of speech and uses as input a D×20 spectral representation of the speech, where D is the dimensionality of the spectral features. The input is downsampled spatially to different sizes using an encoder block to achieve various bitrates. For example, at layer 1, the input is downsampled spatially to a size of 32 × 10 with 256 channels, $\boldsymbol{z_e} \in \mathbb{R}^{256\times32\times10}$. It is then quantized using an 8-bit codebook, $\boldsymbol{e} \in \mathbb{R}^{256\times256}$. The overall bitrate at this layer is 32×10×8 = 2560 bits per 200 ms, or 12.8 kbps. A decoder block reconstructs the spectrogram from the quantized features. To achieve lower bitrates we can further encode this latent map to smaller spatial sizes. We use an 8-bit codebook at each layer, with increasing feature channels at higher layers.

The HVQVAE is trained layer by layer. When training layer n all the encoder and decoder weights for layers $i < n$ are frozen. This is done to incrementally train the model while also ensuring that layer $n-1$ can be used independently of layer n. Let the input to the encoder block at layer i be $\boldsymbol{z_e^{i-1}}$, the output of the same encoder block be $\boldsymbol{z_e^i}$ and let $\boldsymbol{\tilde{z_e^i}}$ be the quantized embeddings obtained by a nearest neighbor mapping of $\boldsymbol{z_e^i}$ in the latent space $\boldsymbol{e}$. Finally, let the output from the decoder at layer i be $\boldsymbol{\hat{z_e^{i-1}}}$. Then the HVQVAE loss function at layer i is

$$\mathcal{L} = \|\boldsymbol{\hat{z_e^{i-1}}} - \boldsymbol{z_e^{i-1}}\|_2^2 + \beta\|\text{sg}[\boldsymbol{\tilde{z_e^i}}] - \boldsymbol{z_e^i}\|_2^2 \\ + \|\boldsymbol{\tilde{z_e^i}} - \text{sg}[\boldsymbol{z_e^i}]\|_2^2, \tag{1}$$

where sg[·] denotes the stop gradient operator and β is the commitment cost. The first term is the reconstruction loss, the second term ensures that the quantized emebeddings are close to the original features and the third term is the VQ loss. The last term is optimized using exponential moving average (EMA) k-means.

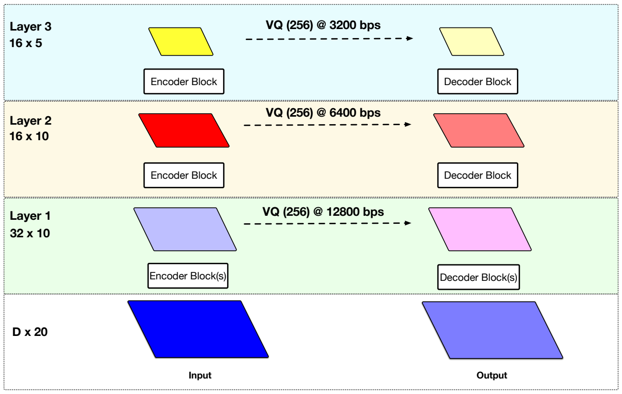


Fig. 2: Hierarchical VQ-VAE architecture. Depending on the desired coding bitrate, the VQ-VAE encodes and decodes the spectral features at different levels. VQ(n) denotes the number of vectors, n, in the vector quantizer codebook.

2.3. Audio Synthesis with WaveRNN/Parametric Vocoder

Once the spectral features have been reconstructed, the final step is to synthesize the speech as shown in Figure 1. If the receiver's device has high-compute capabilities, our codec uses a WaveRNN vocoder [12] for synthesis. The WaveRNN consists of an upsampling ResNet followed by a GRU to decode the audio sample-by-sample. Our WaveRNN model uses a categorical softmax as the final layer to output an 8-bit audio sample. The WaveRNN takes as input binned log magnitude spectrogram features along with a one-hot vector representing the bitrate. While training the WaveRNN, we uniformly sample a decoded spectrogram representation from the available bitrates. This way the WaveRNN learns to decode speech at multiple bitrates within the same training epoch.

