ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096509

AUDIODEC: AN OPEN-SOURCE STREAMING HIGH-FIDELITY NEURAL AUDIO CODEC

Yi-Chiao Wu, Israel D. Gebru, Dejan Marković, Alexander Richard

Meta Reality Labs Research, USA

ABSTRACT

A good audio codec for live applications such as telecommunication is characterized by three key properties: (1) compression, i.e. the bitrate that is required to transmit the signal should be as low as possible; (2) latency, i.e. encoding and decoding the signal needs to be fast enough to enable communication without or with only minimal noticeable delay; and (3) reconstruction quality of the signal. In this work, we propose an open-source, streamable, and real-time neural audio codec that achieves strong performance along all three axes: it can reconstruct highly natural sounding 48 kHz speech signals while operating at only 12 kbps and running with less than 6 ms (GPU)/10 ms (CPU) latency. An efficient training paradigm is also demonstrated for developing such neural audio codecs for real-world scenarios. Both objective and subjective evaluations using the VCTK corpus are provided. To sum up, AudioDec is a well-developed plug-and-play benchmark for audio codec applications.

Index Terms— audio codec, end-to-end neural network, open-source, streaming, high-fidelity audio generation

1. INTRODUCTION

An audio codec is a technique to compress audio signals into codes and reconstruct the audio signals on the basis of the codes. A typical audio codec system is composed of encoder, quantizer, and decoder modules. The bitrate of the quantized codes is usually much lower than that of the input audio signals, so the codes are suitable for transmissions or storage. Audio codec techniques have been applied to a variety of real-world applications such as secure communication [1, 2], low-cost mobile and internet communications [3, 4], and live videos and music streamings [5].

Most of the conventional parametric codecs [6–8] were built according to in-domain knowledge of psychoacoustics, human speech production systems, and traditional digital signal processing. Although the bitrate is low, the quality is also low because of bandwidth limitations. Modern parametric audio codecs [9–11] usually achieve acceptable reconstruction quality with 16 kHz or higher sampling rate audio signals as a consequence of the advanced transmission techniques. However, the ad hoc designs and limited modeling capacity of these codecs still result in a significant quality gap between natural and reconstructed audio signals.

Recently, the rapid developments of neural networks (NNs) provide advanced modeling capacity, so many neural codecs have been proposed in past years. The first category is the hybrid codec, which replaces or integrates part of the parametric codecs' modules with NNs to improve the compression or reconstruction performances. For example, Krishnamurthy et al. [12] proposed a neural vector quantizer for speech and image coding. Wu et al. [13]. proposed a neural predictive speech coder with coded excitation inputs. NN-based encoder-decoder structures also have been proposed for different handcraft acoustic features such as spectrograms [14], phonological features, and pitches [15]. However, these codecs still require lots of ad hoc designs for speech signals.

The second category is the vocoder-based codec, which directly reconstructs audio signals using neural vocoders conditioned on the codes from other parametric coders [16] or quantized acoustic features [17–19]. However, the performance and bitrate of vocoder-based codecs are still bonded with the upstream handcraft encoding. To theoretically achieve global optimations and flexible bitrates, end-to-end autoencoders (E2E AEs) with raw waveform I/O [20–27] recently have been investigated. Although these E2E codecs usually attain high-fidelity speech generations, the open-source benchmark is unavailable, the efficient training paradigm is unclear, the comparison to vocoder-based methods is absent, and the system for different applications is inflexible.

