SPEECH ENHANCEMENT FOR LOW BIT RATE SPEECH CODEC

Ju Lin [1*], *Kaustubh Kalgaonkar* [2], *Qing He* [2], *Xin Lei* [2]

[1] Clemson University, USA [2] Facebook AI, USA

ABSTRACT

Speech codec compresses the input signal into compact bit stream, which is then decoded at the receiver to generate the best possible perceptual quality. This compression makes storing and transmitting speech efficient. In this work, we propose a neural extension to low bit rate speech codec (e.g., Codec2) that aims to improve the perceptual quality of synthesized speech. Our proposed framework combines decoded audio with neural embeddings without breaking the existing speech coders. In addition to embeddings, we also use the least-square generative adversarial network (LSGAN) to reduce artifacts and prevent over-smoothing in the reconstructed audio. The Mean Opinion Scores (MOS) from the listening tests show that our framework can boost the audio quality of speech encoded at 3.6kbps to outperform that of speech encoded at 6kbps using Opus.

Index Terms— Speech Codec, VQ-VAE, generative adversarial network

1. INTRODUCTION

Speech codecs typically compress speech signal to compact bitstream by using hand-crafted features that eliminate redundant and/or unnecessary information. The decoded speech quality depends on the bitrates used. High quality speech codecs typically operate at high bitrates over 6 kbps. Traditional parametric coding of speech facilitates low rate but the resulting speech often sounds with a robotic character. Examples are linear predictive vocoders [1, 2] or sinusoidal coders [3, 4]. Hybrid waveform coders, e.g., AMR coder [5], Opus [6], aim to obtain a faithful reconstruction on a sample-by-sample basis. However, they require higher rates to achieve good quality. The speech quality of waveform matching hybrid coders even underperform the quality of parametric coders at very low rates (below 6kbps) [7].

Recently, deep learning techniques have been introduced to mitigate the limitations of traditional low bit rate speech codecs. These approaches may be classified into end-to-end and neural augmented speech codecs. In neural augmented speech codecs, the decoder of the traditional speech codecs is replaced with a neural decoder. For example, Skoglund et al. [7] proposed to use LPCNet and WaveNet to improve low bit rate Opus quality by resynthesizing speech from the decoded parameters. A WaveNet was used in [8] to enhance the speech quality from the bit stream of a standard parametric coder operating at 2.4 kb/s. For the end-to-end speech codecs, the encoder and decoder are both neural networks learned in a joint training fashion, by using quantized bottlenecks, such as vector-quantized variational autoencoder (VQ-VAE) [9]. For instance, [10] introduced a framework that uses VQ-VAE as encoder and WaveNet as decoder, which can be used to perform very low bit-rate speech coding with high reconstruction quality. In [11], the authors investigated the compressor-enhancer encoders and accompanying decoders based on VQ-VAE autoencoders with WaveRNN [12] decoders.

Although aforementioned deep learning based speech codecs can produce very high-quality speech, these autoregressive decoders also need high computational cost since samples are generated one by one. More recently, generative adversarial networks (GAN) have been applied into speech codecs with non-autoregressive decoders that can generate high-quilaity speech with lower computational cost. SoundStream, proposed in [13], adopted a fully convolutional encoder/decoder network and a residual vector quantizer with adversarial training that can deliver audio at high perceptual quality. In [14], self-supervised discrete representations were investigated with HiFiGAN [15] decoder. In [16], a GAN-based coded audio enhancer was proposed, which directly operates on the decoded audio signals.

In this paper, we propose a neural extension to Codec2 [17] (a parametric codec designed for low bit-rates). This work is an attempt to explore if it is possible to enhance the output of existing low bit rate codecs using some additional information provided in form of embeddings. In addition, the proposed neural enhancement does not break the existing speech coder, which could be also desirable. The proposed framework consists of two branches: first branch is the traditional parametric Codec2 and the second consists of the neural embeddings extracted from original uncompressed audio. A complex convolutional recurrent network (CCRN) is investigated to post-process the decoded audio in a non-autoregressive manner. The CCRN comprises of a convolutional encoder-decoder structure which extracts high-level features with 2-D convolution, as well as long short-term memory (LSTM) layers which capture long-span dependencies in temporal sequences. The quantized neural embeddings are then incorporated into the CCRN framework. We also conduct listening test to validate and compare the quality of the synthesized output with existing standard codecs.