If the receiver has a low-compute device, our codec uses a parametric vocoder based on a modified version of the Griffin-Lim vocoder [18] for synthesis. The parametric vocoder takes as input 48-dimensional cepstral features, 5-dimensional periodicity features and f0 (pitch) information. Since this is an algorithm, the parametric vocoder does not need to be trained.

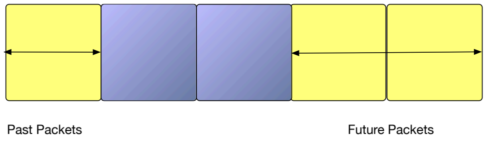


Fig. 3: An example of 80 ms packet loss simulation during training of the topmost layer of the HVQVAE. In this setting, packet loss is simulated by masking out (purple) any two consecutive frames in the 16×5 bottleneck feature. The HVQVAE learns how to reconstruct the input spectrogram by making use of past and future packet information (yellow). In our experiments the models are trained with 40 ms, 80 ms and 120 ms of packet loss.

2.4. Simulating Packet Loss

In order to simulate packet loss during transmission, we train our HVQVAE with masking, an example of which is shown in Figure 3. In general, at layer i, with encoded spatial feature of size m×n, we can simulate $40k$ ms of packet loss by masking out nk/5 packets, where each m×1 feature is considered a packet. We randomize the position of the masked packets so that the model can learn to make use of varying lengths of past and future context to reconstruct the spectral features. This also helps the model learn disentangled feature representations at the bottleneck layers. While training the HVQVAE we simulate packet loss 25% of the time by uniformly choosing between 40 ms, 80 ms and 120 ms of masking.

3. EXPERIMENTS

This section details the dataset, the specifics of the systems architecture and the training pipeline. We close by presenting our results with comparisons against various baselines via a Mean Opinion Scores (MOS) study [1].

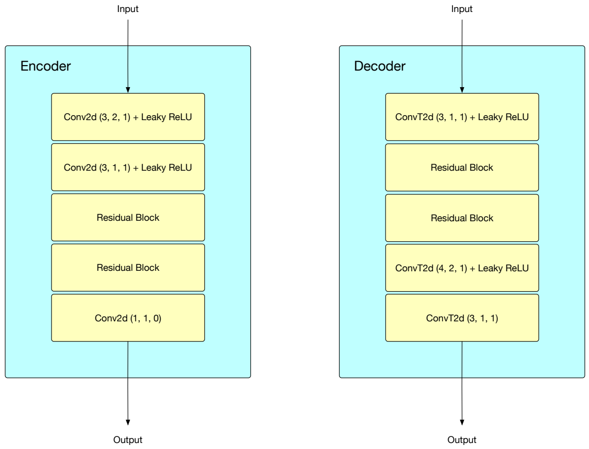
3.1. Dataset

Our models are trained using the dataset from the DNS challenge [19]. We use ∼580 hours of English clean-speech audio sampled at 16 kHz. The dataset is preprocessed by splitting the audio clips into 10-second segments. Our training set consists of around 248698 samples and our validation set consists of 12694 samples. We also have a held-out test set consisting of 150 samples from DNS development set. To compute the log spectrograms for the WaveRNN decoder, we compute a 512-STFT using a 20 ms window size and a 10 ms hop size, resulting in a spectrogram of shape 257×T (since the FFT is real and symmetric). We use a Hanning window and take the log-magnitude of the resulting spectrogram. We quasi-uniformly bin the 257 frequencies into $D = 80$ bins, resulting in a binned-spectrogram of shape 80×T. This spectrogram is split into $\lfloor T/20 \rfloor$ 80×20 frames for input to the HVQVAE.