To tackle these issues, we present an open-source E2E neural codec, AudioDec[1], with an efficient training paradigm in this paper. The modularized architecture provides the flexibility of developing systems for different applications. Specifically, since both the encoder and decoder are easily replaceable, we can separately develop several specific encoders and decoders for any applications and easily integrate or switch among them for different real-world scenarios such as binaural rendering [28]. One of the state-of-the-art neural vocoders, HiFi-GAN [29], is integrated into our codec, and we argue that the combination of a separate powerful vocoder and a well-trained encoder will attain the highest quality. For practicality, we adopt a group convolution mechanism to make the streamable network run in real-time with low latency on both GPU and CPU. Both objective and subjective experiments are conducted, and the comparisons to the vocoder-based mode are also presented. The experimental results show the effectiveness of the proposed codec to generate high-fidelity 48 kHz speech and give more insight into efficiently building a practical neural codec.

2. BASELINE NEURAL AUDIO CODEC

Since AudioDec adopts an E2E AE-based architecture with the SoundStram backbone [25], the E2E AE-based audio codec foundations and the baseline SoundStream are introduced in this chapter.

2.1. End-to-end Autoencoder-based Audio Codec

A typical E2E AE-based codec consists of encoder, projector, quantizer, and decoder modules. In the encoding stage, raw waveform signals are encoded into representations with a much lower temporal resolution and then projected to the designed multidimensional space. The projected representations are further quantized into codes for transmission or storage. In the decoding stage, the codes are first transferred to the representations by a lookup process, and then the decoder reconstructs the raw waveform based on the representations. Morishima et al. [20] proposed the first NN-based AE codec with raw waveform I/O, but the quantizer applied to the bottleneck features is not jointly trained.

[1] https://github.com/facebookresearch/AudioDec

To jointly optimize all modules, many E2E AEs adopting vector quantization (VQ) [30] have been proposed to tackle the gradient of the VQ. For example, the neural codecs with softmax quantization [21], straight-through gradient, exponential moving average (EMA) [22, 23], and Gumble-softmax [31] have been recently proposed and work well for gradient propagation. In addition, the bitrate is directly related to the VQ codebook size, but training the model with a huge plain codebook is impractical. Therefore, scalable codecs decomposing the fine-coarse structures of the encoded latent codes using residual VQ [25], multi-scale VQ [26], and cross-scale VQ [27] or the output waveforms using cross-module residual learning [24] are introduced for tractable hierarchical codebooks.

2.2. SoundStream Audio Codec

SoundStream is an E2E AE-based neural codec adopting the residual VQ mechanism. For 24 kHz audio signal coding, SoundStream is comparative to the modern state-of-the-art parametric codecs, Opus [10] (12 kbps) and EVS [11] (9.6 kbps), with only 3 kbps while a single SoundStream model can work on different bitrate from 3 kbps to 18 kbps. To meet the streamable and real-time requirements, SoundStream adopts a fully causal convolution architecture. The causality makes the network encode/decode audio signals based on only previous samples, so the whole process can run in a continuous segmental manner. The convolution network takes advantage of parallel computations for efficient encoding/decoding.

Both metric and adversarial losses are adopted to train the model. Specifically, given the input signal $\boldsymbol{x}$ and the output signal $\hat{\boldsymbol{x}}$, the adopted metric mel spectral loss L_{mel} is formulated as

$$L_{\mathrm{mel}}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \mathbb{E}\left[\left\|\mathrm{mel}(\boldsymbol{x}) - \mathrm{mel}(\hat{\boldsymbol{x}})\right\|_1\right], \tag{1}$$

where $\mathrm{mel}()$ denotes the mel spectrogram extraction. Two types of fully convolutional discriminators are utilized, and the main difference is that the short-time Fourier transform discriminator (STFTD) takes a complex spectrogram as the input while the multi-scale discriminators (MSDs) [32] take a waveform as the input. All discriminators adopt the hinge loss over the discriminator output logits. Given a discriminator D and a generator G, the adversarial discriminator loss L_{D} is defined as

$$L_{\mathrm{D}} = \mathbb{E}_{\boldsymbol{x}}\left[\max(0, 1 - D(\boldsymbol{x})) + \max(0, 1 + D(G(\boldsymbol{x})))\right], \tag{2}$$

and the adversarial generator loss L_{adv} is defined as

$$L_{\mathrm{adv}} = \mathbb{E}_{\boldsymbol{x}}\left[\max(0, 1 - D(G(\boldsymbol{x})))\right]. \tag{3}$$

