LMCODEC: A LOW BITRATE SPEECH CODEC WITH CAUSAL TRANSFORMER MODELS

Teerapat Jenrungrot[1], *Michael Chinen*[2], *W. Bastiaan Kleijn*[2,3], *Jan Skoglund*[2], *Zalán Borsos*[2], *Neil Zeghidour*[2], *Marco Tagliasacchi*[2]

[1]University of Washington, Seattle
[2]Google
[3]School of Engineering and Computer Science, Victoria University of Wellington

ABSTRACT

We introduce LMCodec, a causal neural speech codec that provides high quality audio at very low bitrates. The backbone of the system is a causal convolutional codec that encodes audio into a hierarchy of coarse-to-fine tokens using residual vector quantization. LMCodec trains a Transformer language model to predict the fine tokens from the coarse ones in a generative fashion, allowing for the transmission of fewer codes. A second Transformer predicts the uncertainty of the next codes given the past transmitted codes, and is used to perform conditional entropy coding. A MUSHRA subjective test was conducted and shows that the quality is comparable to reference codecs at higher bitrates. Example audio is available at `https://mjenrungrot.github.io/chrome-media-audio-papers/publications/lmcodec`.

Index Terms— speech coding, Transformers, self-supervised learning, generative adversarial networks.

1. INTRODUCTION

Speech coding, which consists of compressing speech signals to a limited number of bits with minimal distortion, is at the core of communication technologies such as mobile telephony or Voice over IP (VoIP). Opus [1] and EVS [2] are state-of-the-art speech coding techniques that combine traditional coding tools, such as Linear Predictive Coding (LPC), Code Excited Linear Prediction (CELP), and Modified Discrete Cosine Transformation (MDCT) to achieve high coding efficiency over different content types and bitrates. These waveform and parametric codecs rely on psychoacoustics expertise to design signal processing pipelines with maximal coding efficiency. Yet, while fast and interpretable, such handcrafted pipelines only represent a fraction of the potential models for a speech codec.

This has motivated data-driven approaches to train neural networks to perform speech coding. These networks leverage large amounts of training data while relaxing the assumptions made on the type of transformations applied by the system [3–10]. In particular, the SoundStream neural codec combines a causal convolutional architecture with a residual vector quantizers. This quantization method produces a hierarchy of coarse-to-fine codes, and allows for efficient compression while providing bitrate scalability. As a result, SoundStream at 3 kbps matches the quality Opus at 12 kbps. However, the quality of most codecs, be they handcrafted or trained, degrades significantly at bitrates lower than 3 kbps.

In this work, we introduce LMCodec, a low bitrate speech codec that combines recent advances in neural audio coding and audio generative modeling. LMCodec uses autoregressive Transformers [11] on SoundStream tokens to (i) model the entropy of the distribution of coarse tokens and (ii) predict fine tokens from the coarse ones. At inference, LMCodec extracts the codes of a SoundStream model from the input waveform. However, instead of sending all codes to the receiver like a SoundStream codec would do, LMCodec only transmits entropy-coded coarse tokens. On the receiver side, the generative language model is used to predict fine tokens from the coarse ones, and a SoundStream decoder then reconstructs audio from the complete token sequence.

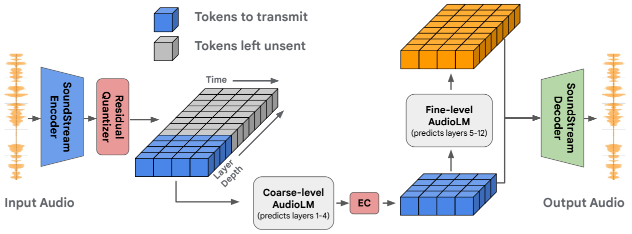

Fig. 1: Overall pipeline of the proposed codec.