2. PROPOSED APPROACH

In this work we chose Codec2 as the underlying parametric speech coder. This choice was motivated by the fact that Codec2 allows us to experiment with ultra low bit-rates if necessary. Even though Codec2 operates at 8kHz we have designed our system to focuses on 16kHz audio. As shown in Fig. 1, our proposed framework is composed of Codec2 components (encoder and decoder), feature encoder, Vector-Quantization layer and neural enhancement networks CCRN. We will describe these components in details in the following subsections.

The codec encoder consists of two branches; the first branch works on 8kHz speech signal and compresses the audio using the Codec2 encoder, the second branch uses the fullband speech signal to extract the compressed neural embeddings. The neural embeddings are extracted form the STFT of the incoming audio and quantized. We extract STFT for each 10ms frame of incoming audio.

*This work was performed while author was an intern at Facebook AI.

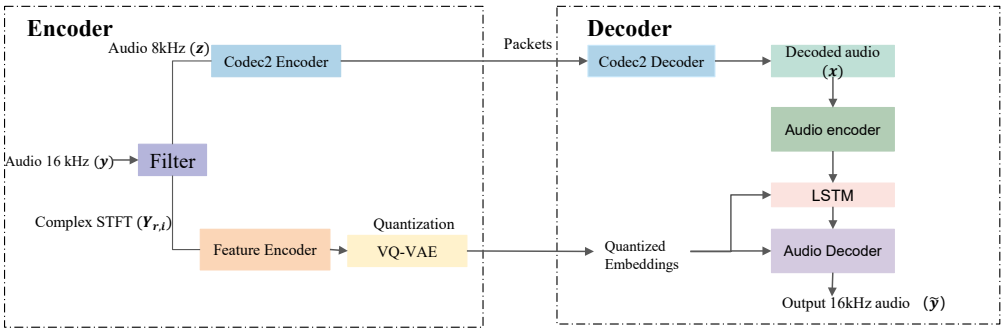


Fig. 1. The proposed neural extension framework to Codec2.

This 100 frames a second rate allows the embeddings generator to be in sync with Codec2 encoder.

2.1. Codec2

Codec2 [17] is an open-sourced speech coder, which belongs to the sinusoidal coder family and can run at various update rates from 450bps to 3.2kbps. Codec2 operates on narrow-band speech with a sampling rate of 8 kHz. It encodes audio into a parametric set: spectral envelope, pitch, and voicing level. For instance, at 1.2 kb/s, each 40 ms audio encodes the short-time spectral envelope using Harmonic magnitudes with 27 bits, the pitch and energy with 16 bits, the voicing level with 4 bits and spare with 1 bit.

2.2. Feature Encoder

The feature encoder takes full-band complex STFT $\boldsymbol{Y}_{r,i}$ as input. The real and imaginary components are ingested as two independent channels. We explore two designs for this feature encoder and these choices are motivated by the fact that speech signal has more energy in the lower part of the spectrum and may benefit if we account for that during embedding extractions.

The first architecture we explicitly split the frequency bins into low and high frequency parts which are independently encoders. Each encoder consists of five stacked convolutional blocks, which are similar to the block used in [18]. Each block is composed of a 2-D convolutional layer, followed by batch normalization [19] and gated linear units (GLU) [20] as activation function.

In the second variant use a single encoder across the entire spectrum. In this case however we use six blocks instead of five used in the split frequency architecture. The embeddings are further processed with multiple codebooks with the assumption that the VQ-VAE layer has the ability to automatically allocate more bits to the regions of interest as needed.