Moreover, the feature matching loss [32] L_{fm} is applied to the feature maps of all discriminators, and the EMA [22] loss L_{vq} is applied to the VQ codebook. Therefore, the overall generator loss is

$$L_{\mathrm{G}} = L_{\mathrm{adv}} + \lambda_{\mathrm{fm}}L_{\mathrm{fm}} + \lambda_{\mathrm{mel}}L_{\mathrm{mel}} + \lambda_{\mathrm{vq}}L_{\mathrm{vq}}, \tag{4}$$

where λ_{fm}, λ_{mel}, and λ_{vq} are the weights.

Although SoundStream achieves high-quality audio reconstruction with a low bitrate, the training efficiency and the model flexibility can be improved. Specifically, in contrast to the very lightweight generator, a generative adversarial network (GAN)-based model usually requires multiple deep discriminators to achieve high-fidelity generation [33], and training these discriminators is time-consuming. However, the GAN training is mostly related to improving the waveform details, high-frequency components, and phase synchronization while modeling the low-frequency component can be learned solely with the metric losses. As a result, directly training the model with both metric and adversarial losses from scratch is inefficient. In addition, training a specific SoundStream model from scratch for each scenario is required, but a practical system should be flexibly adjusted for different scenarios such as switching between mono and binaural outputs according to the user's hardware.

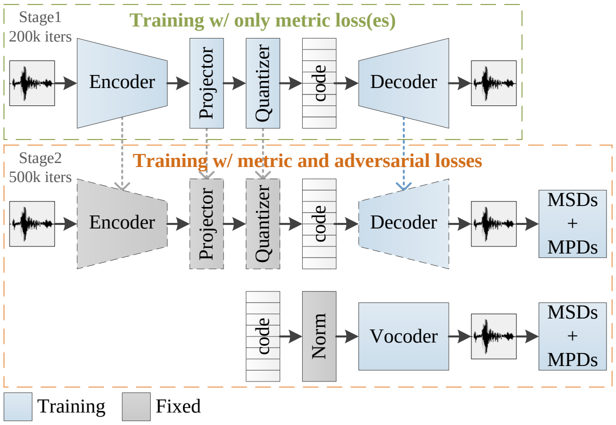

Fig. 1. Training paradigm of AudioDec.

3. PROPOSED NEURAL AUDIO CODEC

To improve the training efficiency, model flexibility, and audio quality of SoundStream, we propose an efficient training paradigm and a modularized architecture and adopt the HiFi-GAN-based multi-period discriminator for developing the AudioDec codec. Moreover, we also apply the group convolution mechanism to the HiFi-GAN vocoder to make it run in real-time on a CPU with 4 threads.

3.1. Efficient Training Paradigm

According to recent speech research, we know that although standard spectral features such as mel spectrogram already include the most high-level information of speech such as contents and speaker identity, the phase information is crucial for generating high-fidelity speech. In addition, for a codec, the codes are expected to contain only essential information for low-bitrate transmissions, and the decoder should be powerful enough to reconstruct high-fidelity waveforms. As a result, we propose an efficient training paradigm to first train both the encoder and decoder with only the metric loss, which makes the training converge fast and stable. Then the discriminators are jointly trained with only the decoder to tackle the details and phase synchronizations of the reconstruction waveforms.