LMCodec takes inspiration from the AudioLM [12] generative model, which also predicts fine SoundStream tokens from coarse ones. However, unlike AudioLM, LMCodec does low bitrate compression rather than generative modeling, and to do so leverages AudioLM both as a generative model and an entropy model. Other Transformer-based models for low bitrate coding have been proposed [7,13]. The codec in [13] enriches SoundStream with embeddings extracted from a self-supervised speech representation model [14] and achieves speech compression at a rate of 600 bps. [7] synthesizes speech from a combination of phonetic, pitch and speaker representations to achieve 365 bps. Unlike these models, LMCodec is a fully causal model, which is thus amenable to online encoding and decoding. Our primary contribution is the design of a new neural speech codec, which achieves state-of-the-art results outperforming many previous codecs operating at three to four times the rates according to subjective human evaluation metrics.

Subjective evaluations demonstrate how LMCodec allows for low bitrate speech coding with minimal distortion, with LMCodec at approximately 1-1.5 kbps matching the performance of Opus at 12 kbps. We furthermore analyze the failure modes of our system, as well as the discrepancies in bit allocations between speech and non-speech sections of an audio signal.

2. PROPOSED MODEL

In this section, we describe our proposed speech codec consisting of four components: an encoder, a residual quantizer, an AudioLM

This work was done during a research internship at Google.

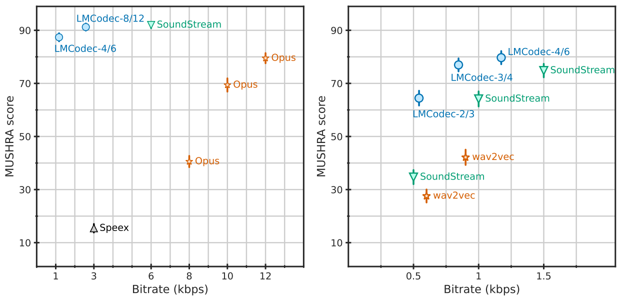

Fig. 2: MUHSRA-like subjective evaluation from state-of-the-art codecs with medium and low bitrates. LMCodec-x/y refers to our model with $N_{\mathcal{C}} = x$ and $N_{\mathcal{C}} + N_{\mathcal{F}} = y$. wav2vec [13] is a recent neural codec based on SoundStream and Transformer.

block, and a decoder. The encoder, residual quantizer, and decoder follow similar structures from SoundStream. At the very high level, the encoder takes raw speech in the time domain as an input and extracts low-rate features that contain sufficient information to reconstruct the speech. The residual quantizer finds discrete representations of the inherently continuous encoded features. The AudioLM poses the modeling of the quantized discrete representation as a language modeling problem and estimates the probability distribution of the next discrete audio token given previous audio tokens. Finally, the decoder reconstructs the input speech signal from the discrete encoded features.

2.1. SoundStream

We now briefly describe the SoundStream model [10] that we used for creating high-quality audio tokens.

2.1.1. *Encoder*

Given a raw speech signal $\mathbf{x} \in [-1, 1]^T$ of length T, the encoder $\mathcal{E} : [-1, 1]^T \to \mathbb{R}^{T_e \times N_e}$ creates a sequence of embeddings of length $T_e \ll T$, each with dimension N_e. In our proposed model, the encoder takes raw waveform speech at $T = 16\,\text{kHz}$ as input and generates $N_e = 128$ dimensional speech features with a frame rate of 50 Hz. The architecture of the encoder is fully convolutional based on causal 1D convolutions. Hence, the algorithmic delay is determined by the overall striding factor (i.e., $T/T_e = 320$ samples or 20 ms).

2.1.2. *Residual Vector Quantizer (RVQ)*

Transmission of continuous speech features over low-bandwidth channels is achieved via vector quantizers (VQs) [10], where the features are turned into discrete representations while introducing minimal distortion. Given the encoded features $\mathbf{e} \in \mathbb{R}^{T_e \times N_e}$, the residual quantizer $\mathcal{Q} : \mathbb{R}^{T_e \times N_e} \to \{0, \dots, 2^{\lceil \log N_c \rceil} - 1\}^{T_e \times N_q}$ computes the corresponding binary representation of $\mathbf{e}$ and its inversion, where N_q is the number of quantizers and N_c is the codebook size of a single quantizer. In our proposed model, we always use the codebook of size $N_c = 2^{10}$ and vary the number of layers in the residual VQs: $N_q \in \{3, 4, 6, 12, 24\}$.