2.3. Vector-Quantization Layer

The vector quantization (VQ) layer quantizes the outputs of the feature encoder using a set of codebooks. We uses the VQ layer in two different configurations in line with how the feature encoder is configured. The first configuration which we call Split Frequency (SF) employs different codebooks across section of the spectrum. The second Split Channels (SC) employs different codebooks across cluster of channels.

Different combinations of split and choice of number of codebooks will control the bit overall rate of the end to end system. We have experimented with two different configurations: two 9-bit and four 6-bit codebooks which translated to a bitrate increase of 1.8kbps and 2.4kbps over Codec2 baseline respectively.

The quantized embedding $\tilde{\boldsymbol{z}}_{\boldsymbol{e}}$ is calculated by a nearest neighbour look-up using the shared embedding space $\boldsymbol{e}$. Where $\boldsymbol{e} \in \mathbb{R}^{K \times D}$ is the latent embedding space, K is the size of the discrete latent space, D is the dimensionality of each latent embedding vector e_i and $\boldsymbol{z}_{\boldsymbol{e}}$ denote the output of the feature encoder. We use VQ-VAE defined in loss [9]:

$$\mathcal{L}_{\text{vq}} = ||\text{sg}[\boldsymbol{z}_{\boldsymbol{e}}] - \tilde{\boldsymbol{z}}_{\boldsymbol{e}}||_2^2 + \beta||\boldsymbol{z}_{\boldsymbol{e}} - \text{sg}[\tilde{\boldsymbol{z}}_{\boldsymbol{e}}]||_2^2, \quad (1)$$

where sg denotes $stopgradient$ operator and the weight β controls the amount of contribution of this loss. We set commitment loss coefficient $\beta = 0.25$. The first term is optimized by an exponential moving average (EMA) k-means [9].

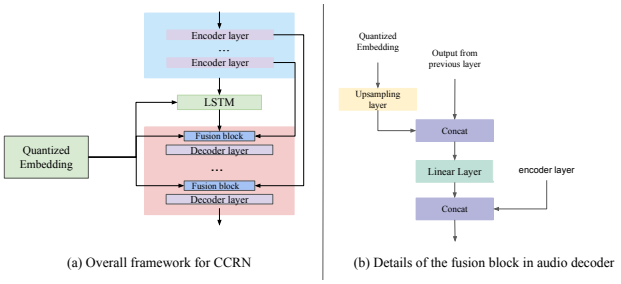
2.4. Complex Convolutional Recurrent Network

The enhancement CCRN has an encoder-decoder structure. The audio-encoder is similar to feature-encoder, which consists of 5 Conv2d blocks, each of which includes a 2-D convolutional layer, a batch normalization layer [19], and the GLU as activation function. The output of the encoder is passed through two LSTM layers, which capture long-term temporal dependencies. The audio-decoder consists of 5 Transposed Convolution2d blocks, and serves to convert the low-resolution features generated by the LSTM layers into high-resolution spectrograms. Each transposed convolutional block consists of a 2-D transposed convolutional layer, followed by batch normalization and GLU. We include skip connections from each encoder layer to its corresponding decoder layer. This is done in order to avoid losing fine-resolution details and to facilitate optimization.

As shown in Fig.2 (a), the quantized embedding is fused in both the LSTM layers and audio-decoder layers. Fig.2 (b) illustrates the details of the fusion block in the audio-decoder. The fusion block is inserted before each decoder layer. The quantized embedding is first passed to a upsampling layer with stride of 2 before concatenating the output with the result of previous decoder layer. This output of the fusion layer is combined with the output of correspond encoder layer after a linear projection to match the dimensions.

2.5. Adversarial Training

In our preliminary experiments we noticed that the reconstructed spectrograms had little variation in the high frequency band, this observations was similar to one in [21]. The lack of detail in the high frequency region resulted in poor audio quality specifically in unvoiced speech. To address this problem we fine-tuned our models using LSGAN.