As shown in Fig. 1, given the encoder (including the projector and quantizer) parameters θ and the decoder parameters ϕ, the AudioDec generator is first trained using eq. 1 for the first 200k iterations (stage1). For the following 500k iterations (stage2), the whole model is trained with the updated mel spectral loss:

$$L_{\mathrm{mel}'}(\boldsymbol{x},\hat{\boldsymbol{x}}) = \mathbb{E}\left[\left\|\mathrm{mel}(\boldsymbol{x}) - \mathrm{mel}(f_\phi(\mathrm{sg}[f_\theta(\boldsymbol{x})]))\right\|_1\right], \tag{5}$$

where $\mathrm{sg}[]$ denotes the stop gradient operator, the updated adversarial discriminator loss:

$$L_{\mathrm{D}'} = \mathbb{E}_{\boldsymbol{x}}\left[(1 - D(\boldsymbol{x}))^2 + D(f_\phi(\mathrm{sg}[f_\theta(\boldsymbol{x})]))^2\right], \tag{6}$$

and the updated adversarial generator loss:

$$L_{\mathrm{adv}'} = \mathbb{E}_{\boldsymbol{x}}\left[(1 - D(f_\phi(\mathrm{sg}[f_\theta(\boldsymbol{x})])))^2\right]. \tag{7}$$

The least squares-GAN is adopted to improve the training stability.

3.2. Modularized Architecture

Since the encoder and decoder may be rapidly replaced for different scenarios such as denoising and binaural rendering, modularizing each component of a codec is essential for simultaneously developing encoders and decoders. Inspired by the pretrain mechanism [28, 31], the proposed AudioDec model is first trained with a high-quality clean corpus to obtain a standard quantizer and a complete codebook. By fixing the quantizer and codebook, we can easily develop arbitrarily encoders and decoders for any new scenarios.

Furthermore, we find that the symmetric encoder-decoder architecture is essential for the training stability of an AE. However, the waveform decoder is usually expected to be more powerful to cover all details of the high-resolution audio signals while the encoder is expected to preserve limited essential information of the input audio signals. That is, asymmetric architecture tends to be unstable, and symmetric powerful architecture is inefficient. Therefore, the modularized architecture provides a more flexible design that the decoder of a well-trained lightweight AE codec can be easily replaced by powerful vocoders, and the following experiments actually show that running AudioDec in a vocoder-based mode using a separately trained HiFi-GAN vocoder achieves the best performance.

3.3. HiFi-GAN-based Multi-Period Discriminator

HiFi-GAN is one of the state-of-the-art neural vocoders achieving very high-fidelity speech generation. The main breakthrough of HiFi-GAN is the effectiveness of the proposed multi-period discriminator (MPD) [29]. Specifically, different from MSDs [32] working on the original and downsampling signals, MPDs work on the segmental signals with different segment lengths (periods). Compare to MSD capturing long-term dependency, MPD is effective to capture the periodic details. Since STFTD is also more related to long-term dependency, we find that replacing the redundant STFTD with MPDs does improve the audio quality.

3.4. Low-latency Implementation

Streaming and real-time are two main factors of low-latency encoding/decoding. The causal convolutions and deconvolutions of AudioDec are implemented using one-side padding for streaming. Non-autoregressive (Non-AR) architecture is adopted to achieve real-time coding by parallel computations. However, because the multi-receptive field fusion (MRF) module of HiFi-GAN is not parallel-computation-friendly, it is difficult to run the vocoder-based AudioDec with the vanilla HiFi-GAN generator in real-time. The different kernel sizes of each MRF hinder fully parallel computations, but adopting the largest kernel sizes for all MRFs theoretically attains the same or even better modeling ability. In this paper, we find that the ad hoc kernel sizes are unnecessary, so we utilize the group convolution mechanism [34] to simulate the MRFs with the same kernel size. The proposed group HiFi-GAN generator greatly improves its running time on both GPU and CPU.

4. EXPERIMENTS

4.1. Experimental Setting

The Valentini dataset [35], which is derived from the VCTK corpus [36], is adopted for the evaluations. This English corpus includes 84 gender-balance speakers with accents from England, Scotland, and United States for training and two England speakers (female p257 and male p232) for testing. The number of available utterances of each speaker is around 400. The sample rate of all data is 48 kHz.