2.1.3. *Decoder*

The decoder $\mathcal{D} : \mathbb{R}^{T_e \times N_e} \to [-1, 1]^T$ synthesizes the original speech signal from the post-quantized embeddings. In our work, we adopt the CNN-based decoder method trained with adversarial loss in addition to losses on waveform and spectral domains. The architecture of the decoder is similar to that of the encoder, with a transposed convolutional layer to upsample the output. The adversarial training framework relies on two types of discriminators: waveform domain and short time Fourier Transform (STFT) domain discriminators.

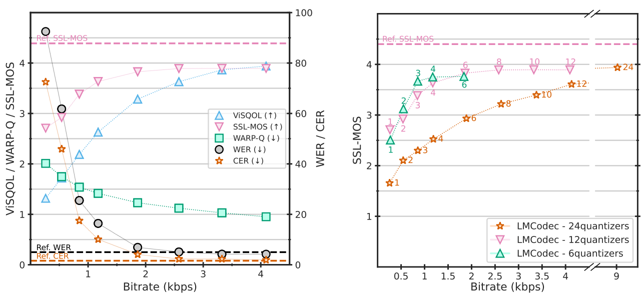

Fig. 3: Objective evaluation of different LMCodec models. (left) LMCodec with a fixed number of RVQ layers (i.e., $N_{\mathcal{C}} + N_{\mathcal{F}} = 12$) on various standard metrics. (right) LMCodec with $N_{\mathcal{C}} + N_{\mathcal{F}} \in \{6, 12, 24\}$ on SSL-MOS [15]. Numbers next to the markers refer to the number of coarse-level codes $N_{\mathcal{C}}$.

2.2. AudioLM

In this subsection, we describe the problem of language modeling of SoundStream tokens. Adding a language model in the bottleneck enables interesting modeling tasks, including modeling the distribution of future SoundStream tokens (Section 2.2.1) or tokens at different VQ layers (Section 2.2.2).

For the rest of this paper, let $N_{\mathcal{C}}$ and $N_{\mathcal{F}}$ denote the number of quantizers for the coarse-level and fine-level AudioLMs, respectively. Figure 1 shows the overall architecture of our proposed model, in which we use $N_{\mathcal{C}} = 4$ and $N_{\mathcal{F}} = 8$. In our experiment, we use various combination of $(N_{\mathcal{C}}, N_{\mathcal{F}})$ ranging from $N_{\mathcal{C}} + N_{\mathcal{F}} = 3$ to $N_{\mathcal{C}} + N_{\mathcal{F}} = 24$. Additionally, let $c_k^{(n)}$ denote the SoundStream token at frame n and VQ layer k.

2.2.1. *Coarse-level AudioLM*

The goal of the coarse-level AudioLM is to model the distribution of the next coarse SoundStream tokens. Specifically, we are interested in modeling the conditional distribution of the next SoundStream tokens given the past information

$$p_{\mathcal{C}}\left(c_k^{(n)} \,\middle|\, \underbrace{c_{k-1}^{(n)}, \dots, c_1^{(n)}}_{\text{coarse-level current frame}}, \underbrace{c_{N_{\mathcal{C}}}^{(n-1)}, \dots, c_1^{(1)}}_{\text{past information}}\right) \tag{1}$$

for $k \in \{1, \dots, N_{\mathcal{C}}\}$.

Given the distribution of the future SoundStream tokens, we build a codec by using lossless Entropy Coding (Section 2.3). More specifically, the discrete probability distribution of SoundStream tokens can be estimated both at the sender and the receiver sides, and we use this to drive an entropy codec. Note that in our proposed method, we only need to transmit $N_{\mathcal{C}}$ tokens per single audio frame. The remaining $N_{\mathcal{F}}$ tokens are generated at the receiver side only as described in the next section.