(a) Overall framework for CCRN (b) Details of the fusion block in audio decoder

Fig. 2. The fusion block in the audio decoder for quantized embedding.

A generative adversarial network (GAN) consists of a generator network (G) and a discriminator network (D). The two components are trained in an "adversarial" fashion: the discriminator tries to distinguish between the samples produced by the generator from real samples, and the generator tries to fool the discriminator by generating realistic samples.

We use the CCRN with VQ-VAE introduced above as the generator G and are trained jointly. The discriminator takes paired STFT magnitudes as input. Both target and enhanced magnitude are conditioned on the input magnitude that is extracted from the upsampled decoded audio. For the discriminator D, we investigate two configurations based on how real pair and fake pair data combined. Channel-wise concatenation is referred to as LSGAN-V1 and frequency-wise concatenation is referred to as LSGAN-V2. Note that in the frequency-wise concatenation, only first 4kHz frequency bands of the upsampled decoded audio are used. Discriminator consists of several Conv2d blocks and two fully connected layers. We use the leaky ReLU activation in all the Conv2d blocks. The last fully connected layer does not use any non-linear activation function, to produce a score. This score is expected to be close to 1 for real samples, and close to 0 for fake samples.

2.6. Loss Function

Our proposed framework is trained jointly using reconstruction loss and adversarial loss. Let $\tilde{\boldsymbol{y}}$ be the enhanced 16 kHz signal in the time domain. Let $\boldsymbol{X}$ denote the STFT magnitude of upsampled decoded audio $\mathbf{x}$.

The reconstruction loss includes an L_1 loss in the time domain, a weight STFT loss (WSTFT) and the VQ-VAE loss (equation 1). In all our experiments we set $\lambda_1 = 1$, $\lambda_2 = 22$ (FFT scaling factor which in our case $\lfloor\sqrt{512}\rfloor$) and $\lambda_3 = 1$.

$$\mathcal{L}_{\text{recon}} = \lambda_1||\boldsymbol{y} - \tilde{\boldsymbol{y}}||_1 + \lambda_2\mathcal{L}_{\text{WSTFT}}(\boldsymbol{Y}, \tilde{\boldsymbol{Y}}) + \lambda_3\mathcal{L}_{\text{vq}}, \quad (2)$$

The WSTFT is proposed to emphasize the high frequency region. We split the frequency bins into 4 sub-bands and each sub-band is assigned a specific weight. The hyperparameter w_k in Eq. 3 was set to (0.1, 1.0, 1.5, 1.5) empirically.

$$\mathcal{L}_{\text{WSTFT}} = \sum_{k=1}^{4} w_k||\boldsymbol{Y}_k - \tilde{\boldsymbol{Y}}_k||_1, \quad (3)$$

The adversarial loss for the generator is defined as:

$$\mathcal{L}_{\mathbf{adv}} = \frac{1}{2}\mathbb{E}_{(\boldsymbol{X_{r,i}}, \boldsymbol{Y_{r,i}})\sim p_{\text{data}}(\boldsymbol{X_{r,i}}, \boldsymbol{Y_{r,i}})}[(\mathbf{D}(\mathbf{G}(\boldsymbol{X_{r,i}}, \boldsymbol{Y_{r,i}}), \boldsymbol{X}) - 1)^2] \quad (4)$$

where it tries to fool the discriminator and produces scores close to 1 on its output. Note that the G takes inputs from feature encoder and audio encoder. The total loss function of the generator can be written as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\mathbf{adv}}, \quad (5)$$

The discriminator network D seeks to distinguish real data from generated data by minimizing the following loss function:

$$\mathcal{L}_{\mathbf{D}} = \frac{1}{2}\mathbb{E}_{\boldsymbol{Y},\boldsymbol{X}\sim p_{\text{data}}(\boldsymbol{Y},\boldsymbol{X})}[(\mathbf{D}(\boldsymbol{Y}, \boldsymbol{X}) - 1)^2] + \frac{1}{2}\mathbb{E}_{(\boldsymbol{X_{r,i}}, \boldsymbol{Y_{r,i}}))\sim p_{\text{data}}(\boldsymbol{X_{r,i}}, \boldsymbol{Y_{r,i}}))}[\mathbf{D}(\mathbf{G}(\boldsymbol{X_{r,i}}, \boldsymbol{Y_{r,i}}), \boldsymbol{X})^2] \quad (6)$$