Table 1. Objective evaluations of 48 kHz codecs w/ 12.8 kbps

	F_0RMSE (Hz)↓	U/V (%)↓	MCD (dB)↓	LSD (dB)↓	DNSMOS ↑
Natual	-	-	-	-	3.95
SoundStream	12.5	5.2	4.42	0.89	3.81
symAD	11.8	4.9	4.36	0.89	3.88
symAD*	12.9	5.3	4.57	0.89	3.86
asymAD	14.1	5.6	4.45	0.90	3.78
AudioDec v0	12.0	4.9	**4.28**	**0.88**	**3.89**
AudioDec v1	**10.7**	**4.5**	4.29	**0.88**	**3.89**
AudioDec v2	11.8	5.0	4.33	0.89	3.86

symAD*: symAD w/o fixing the encoder

The baseline SoundStream (SS), two symmetric AudioDec (symAD), one asymmetric AudioDec (asymAD), and three vocoder-based AudioDec (AD) systems are evaluated in this chapter. Specifically, to evaluate the effectiveness of the proposed training paradigm, two AudioDecs with symmetric encoder-decoder were respectively trained with and without fixing the encoder during the training of the decoder and discriminators. To investigate the importance of symmetric structure, an asymmetric AudioDec adopting a powerful HiFi-GAN-like decoder with the lightweight AudioDec encoder was trained following the proposed training paradigm. To compare with the vocoder-based approach, three HiFi-GANs including vanilla MRF with kernel sizes 3, 7, and 11 (v0), three group convolutions with kernel size 11 (v1), and three group convolutions with kernel size 3 (v2) were trained using the global normalized codes extracted from the natural waveforms by the well-trained AudioDec encoder.

The neural network architecture and hyperparameters of SS, symADs, asymAD, and ADs followed the settings in the binaural SS paper [28]. Compared with SS, the main modifications of the proposed AudioDec models are the training paradigm, modularized architecture for switching between symmetric and vocoder-based modes, replacing STFTD with MPDs, and adopting LS-GAN for training. The HiFi-GAN-based vocoders followed the popular open-source repository[2]. The number of overall training iterations is 700k. The encoders, quantizers, and codebooks of the models with the proposed training paradigm were fixed after the first 200k training. The HiFi-GAN-based vocoders were trained with 500k iterations to match the overall training iterations. The downsampling rate was set to 300, and each code was represented by eight codebooks with a 1024 book size (8×10 bits), so the bitrate of each codec is 12.8 kbps for 48 kHz audio coding. Details can be referred to our repository[1].

4.2. Objective Evaluation

Five objective measurements were adopted, and the results are the averages of all testing utterances. Specifically, speech quality and prosody are two main factors for evaluating codecs. To evaluate the speech prosody, root mean square errors of fundamental frequency (F_0RMSE) and unvoice/voice (U/V) errors were adopted. To evaluate the speech quality, mel-ceptral distortion (MCD), log-spectral distortion (LSD), and non-intrusive speech quality metric Deep Noise Suppression Mean Opinion Score (DNSMOS) [37] were adopted. The WORLD vocoder [38] was utilized to extract F_0, U/V flags, and mel-cepstral coefficients ($mcep$). A public DNSMOS[3]

[2] https://github.com/kan-bayashi/ParallelWaveGAN

[3] https://github.com/microsoft/DNS-Challenge/tree/master/DNSMOS

Table 2. Mean Opinion Scores of 48 kHz codecs w/ 12.8 kbps

Natural	SoundStream	symAD	AD v0	AD v1	AD v2
4.27±.10	3.28±.13	3.72±.12	3.90±.11	**3.92**±.10	3.78±.12

Table 3. Training speed w/ GPU A100

	Encoder	Decoder	Discriminator	Speed
symAD stage1	✓	✓	×	15.19 it/s
symAD stage2	×	✓	✓	3.77 it/s
symAD*	✓	✓	✓	3.43 it/s

model was adopted. The inputs of the DNSMOS model were downsampled to 16 kHz to match the model, and the SIG_raw scores (for evaluating clean speech) are reported.