3.2. Architecture and Training Details

3.2.1. Hierarchical VQ-VAE

The encoder and decoder block architectures are shown in Figure 4. Layer 1 of the HVQVAE has additional encoder blocks to reduce the dimensionality of the D×20 input to 32×20. The residual block uses two 3×3 convolutional layers with no batch normalization. The number of output channels at each layer of the HVQVAE is [256, 512, 1024] on the encoder side and [1024, 512, 256] on the decoder side.



(a) Encoder Block

(b) Decoder Block

Fig. 4: Encoder and decoder blocks architecture details. ConvT stands for transposed convolutions and convolutions are denoted by Conv(kernel size, stride, padding).

While training layer n the weights of all the layers $i < n$ are frozen. We use learning rates of 1e-4, 2e-5 and 4e-6 while training layers 1, 2 and 3 respectively. Each layer is trained with the Adam optimizer [20] for 500k steps on 32 V100 GPUs. We also gradually increase the commitment cost β from 0 at step 0 to 0.25 at step 30k to prevent the codebook vectors collapsing to a fixed vector.

3.2.2. HVQVAE + WaveRNN

The WaveRNN consists of two main components — the feature ResNet and the GRU. The feature ResNet has 10 residual blocks, each which use two 1x1 1-D convolutions, batch normalization and ReLU activation functions. The feature ResNet is followed by a few stretch layers that upsample the features to 16 kHz. In our experiments the spectrograms are sampled at 100 Hz and we use three stretch layers with upsampling factors of 4, 5 and 8 respectively. The ResNet uses 128 channels in all intermediate layers and 256 channels at the output. The upsampled features are then concatenated with the ground truth audio (right shifted by one sample) and fed to the GRU. The GRU uses 1024 channels. We use teacher forcing to train the GRU, i.e., the ground truth sample at time step t is fed to the GRU cell at $t + 1$ instead of the predicted sample at the previosu time step.

The HVQVAE is frozen while training the WaveRNN. The bitrate for decoding is chosen uniformly at random. The decoded spectrogram is concatenated with a one-hot vector representing the bitrate resulting in an input of shape 84×20. The model is trained with the AdamW optimizer [21] with a learning rate of 1e-4 for 2M steps on 32 V100 GPUs. We use a batch size of 128 and a linear decay learning rate schedule.

3.3. Evaluation Metric

We measure the performance of our models using Mean Opinion Scores (MOS). We selected a total of 30 sentences and each sentence was rated by about 300 raters. The listeners were asked to rate each sentence on a scale of 1 to 5.

[1] Audio samples: https://variable-neural-audio-codec.github.io/samples/

System	Bitrate	MOS
Groundtruth	-	4.06 ± 0.046
Opus [5]	6 kbps	3.79 ± 0.057
Codec2 [4]	3.2 kbps	3.76 ± 0.054
Lyra [11]	3.0 kbps	3.88 ± 0.054
(Ours) Single rate VQ-VAE + WaveRNN	3.2 kbps	3.99 ± 0.051
(Ours) HVQVAE + Parametric Vocoder	3.2 kbps	3.78 ± 0.061
(Ours) HVQVAE + WaveRNN	3.2 kbps	3.81 ± 0.057

Table 1: MOS Study 1: Comparison of our models operating at a bitrate of 3.2 kbps against various baselines. Our singlerate model achieves the highest MOS and our variable rate WaveRNN model operating at the same bitrate achieves a MOS score similar to Lyra.

System	Bitrate	MOS
Groundtruth	-	4.07 ± 0.048
HVQVAE + Parametric Vocoder	3.2 kbps	3.65 ± 0.071
HVQVAE + WaveRNN	3.2 kbps	3.63 ± 0.067
HVQVAE + Parametric Vocoder	6.4 kbps	3.79 ± 0.066
HVQVAE + WaveRNN	6.4 kbps	3.82 ± 0.058
HVQVAE + Parametric Vocoder	12.8 kbps	3.86 ± 0.059
HVQVAE + WaveRNN	12.8 kbps	3.91 ± 0.054

Table 2: MOS Study 2: Comparison of our models across different bitrates. The speech quality improves as the bitrate increases, with larger gains in speech quality when using a WaveRNN for decoding at higher bitrates.