3. EXPERIMENTS

3.1. Dataset

The database is derived from the DNS Challenge [22]. The training clean speech is cut into 10-second segments, which is about 204k segments in total. The validation set uses 150 10-second segments from the DNS development data set, and the test set uses 15 sentences from Librispeech [23] test-clean data set and 15 sentences from VCTK [24] data set. The models and codec need on 16kHz audio and we downsample the utterances from original dataset for this training. For the short-term Fourier transform, we use Hanning windows of 20 ms, hop size of 10 ms and FFT length of 512 points. For the loss calculation we use a 32 ms Hanning window with 16 ms hop. Using a larger hop for loss calculation which is different than the training window helps account for and reduce boundary artifacts in overlap-and-add synthesis.

3.2. Evaluation metrics

We measure the performance of our models and compare it with other codecs using a listening test and Mean Opinion Scores (MOS). We selected a total of 30 sentences mentioned in section 3.1 and each sentence was rated by about independent 20 raters, resulting in a total of around 6000 ratings. The listeners were asked to rate each sentence on a scale of 1 to 5. The MOS scores are with 95% confidence intervals.

3.3. System Setup

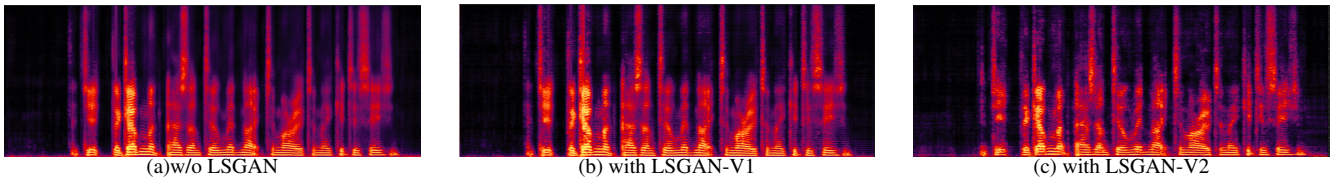
3.3.1. Baseline systems

We compared low bit rate codec to two popular speech codecs Opus and Codec2. Opus supports speech and audio with bandwidth from 4 kHz to 24 kHz and bitrates from 6 kbps to 510 kbps and Codec2 supports 8kHz audio with bit rate from 450bps to 3.2kbps. For comparisons in this paper we set Opus to a fixed bit rate 6kbps and Codec2 at 3.2kbps.

3.3.2. Model Configuration

The audio-encoder in the CCRN consists of five Conv2d blocks each with 128 channels, filters of size 2×6 and a stride setting of 1×2. Each of the LSTM layers has 512 hidden nodes. The audio-decoder has the same channels, filter and stride settings as the audio-encoder except for the last block which has complex STFT as two channel output. Feature-encoder is similar to the one used in audio encoder except for the last block which used a filter of sizes 2×4.

The discriminator in LSGAN-V1 consists of six Conv2d blocks with same filter sizes of 2×5, but different number output channels [8, 32, 64, 128, 128, 128]. This is followed by two fully connected layers ($256 \rightarrow 1$).



(a)w/o LSGAN (b) with LSGAN-V1 (c) with LSGAN-V2

Fig. 3. The spectrograms of estimated signal by the proposed approaches.

The discriminator in LSGAN-V2 uses [discriminator in LSGAN-V1] + Conv2d blocks with output channel 64 and kernel size 1×1 to reduce feature dimensions.