As shown in Table 1, the performance differences between symAD and SS show the effectiveness of the MPDs. The even worse results of the symAD w/o the proposed training paradigm (symAD*) demonstrate that the GAN training is more related to improving the decoder. The worst results of asymAD yield the difficulties of training an asymmetric AE. Furthermore, the proposed vocoder-based codecs (AudioDec v*) achieve the best performance of all measurements, and the group convolution networks work as well as the MRF network showing that the ad hoc kernel sizes are unnecessary. In conclusion, training a powerful vocoder with the globally normalized codes extracted from a well-trained audio encoder is a efficient and trackable way to build a neural codec.

4.3. Subjective Evaluation

To evaluate the perceptual quality of the codecs, we conducted MOS tests of a testing subset including randomly selected 15 utterances of each speaker. Baseline SS-, symAD-, and three vocoder-based AD-generated utterances were evaluated for their overall quality. Natural speech was also included as a reference, so the total number of testing utterances was 180. Ten subjects, either audio experts or native speakers, with headphones participated in the tests. As shown in Table 2, although there is still a gap between these coding and natural speech, the proposed AD-series codecs significantly outperform the baseline SS, which shows the effectiveness of the proposed mechanisms. The results also show that if the modeling capacity of a vocoder is advanced (e.g. AD v1), the vocoder-based approaches achieve the best performance. Moreover, the competitive scores of AD v1 and AD v0 indicate that the ad hoc kernel size is unnecessary, and the group convolutions work as well as MRF. These comparisons can be found on our demo page[4].

4.4. Discussion

To show the training efficiency of the proposed training paradigm, the symmetric AudioDec was adopted to conduct the training speed evaluations. The models were trained using one NVIDIA A100 SXM 80GB, and the training speed is presented by the average number of iterations in one second. The results in Table 3 show that training with only metric loss is much fast than training with discriminators, so the proposed paradigm is effective for developing encoders of different applications. Moreover, fixing the encoder during the decoder and discriminator training also slightly improves the training speed. In conclusion, with the proposed training paradigm and an A100 GPU, training an encoder for a new application such as denoising takes only 3.5 hrs, which markedly improves the efficiency of new codec developments for different scenarios.

Table 4. Latency analysis w/ GPU RTX3090 (ms)

Window length	Encoder	Decoder			
	AD	sym	v0	v1	v2
12.5 ms	4.8±.00	3.0±.00	12.7±.01	5.6±.00	5.4±.00
25 ms	6.0±.05	3.8±.04	13.1±.02	5.8±.01	5.5±.01
50 ms	5.2±.02	3.3±.01	14.0±.03	6.7±.02	6.2±.02
100 ms	5.1±.01	3.2±.00	13.2±.02	6.0±.03	5.7±.03

Table 5. Latency analysis w/ CPU 3970X and 4 threads (ms)

Window length	Encoder	Decoder			
	AD	sym	v0	v1	v2
12.5 ms	6.8±.02	6.8±.01	28.7±.05	18.5±.02	9.5±.01
25 ms	8.2±.02	8.6±.03	35.1±.11	22.8±.07	11.2±.02
50 ms	9.0±.03	9.4±.03	37.6±.13	29.4±.12	14.1±.05
100 ms	11.8±.04	13.1±.06	45.1±.15	45.5±.30	21.2±.12

To evaluate the streaming capability, the processing times of segmental encoding and decoding with a nonoverlapping sliding window using the codecs were recorded. Although the overall latency is determined by the window length or the overall processing time, the streaming capability depends on only the longest processing time because the encoding and decoding can run in parallel. The evaluations were conducted using 50 randomly selected testing utterances on one NVIDIA GeForce RTX3090 GPU or AMD Ryzen Threadripper 3970X 32-core processor 3.70 GHz CPU with four threads.