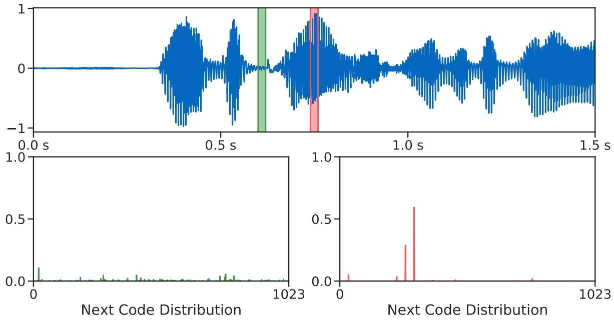

Fig. 4: Distribution of codes prediction for inputs from the non-voice section and inputs from the middle of phonemes

2.2.2. *Fine-level AudioLM*

Similar to the coarse-level AudioLM, the fine-level AudioLM predicts the top VQ layers given the information about bottom VQ layers in addition to the past information. Specifically, we are interested in modeling the distribution of the fine-level SoundStream tokens conditioned on the coarse-level tokens and the past information:

$$p_{\mathcal{F}}\Big(c_k^{(n)}\;\Big|\;\underbrace{c_{k-1}^{(n)},\dots,c_{N_{\mathcal{C}}+1}^{(n)},}_{\text{fine-level current frame}}\;\underbrace{c_{N_{\mathcal{C}}}^{(n)},\dots,c_1^{(n)}}_{\text{coarse-level current frame}}\;,\underbrace{c_{N_{\mathcal{C}}+N_{\mathcal{F}}}^{(n-1)},\dots,c_1^{(1)}}_{\text{past information}}\Big) \quad (2)$$

for $k \in \{N_{\mathcal{C}}+1,\dots,N_{\mathcal{C}}+N_{\mathcal{F}}\}$. Note that our model is causal, in contrast to AudioLM.

Since we only transmit the coarse-level tokens, we model the distribution of the fine-level tokens by assuming that we have access to ground-truth coarse-level SoundStream tokens. We note that, while [12] also proposes a similar fine-level AudioLM stage, our contribution here is the causal formulation of the task, which makes our approach more suitable and amenable to online decoding.

2.3. Entropy Coding (EC)

Given the distribution of coarse-level SoundStream tokens, we transmit data by using entropy coding, a lossless data compression technique. In this work, we provide experimental results using Huffman coding, in addition to the estimated entropy rate. We treat each code from the residual VQs separately and do not perform any grouping to reduce the upper bound on the bitrate.

We first note that our proposed codec requires only sending coarse-level SoundStream tokens using entropy coding. Specifically, given raw audio, LMCodec first encodes audio into SoundStream tokens and models the probability distribution of the next SoundStream tokens, driving the entropy codec. Note that the discrete probability distribution of SoundStream tokens can be estimated both at the sender and the receiver sides, so the receiver can losslessly reconstruct the coarse tokens. To generate audio output from only coarse-level tokens, we use a fine-level AudioLM to synthesize fine-level tokens from the transmitted coarse-level tokens and then generate audio from both coarse-level and fine-level tokens using SoundStream decoder.

2.4. Training Strategy

We adopt a 2-stage training paradigm. First, we train only the encoder, quantizer, and decoder. Then, we freeze the weights of these components and train only the AudioLM components. We train the coarse-level AudioLM and the fine-level AudioLM separately.

2.4.1. *Loss Functions*

We trained the SoundStream model using the standard adversarial loss, feature matching loss, reconstruction loss, and quantization loss according to [10]. In training AudioLM models, we use the standard cross-entropy loss for language modeling over the vocabulary space.

2.4.2. *Training configurations*

To create our codec modules, we adapted the architectures of the encoder, quantizer, generator, and discriminators used in SoundStream [10] and AudioLM from `T5X`. Both AudioLM models are the decoder-only models based on the `base` model of `t5.1.1` (with approximately 250 million parameters).

The SoundStream model is trained on 16 kHz audio from the LibriVox dataset [16] for 1M steps. Both coarse-level and fine-level AudioLM are trained on 16 kHz audio from the Libri-Light dataset [17] for 1M steps with a batch size of 32 and sequence length of 1024 SoundStream tokens with Adafactor optimizer [18] with a decay rate of 0.8.