All the models are trained using the Adam optimizer [25] with an initial learning rate of 0.0002. A mini-batch size of ten segments per GPU is used for all models. We first train the generator with 100 epochs using only the reconstruction loss Equation (2). This pre-trained model is fine-tuned with combination of reconstruction loss and LSGAN loss Equation (5) for 30-60 epochs to avoid model collapse problem. At the fine-tuned stage, we extract four seconds segment of enhanced audio to compute the discriminator loss for computational efficiency.

3.4. Results

3.4.1. Comparison with baseline systems

We first compare the proposed framework with the Opus, Codec2 and the ground truth 16kHz audio. The results of this listening test are presented in the Table 1. We can observe that our proposed system at 3.6kbps outperforms baseline systems (Codec2 and Opus). The average MOS score for the proposed system is 3.58@3.6kbps compared 3.38@6kbps for opus and 3.26@3.2kbps for Codec2. An additional 1.2kbps increase in the bitrate to Codec2 parameters further improves MOS of the overall system from 3.58 to 3.67.

System	Codec2	VQ-VAE	Total bitrates	MOS
Original Audio (16 kHz)	-			4.10 ± 0.070
Opus	-		6kbps	3.38 ± 0.088
Codec2	-		3.2kbps	3.26 ± 0.098
Ours (LSGAN-V2)	1200	2400 (SC)	3.6kbps	3.58 ± 0.082
Ours (LSGAN-V2)	2400	2400 (SC)	4.8kbps	3.67 ± 0.083

Table 1. Performance in terms of MOS score for the proposed and baseline systems.

3.4.2. Impact of the LSGAN

We conducted ablation studies to measure effectiveness of LSGAN. In these studies we fixed the bitrate of the end-to-end system to 4.8kbps. The Table 2 contains the results from listening test that compare three such systems.

Comparing the systems O1 and O2 we can observe including finetuning with LSGAN-V1 impoves the MOS from 3.44 to 3.53. We also observe that using LSGAN-V2 (O3) further improves the audio quality to setup. System O3 trained with LSGAN-V2 has the best MOS score of 3.67. Our current hypothesis is that the upsampled audio only contains 4kHz speech spectrum due to Codec2 constraints and channel-wise concatenation (in LSGAN-V1) with paired data will leave the high frequency spectrum empty.

ID	System	Codec2	VQ-VAE	Total bitrates	MOS
O1	Ours (w/o LSGAN)	2400	2400 (SC)	4.8kbps	3.44 ± 0.093
O2	Ours (LSGAN-V1)	2400	2400 (SC)	4.8kbps	3.53 ± 0.082
O3	Ours (LSGAN-V2)	2400	2400 (SC)	4.8kbps	3.67 ± 0.083

Table 2. Ablation studies for effectiveness of the LSGAN.

The Figure 3 show the spectrograms for enhanced audio from these three system. The output without LSGAN is considerably different than that of the ones with LSGAN fine-tuning. This difference can be observed in the high-frequency regions note in particular the "over-smoothing" effect that happens in the systems without adversarial training (a): the high-frequency part of the estimated spectrogram exhibit patterns of "vertical bars" without much variation across frequency. The systems with adversarial training (b and c) are able to generate more realistic spectrograms with natural variation.

3.4.3. Bit allocation and its importance

We wanted to compare and contrast the bit allocation to the two branches of the system. The systems compared in this section are all trained using LSGAN-V2. The Table 3 consolidates the results from the listening test for three systems that only differ in bit allocation. Comparing O4 to O5 and O5 to O6 we can observe that allocating more bits to the embeddings is better than allocating more bits to the Codec2 parameters. We also have to be cognizant of the fact that bit rate increase in the embeddings section will be accompanied with relatively larger compute increase than what would happen if we increased the Codec2 bitrate.

ID	System	Codec2	VQ-VAE	Total bitrates	MOS
O4	Ours (LSGAN-V2)	1200	2400 (SF)	3.6kbps	3.54 ± 0.082
O5	Ours (LSGAN-V2)	2400	1800 (SF)	4.2kbps	3.54 ± 0.084
O6	Ours (LSGAN-V2)	2400	2400 (SF)	4.8kbps	3.58 ± 0.083

Table 3. Performance in terms of MOS score for variable bitrates.