3.4. Results

3.4.1. Comparing HVQVAE reconstructions across bitrates

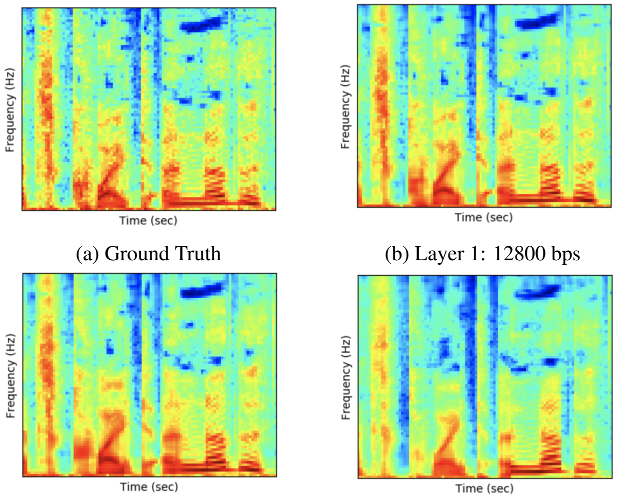
Figure 5 shows the spectrogram reconstruction from different layers of the HVQVAE. As is seen, the decoded spectrograms look slightly more smoothened as the bitrate decreases which is evident from the frequency spread in the high frequency range. Moreover, the unvoiced regions (blue) look more denoised as the bitrate decreases. It is due to this differing reconstruction quality that the variable bitrate model has slightly degraded quality in comparison to our singlerate model as discussed below in MOS study 1.

3.4.2. MOS Study 1: Comparison with baselines

We first test our VQ-VAE + WaveRNN setup by training a singlerate model at 3.2 kbps. As shown in Table 1, this setup achieves the best MOS score and even achieves higher quality than Lyra. The variable rate HVQVAE + WaveRNN setup does cause a quality degradation but still outperforms Codec2@3.2 kbps and Opus@6 kbps while achieving speech quality comparable to Lyra. Our low-compute model that uses a parametric vocoder is able to outperform Codec2 and produce speech quality on par with Opus as it makes use of the high fidelity HVQVAE reconstructed spectral features.

3.4.3. MOS Study 2: Comparison across bitrates

We perform another MOS study to compare our model across different bitrates as shown in Table 2. As expected, our models demonstrate better speech quality as the bitrate increases. At 3.2 kbps both the parametric vocoder and WaveRNN exhibit similar quality but as the bitrate increases the higher synthesis quality of the autoregressive WaveRNN over the parametric decoder is more evident.



(a) Ground Truth

(b) Layer 1: 12800 bps

(c) Layer 2: 6400 bps

(d) Layer 3: 3200 bps

Fig. 5: Example spectrogram reconstructions of a 1 second speech sample from the different layers of the HVQVAE. As the bitrate decreases, the HVQVAE starts to smooth out the frequency content as seen in the 3200 bps reconstruction in (d).

3.4.4. MOS Study 3: Comparison with and without PLC

We perform one last MOS study to evaluate decoded speech quality in the presence and absence of packet loss. We trim our samples down to 1.5 seconds and pass it through our codec with and without 120 ms of packet loss. The MOS score difference between no packet loss and 120 ms of packet loss are 0.13, 0.05, 0.07 for the parametric vocoder; 0.02, 0.01, -0.01 for the WaveRNN, for bitrates 3.2 kbps, 6.4 kbps and 12.8 kbps respectively. Since the HVQVAE was trained to simulate packet loss via masking, there is almost no quality difference between samples decoded with and without 120 ms of packet loss.