We trained multiple coarse-level and fine-level AudioLM models to achieve varieties of bitrates. The bitrates are calculated based on the entropy coding of codes from coarse-level AudioLM.

3. EVALUATION

To demonstrate the performance of our proposed method, we evaluate LMCodec using both objective and subjective evaluations. For objective evaluation, we report the accuracy of LMCodec future token prediction and objective metrics including ViSQOL [19], WARP-Q [20], SSL-MOS [15], WER, and CER together with bitrate based on the test split from the clean LibriSpeech dataset [21].

For subjective evaluation, we perform two MUSHRA-like [22] subjective tests to compare the audio quality with standard state-of-the-art speech codecs at medium bitrate (i.e., 1 kbps to 12 kbps) and low rate (i.e., 0.5 kbps to 1.5 kbps). The tests were conducted respectively on 91 and 94 crowd-sourced raters using headphones over 32 clean utterances from VCTK dataset [23]. Raters who did not score the reference above 80 at least 80% of the time were discarded, as were raters who rated more than 75% of non-reference samples 80 or above. 40 raters for the medium rate test and 33 raters for the low rate test met this requirement.

As shown in Figure 2, the raters found that LMCodec-4/6 with 4 quantizers at 1.1 kbps perform significantly better than 12 kbps Opus. LMCodec-8/12 with 8 quantizers at 2.6 kbps has comparable performancce to SoundStream at 6 kbps. The low-rate MUSHRA test compares recent transformer neural codecs and lower bitrate SoundStream models. The raters preferred LMCodec to the transformer models from [13] and SoundStream at the same rate.

3.1. Discussion

Table 1 shows the accuracy of the future token prediction and the bitrate performance of LMCodec from the test split of the clean LibriSpeech [21]. For accuracy, we note that perfect accuracy means the model knows perfectly what the next tokens are. In the context of fine-level AudioLM, this suggests that the model does not necessarily need to synthesize the correct code to produce reasonable audio output. The bitrates are computed based on the future token's distributions obtained from LMCodec. For Huffman coding, we use the ground truth tokens encoded with the Huffman algorithm. Additionally, we note that the distributions of future tokens are updated every timestep based on the model, different from how other entropy codecs that may have fixed distributions operate. So, the Huffman bitrate may sometimes be lower than the bitrate derived from the entropy.

In this section, we additionally discuss some of the interesting audio effects from LMCodec. We suggest that readers listen to some of the audio samples from our model. In particular, our model with only one quantizer is able to produce reasonable human voice with some babbling effects. The amount of babbling is reduced as the number of quantizers used in the codec increases. This suggests that there are some underlying hierarchical structure in SoundStream tokens, and the proposed codec can potentially be operating at very low bitrate, given that the coarse-to-fine prediction is accurate.

In Figure 4, we visualize the distribution of code prediction from the AudioLM when the input is at the middle of a phoneme and between phonemes. We also found that the model is very confident if the audio input is the middle of the phonemes, as the language model network is able to learn underlying linguistic behavior of the utterances. On the other hand, the model has lower confidence in predicting the next token when reaching silence sections, suggesting that our proposed causal model is unable to predict future word really well. This confirms the babbling effect that we observed in the audio output from our proposed codec, which increases as we restrict the amount of information to describe each frame (e.g., by transmitting fewer codes or dropping frames).

Figure 2 shows the comparison of LMCodec with low-rate and medium-rate audio codecs. In particular, we find that LMCodec-4/6 performs better than SoundStream with 3 quantizers at 1.5 kbps but slightly worse than SoundStream with 12 quantizers at 6 kbps which is on par with LMCodec-8/12. We note that LMCodec-4/6 and LMCodec-8/12 are based on SoundStream with 6 and 12 quantizers respectively. Our results suggest that LMCodec effectively takes advantages from entropy coding and synthesizing reasonable fine-level codes from coarse-level codes. When comparing with SoundStream at similar rate, LMCodec essentially outperforms.