As shown in Table 4, because of the powerful GPU parallel computation capacity, the processing times of different window lengths are almost the same. We can find that the symAD, AD v1, and AD v2 codecs are potentially streamable (In a real scenario, there are some additional delays such as transmissions) even with 12.5 ms buffers, which demonstrates the effectiveness of the group convolutions to take advantage of the parallel computations for processing time reductions. On the other hand, even on the CPU with four threads as shown in Table 5, the symAD and AD v2 codecs are still potentially streamable with 12.5 ms buffers. According to our preliminary experiments, a stand-alone audio recording, encoding, decoding, and playing pipeline can work smoothly with a 25 ms buffer size on the GPU using AD v1 and with a 35 ms buffer size on the CPU using AD v2. Since the acceptable maximum latency of normal internet calls is 150 ms, there is still room for other processes. The streaming demo with the pretrained models is also released on our repository[1].

5. CONCLUSION

In this paper, we present an open-source neural audio codec, AudioDec, for high-fidelity 48 kHz audio. The proposed training paradigm markedly reduces the training time for a new encoder while achieving better quality. The proposed modularized architecture enables us to greatly improve the codec speech quality by using a powerful vocoder and advances the flexibility of developing codecs for different scenarios. The proposed low-latency implementations make AudioDec streamable in real-time with a 25 ms window length on both GPU and CPU. In conclusion, AudioDec is a high-quality, efficient, and convenient benchmark for audio codec research.

[4] https://bigpon.github.io/AudioDec_demo/

6. REFERENCES

[1] T. Tremain, "Linear predictive coding systems," in *Proc. ICASSP*. IEEE, 1976, vol. 1, pp. 474–478.

[2] K. Brandenburg and G. Stoll, "ISO/MPEG-1 audio: A generic standard for coding of high-quality digital audio," *AES*, vol. 42, no. 10, pp. 780–792, 1994.

[3] P. Kroon, E. Deprettere, and R. Sluyter, "Regular-pulse excitation–a novel approach to effective and efficient multipulse coding of speech," *IEEE/ACM TASLP*, vol. 34, no. 5, pp. 1054–1063, 1986.

[4] R. Salami et al., "A toll quality 8 kb/s speech codec for the personal communications system (pcs)," *IEEE TVT*, vol. 43, no. 3, pp. 808–816, 1994.

[5] K. R. Rao and J. J. Hwang, *Techniques and standards for image, video, and audio coding*, Prentice-Hall, Inc., 1996.

[6] B. S. Atal and M. R Schroeder, "Adaptive predictive coding of speech signals," *Bell System Technical Journal*, vol. 49, no. 8, pp. 1973–1986, 1970.

[7] M. Schroeder and B. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," in *Proc. ICASSP*. IEEE, 1985, vol. 10, pp. 937–940.

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

[9] B. Bessette et al., "The adaptive multirate wideband speech codec (AMR-WB)," *IEEE TSAP*, vol. 10, no. 8, pp. 620–636, 2002.

[10] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-quality, low-delay music coding in the opus codec," in *AESC 135*. Audio Engineering Society, 2013.

[11] M. Dietz et al., "Overview of the evs codec architecture," in *Proc. ICASSP*. IEEE, 2015, pp. 5698–5702.

[12] A. K. Krishnamurthy, S. C. Ahalt, D. E. Melton, and P. Chen, "Neural networks for vector quantization of speech and images," *IEEE J-SAC*, vol. 8, no. 8, pp. 1449–1457, 1990.

[13] L. Wu, M. Niranjan, and F. Fallside, "Fully vector-quantized neural network-based code-excited nonlinear predictive speech coding," *IEEE TSAP*, vol. 2, no. 4, pp. 482–489, 1994.