4. DISCUSSION

While we present results for three different bitrates, the key feature of our architecture is that the HVQVAE can achieve multiple bitrates by changing the size of the encoded features and/or the size of the codebook. Furthermore, the HVQVAE is trained independent of the vocoder. This leads to the following observations for future work — 1) The WaveRNN can be replaced by other neural vocoders such as the MelGAN [22] or HiFi-GAN [23], which have been reported to produce higher speech quality with more efficient compute, and 2) Explicit f0 features can be provided with the cost of additional bitrate to the neural vocoder to improve stability of decoded speech.

5. CONCLUSION

In this paper we present a variable bitrate neural speech codec that uses a hierarchical VQ-VAE encoder with a configurable decoder that uses a WaveRNN on high-compute devices and a parametric vocoder on low-compute devices. MOS experiments show that the decoded speech quality at the lowest bitrate of our model is able to achieve quality that is on par with Lyra, with the quality improving at higher bitrates. We also demonstrate the effectiveness of our models in combating packet loss by simulating PLC masking during training. In future work, we will explore expanding our model to music and also experiment with adversarial losses to bolster audio quality.

6. REFERENCES

[1] Douglas O'Shaughnessy, "Linear predictive coding," *IEEE potentials*, vol. 7, no. 1, pp. 29–32, 1988.

[2] Manfred Schroeder and BS Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," in *ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1985, vol. 10, pp. 937–940.

[3] Robert J McAulay and TF Quatieri, "Sinusoidal coding.," Tech. Rep., MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 1995.

[4] D Rowe, "Codec 2 - Open source speech coding at 2400 bits/s and below," in *TAPR and ARRL 30th Digital Communications Conference*, 2011, pp. 80–84.

[5] Jean-Marc Valin, Koen Vos, and Timothy Terriberry, "Definition of the Opus audio codec," *IETF, September*, 2012.

[6] Jan Skoglund and Jean-Marc Valin, "Improving Opus low bit rate quality with neural speech synthesis," *arXiv preprint arXiv:1905.04628*, 2019.

[7] Jean-Marc Valin and Jan Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[8] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[9] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu, "Neural discrete representation learning," *arXiv preprint arXiv:1711.00937*, 2017.

[10] Cristina Gârbacea, Aäron van den Oord, Yazhe Li, Felicia SC Lim, Alejandro Luebs, Oriol Vinyals, and Thomas C Walters, "Low bit-rate speech coding with VQ-VAE and a WaveNet decoder," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 735–739.

[11] W Bastiaan Kleijn, Andrew Storus, Michael Chinen, Tom Denton, Felicia SC Lim, Alejandro Luebs, Jan Skoglund, and Hengchin Yeh, "Generative speech coding with predictive variance regularization," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6478–6482.

[12] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[13] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi, "SoundStream: An end-to-end neural audio codec," *arXiv preprint arXiv:2107.03312*, 2021.

[14] Kai Zhen, Jongmo Sung, Mi Suk Lee, Seungkwon Beack, and Minje Kim, "Cascaded cross-module residual learning towards lightweight end-to-end speech coding," *arXiv preprint arXiv:1906.07769*, 2019.

[15] Darius Petermann, Seungkwon Beack, and Minje Kim, "HarpNet: Hyper-autoencoded reconstruction propagation for scalable neural audio coding," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2021, pp. 316–320.

[16] Ali Razavi, Aaron van den Oord, and Oriol Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Advances in neural information processing systems*, 2019, pp. 14866–14876.

[17] Will Williams, Sam Ringer, Tom Ash, John Hughes, David MacLeod, and Jamie Dougherty, "Hierarchical quantized autoencoders," *arXiv preprint arXiv:2002.08111*, 2020.

[18] D. Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[19] Chandan KA Reddy, Harishchandra Dubey, Kazuhito Koishida, Arun Nair, Vishak Gopal, Ross Cutler, Sebastian Braun, Hannes Gamper, Robert Aichner, and Sriram Srinivasan, "Interspeech 2021 deep noise suppression challenge," in *INTERSPEECH*, 2021.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[22] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *arXiv preprint arXiv:1910.06711*, 2019.

[23] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *arXiv preprint arXiv:2010.05646*, 2020.