3.2. Voice Activity Detection (VAD)

In this section, we show the performance of LMCodec applied only on audio regions with voice activity. We use an open-source RNNoise model [24], which uses Mel-Frequency Cepstral Coefficients (MFCC) and outputs the probability of voice activity every 10 ms frame size. Since the frame size of SoundStream tokens is 20 ms, we run RNNoise on 2 consecutive 10-ms frames and define that the 20-ms SoundStream frame has a voice activity if and only if the probability that 2 consecutive frames have voice is over 0.8.

Table 2 shows the bitrate of LMCodec on two scenarios: (i) transmitting only voices and (ii) transmitting entire speech signals but using zero bits for non-voices. We report the bitrate derived from the entropy and the bitrate based on Huffman coding. We note the first scenario has slightly lower bitrates as compared to bitrates from Table 1 because the entropy for non-speech signals is usually higher than the entropy for speech signals. Additionally, the second scenario provides the lower bound estimate of bitrates when transmitting very low bits for non-voice signals similar to Opus with variable bitrate scheme.

3.3. Objective Evaluation

We presented an objective evaluation on the audio examples from VCTK dataset [23] in Figure 3. First, we demonstrated that the word error rate (WER) and character error rate (CER) are decreasing as the number of quantizers used in the LMCodec increases until around 4-6 quantizers, suggesting that the semantic contents are stored in the coarse tokens. To evaluate WER and CER, we used two ASR models from AWS Transcribe service and Conformer model [14] trained on LibriSpeech [21]. Second, ViSQOL [19] and WARP-Q [20], metrics designed for neural speech codecs, increases and decreases respectively, implying that the fine tokens are responsible for fine-grained acoustic details. Third, the SSL-MOS [15] shows that the overall speech quality improves by increasing the number of quantizers.

$(N_C, N_\mathcal{F})$	**Accuracy**	**Entropy**	**Huffman**
$(2, 1)$	15.5%	534.0 bps	542.5 bps
$(3, 1)$	14.3%	837.1 bps	845.7 bps
$(4, 2)$	13.1%	1163.9 bps	1173.5 bps
$(1, 11)$	16.1%	262.8 bps	262.6 bps
$(2, 10)$	15.7%	533.5 bps	540.7 bps
$(3, 9)$	14.9%	844.6 bps	847.4 bps
$(4, 8)$	13.4%	1154.2 bps	1174.3 bps
$(6, 6)$	11.9%	1853.7 bps	1861.2 bps
$(8, 4)$	10.6%	2561.8 bps	2577.6 bps
$(10, 2)$	9.7%	3300.0 bps	3324.8 bps
$(12, 0)$	8.9%	4094.5 bps	4092.1 bps

Table 1: Accuracy and bitrates. Bitrate without entropy coding is equivalent to 500 bps per quantizer (i.e., 6 kbps for 12 quantizers). Given the space limit, we only present the numerical results for LMCodec with 12 RVQ layers and LMCodec models shown in Figure 2.

$(N_C, N_\mathcal{F})$	Transmitting only voices		Transmitting non-voices with zero bits	
	Entropy	**Huffman**	**Entropy**	**Huffman**
$(2, 1)$	545.6 bps	554.1 bps	303.1 bps	307.9 bps
$(3, 1)$	850.5 bps	858.6 bps	472.1 bps	476.6 bps
$(4, 2)$	1165.6 bps	1173.7 bps	647.2 bps	651.7 bps
$(1, 11)$	268.3 bps	268.7 bps	149.3 bps	149.5 bps
$(2, 10)$	523.7 bps	530.0 bps	290.8 bps	294.3 bps
$(3, 9)$	816.5 bps	819.1 bps	453.2 bps	454.7 bps
$(4, 8)$	1108.5 bps	1129.7 bps	615.2 bps	627.0 bps
$(6, 6)$	1775.2 bps	1783.6 bps	985.5 bps	990.2 bps
$(8, 4)$	2457.3 bps	2471.5 bps	1363.5 bps	1371.4 bps
$(10, 2)$	3170.9 bps	3196.7 bps	1763.4 bps	1777.8 bps
$(12, 0)$	3958.2 bps	3951.5 bps	2207.6 bps	2203.9 bps

Table 2: Coding performance of LMCodec with VAD.