[14] L. Deng, M. L. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Proc. Interspeech*. Citeseer, 2010.

[15] M. Cernak et al., "Composition of deep and spiking neural networks for very low bit rate speech coding," *IEEE/ACM TASLP*, vol. 24, no. 12, pp. 2301–2312, 2016.

[16] 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. ICASSP*. IEEE, 2018, pp. 676–680.

[17] J. Klejsa, P. Hedelin, C. Zhou, R. Fejgin, and L. Villemoes, "High-quality speech coding with sample rnn," in *Proc. ICASSP*. IEEE, 2019, pp. 7155–7159.

[18] J.-M. Valin and J. Skoglund, "A real-time wideband neural vocoder at 1.6 kb/s using lpcnet," in *Proc. Interspeech*, 2019, pp. 3406–3410.

[19] A. Mustafa, J. Büthe, S. Korse, K. Gupta, G. Fuchs, and N. Pia, "A streamwise gan vocoder for wideband speech coding at very low bit rate," in *Proc. WASPAA*. IEEE, 2021, pp. 66–70.

[20] S. Morishima, H. Harashima, and Y. Katayama, "Speech coding based on a multi-layer neural network," in *ICC*. IEEE, 1990, pp. 429–433.

[21] S. Kankanahalli, "End-to-end optimized speech coding with deep neural networks," in *Proc. ICASSP*. IEEE, 2018, pp. 2521–2525.

[22] A. Van Den Oord et al., "Neural discrete representation learning," *NIPS*, vol. 30, 2017.

[23] C. Gârbacea and other, "Low bit-rate speech coding with vq-vae and a wavenet decoder," in *Proc. ICASSP*. IEEE, 2019, pp. 735–739.

[24] K. Zhen, J. Sung, M. S. Lee, S. Beack, and M. Kim, "Cascaded cross-module residual learning towards lightweight end-to-end speech coding," in *Proc. Interspeech*, 2019, pp. 3396–3400.

[25] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An end-to-end neural audio codec," *IEEE/ACM TASLP*, vol. 30, pp. 495–507, 2021.

[26] D. Petermann, S. Beack, and M. Kim, "Harp-net: Hyper-autoencoded reconstruction propagation for scalable neural audio coding," in *WASPAA*. IEEE, 2021, pp. 316–320.

[27] X. Jiang, X. Peng, H. Xue, Y. Zhang, and Y. Lu, "Cross-scale vector quantization for scalable neural speech coding," in *Proc. Interspeech*, 2022, pp. 4222–4226.

[28] W.-C. Huang, D. Markovic, A. Richard, I. D. Gebru, and A. Menon, "End-to-end binaural speech synthesis," in *Proc. Interspeech*, 2022, pp. 1218–1222.

[29] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *NIPS*, vol. 33, pp. 17022–17033, 2020.

[30] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proceedings of the IEEE*, vol. 73, no. 11, pp. 1551–1588, 1985.

[31] K. Yang et al., "Audio-visual speech codecs: Rethinking audio-visual speech enhancement by re-synthesis," in *Proc. CVPR*, 2022, pp. 8227–8237.

[32] K. Kumar et al., "MelGAN: generative adversarial networks for conditional waveform synthesis," in *NIPS*, Dec. 2019, pp. 14910–14921.

[33] J. You, D. Kim, G. Nam, G. Hwang, and G. Chae, "GAN vocoder: multi-resolution discriminator is all you need," in *Proc. Interspeech*, 2021, pp. 2177–2181.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[35] C. Valentini-Botinhao, "Noisy speech database for training speech enhancement algorithms and TTS models," *University of Edinburgh. CSTR*, 2017.

[36] C. Veaux et al., "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *University of Edinburgh. CSTR*, 2017.

[37] Chandan KA Reddy et al., "DNSMOS P.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *ICASSP*, 2022.

[38] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.