3.4.4. Comparison of SC and SF

The systems O3 and O6 both operate at 4.8kbps with same bit allocation two branches. These two system are also trained in similar fashion using LSGAN-V2. The only difference in the two systems is types of codebooks used in VQ-VAE. We observe that using SC can achieve better performance than using SF setting. The SC codebook based setup has 0.1 delta MOS over the SF codebook setup. This delta highlights the fact that neural network probably is better at bit allocation than an explicit attempt to model and allocate bits across spectrum.

4. CONCLUSIONS

In this work, we have presented a hybrid speech codec that combines the traditional parametric codec Codec2 and neural embeddings. MOS experiments show that this combination can improve the perceptual quality of synthesized speech significantly. This type of hybrid systems can be integrated with existing Codec2 systems with minimal integration effort. We explored multiple methods to improve the quality of the synthesized audio by incorporating multiple existing techniques like LSGAN and multi-codebook VQ-VAE in training and fine-tuning. In the future we intend to explore architectures with better compute efficiency that do not sacrifice audio quality. We also provided supplementary material for additional information [1].

[1] https://linjucs.github.io/ICASSP2022_supplementary.pdf

5. REFERENCES

[1] Thomas Tremain, "Linear predictive coding systems," in *ICASSP'76. IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1976, vol. 1, pp. 474–478.

[2] Alan McCree, Kwan Truong, E Bryan George, Thomas P Barnwell, and Vishu Viswanathan, "A 2.4 kbit/s melp coder candidate for the new us federal standard," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. IEEE, 1996, vol. 1, pp. 200–203.

[3] Per Hedelin, "A tone oriented voice excited vocoder," in *ICASSP'81. IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1981, vol. 6, pp. 205–208.

[4] Robert McAulay and Thomas Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[5] Bruno Bessette, Redwan Salami, Roch Lefebvre, Milan Jelinek, Jani Rotola-Pukkila, Janne Vainio, Hannu Mikkola, and Kari Jarvinen, "The adaptive multirate wideband speech codec (amr-wb)," *IEEE transactions on speech and audio processing*, vol. 10, no. 8, pp. 620–636, 2002.

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

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

[8] W Bastiaan Kleijn, Felicia SC Lim, Alejandro Luebs, Jan Skoglund, Florian Stimberg, Quan Wang, and Thomas C Walters, "Wavenet based low rate speech coding," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 676–680.

[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] Jonah Casebeer, Vinjai Vale, Umut Isik, Jean-Marc Valin, Ritwik Giri, and Arvindh Krishnaswamy, "Enhancing into the codec: Noise robust speech coding with vector-quantized autoencoders," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 711–715.

[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] Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux, "Speech resynthesis from discrete disentangled self-supervised representations," *arXiv preprint arXiv:2104.00355*, 2021.

[15] 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.

[16] Arijit Biswas and Dai Jia, "Audio codec enhancement with generative adversarial networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 356–360.

[17] 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.

[18] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[19] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[20] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[21] Jonas Sautter, Friedrich Faubel, Markus Buck, and Gerhard Schmidt, "Artificial bandwidth extension using a conditional generative adversarial network with discriminative training," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7005–7009.

[22] Chandan K. A. Reddy, Harishchandra Dubey, Vishak Gopal, Ross Cutler, Sebastian Braun, Hannes Gamper, Robert Aichner, and Sriram Srinivasan, "ICASSP 2021 deep noise suppression challenge," *arXiv:2009.06122*, Sept. 2020.

[23] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," Brisbane, Australia, Apr. 2015, pp. 5206–5210.

[24] Cassia Valentini-Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi, "Investigating RNN-based speech enhancement methods for noise-robust text-to-speech," in *SSW*, 2016, pp. 146–152.

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