Despite neural speech codecs metrics ViSQOL and WARP-Q indicating worse performance at about 4-6 quantizers, our listening test shows very high quality audio results with small number of quantizers. This suggests that the language model of LMCodec is able to model the distribution of the fine tokens given the coarse tokens reasonably well even if the synthesized fine tokens are different from the ground truth ones. This drives metrics like ViSQOL and WARP-Q down as they primarily rely on the comparison between synthesized audio and its corresponding ground truth reference audio.

When comparing LMCodec with different total number of quantizers, we first note that the upper bound performance of LMCodec with 6 quantizers is lower than the upper bound performance of LMCodec with 12 or 24 quantizers. However, LMCodec with a lower total number of quantizers reaches better performance faster than LMCodec with a higher total number of quantizers.

4. CONCLUSION

Our experiments show that the proposed codec significantly outperforms the original neural speech codec with respect to the quality of synthesized speech when operating in the ultra-low bitrate regime. In addition, the subjective experiments indicate comparable to or better perceptual speech quality compared to conventional codecs operating at higher rates.

5. REFERENCES

[1] J.-M. Valin, K. Vos, and T. Terriberry, "Definition of the Opus audio codec," IETF RFC 6716, 2012.

[2] M. Dietz, M. Multrus, V. Eksler, V. Malenovsky, E. Norvell, H. Pobloth, L. Miao, Z. Wang, L. Laaksonen, A. Vasilache, et al., "Overview of the EVS codec architecture," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5698–5702.

[3] S. Morishima, H. Harashima, and Y. Katayama, "Speech coding based on a multi-layer neural network," in *IEEE International Conference on Communications, Including Supercomm Technical Sessions*, 1990, pp. 429–433.

[4] S. Kankanahalli, "End-to-end optimized speech coding with deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2521–2525.

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

[6] C. Gârbacea, A. van den Oord, Y. Li, F. S. C. Lim, A. Luebs, O. Vinyals, and T. C. Walters, "Low bit-rate speech coding with VQ-VAE and a WaveNet decoder," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 735–739.

[7] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech resynthesis from discrete disentangled self-supervised representations," *arXiv preprint arXiv:2104.00355*, 2021.

[8] 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 *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, 2019, pp. 3396–3400.

[9] D. Petermann, S. Beack, and M. Kim, "Harp-net: Hyper-autoencoded reconstruction propagation for scalable neural audio coding," *CoRR*, vol. abs/2107.10843, 2021.

[10] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[12] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "AudioLM: a Language Modeling Approach to Audio Generation," *arXiv preprint arXiv:2209.03143*, 2022.

[13] A. Siahkoohi, M. Chinen, T. Denton, W. B. Kleijn, and J. Skoglund, "Ultra-low-bitrate speech coding with pretrained transformers," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association*, 2022.

[14] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association*, 2020, pp. 5036–5040.

[15] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of mos prediction networks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8442–8446.

[16] J. Kearns, "Librivox: Free public domain audiobooks," *Reference Reviews*, 2014.

[17] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, et al., "Libri-light: A benchmark for ASR with limited or no supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7669–7673.

[18] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4596–4604.

[19] M. Chinen, F. S. Lim, J. Skoglund, N. Gureev, F. O'Gorman, and A. Hines, "ViSQOL v3: An open source production ready objective speech and audio metric," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–6.

[20] W. A. Jassim, J. Skoglund, M. Chinen, and A. Hines, "WARP-Q: Quality prediction for generative neural speech codecs," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 401–405.

[21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[22] "ITU-R Recommendation BS 1534-3. Method for the subjective assessment of intermediate quality levels of coding systems," *International Telecommunications Union*, 2015.

[23] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," 2019.

[24] J.-M. Valin, "A hybrid DSP/deep learning approach to real-time full-band speech enhancement," in *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, 2018, pp. 1–5.