Interspeech 2022
18-22 September 2022, Incheon, Korea

Cross-Scale Vector Quantization for Scalable Neural Speech Coding

Xue Jiang[1*], *Xiulian Peng*[2], *Huaying Xue*[2], *Yuan Zhang*[1], *Yan Lu*[2]

[1]Communication University of China, Beijing, China
[2]Microsoft Research Asia, Beijing, China

jiangxhoho@cuc.edu.cn,xipe@microsoft.com,
huxue@microsoft.com,yzhang@cuc.edu.cn,yanlu@microsoft.com

Abstract

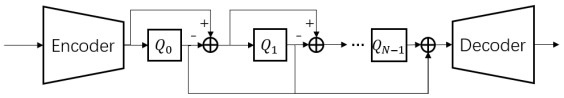
Bitrate scalability is a desirable feature for audio coding in real-time communications. Existing neural audio codecs usually enforce a specific bitrate during training, so different models need to be trained for each target bitrate, which increases the memory footprint at the sender and the receiver side and transcoding is often needed to support multiple receivers. In this paper, we introduce a cross-scale scalable vector quantization scheme (CSVQ), in which multi-scale features are encoded progressively with stepwise feature fusion and refinement. In this way, a coarse-level signal is reconstructed if only a portion of the bitstream is received, and progressively improves the quality as more bits are available. The proposed CSVQ scheme can be flexibly applied to any neural audio coding network with a mirrored auto-encoder structure to achieve bitrate scalability. Subjective results show that the proposed scheme outperforms the classical residual VQ (RVQ) with scalability. Moreover, the proposed CSVQ at 3 kbps outperforms Opus at 9 kbps and Lyra at 3kbps and it could provide a graceful quality boost with bitrate increase.

Index Terms: neural audio coding, bitrate scalable, vector quantization

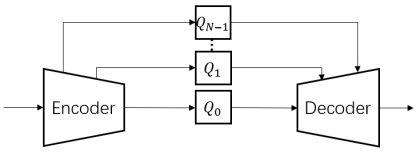
1. Introduction

Audio coding typically employs a carefully-designed pipeline to remove the redundancy in the source signal and yield a compact bitstream. The goal of audio coding is to represent a audio signal with minimum number of bits while retaining its quality. Recently many deep learning-based methods have been proposed for audio coding and achieved very promising results. Some researchers leverage advances in speech synthesizing with generative models [1, 2, 3, 4, 5, 6, 7, 8], such as WaveNet [5], its variants WaveGRU in Lyra [6], LPCNet [7] and SampleRNN [8]. They typically utilize a powerful generative decoder model conditioned on handcrafted acoustic features extracted from a speech signal. On the other hand, some researchers propose end-to-end neural networks based on the vector-quantized variational autoencoder framework (VQ-VAE [9]) where the encoder, codebook and the decoder are learned in a joint fashion [10, 11, 12, 13, 14, 15]. These methods encode the input signal into a discrete representation and then reconstruct the original signal from these latent sequences. These methods have demonstrated the ability of deep neural networks to produce high quality audio at a low bitrate. However, they usually enforce a specific bitrate during training and require retraining multiple models for different target bitrates.

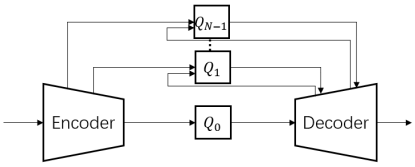
Bitrate scalability is a desirable feature in coding, especially for streaming and real-time communications. Some researchers introduce residual vector quantization (RVQ [16]) into audio coding to achieve bitrate scalability [12, 14, 17, 18]. RVQ provides a convenient framework for controlling the bitrate, where each quantizer quantizes the residual from the previous stage and thus gradually improves the quality of the quantized vector as shown in Figure 1(a). However, the N quantizers in RVQ all perform on a fixed-scale feature, typically the output of the encoder, ignoring the multi-scale information from different layers of the encoder. Some researchers [19] replace the identity shortcuts in the original U-Net with additional autoencoders and deliver the multi-scale features in the compressed form to the decoder side, as shown in Figure 1(b). The bitstreams at multiple scales promote the information communication between encoder and decoder so that better layer-wise approximation to the encoder feature is achieved in the decoder which finally leads to better output quality. However, there is no explicit dependency between these bitstreams where the redundancy between them is not controlled.



(a) Single-scale residual VQ [16]



(b) Multi-scale VQ [19]



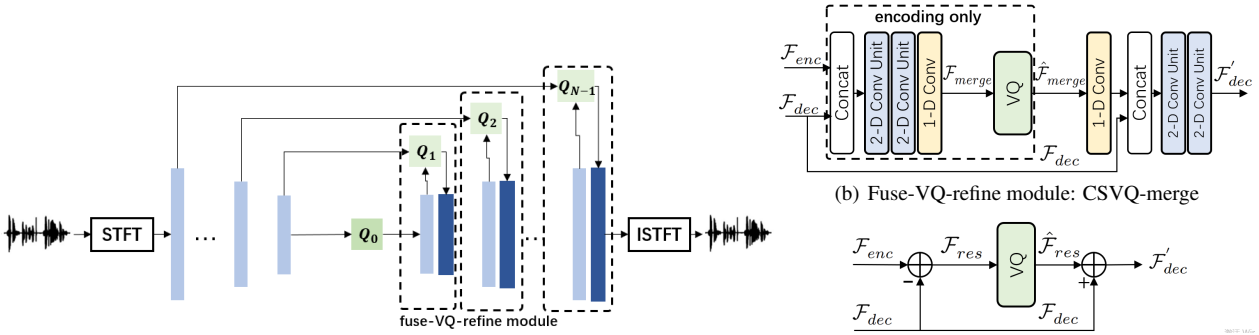
(c) Our cross-scale VQ

Figure 1: *Scalable vector quantization.*

In this paper, we propose the cross-scale scalable vector quantization scheme (CSVQ) as shown in Figure 1(c). Different from that in Figure 1(b), the additional short cuts with bottleneck VQs encode the layer feature of the encoder conditioned on the decoder feature that is produced from previous bitstreams. This explicitly removes the redundancy between different bitstreams and helps to boost the rate-distortion performance. The additional information by the short cut is then merged with the decoder feature for refinement. In CSVQ, the base quantizer Q_0 produces the bitstream with more high-level information and other quantizers $Q_1, Q_2, ..., Q_{N-1}$ produce additional bitstreams containing more detailed high-frequency information. This fuse-VQ-refine module by CSVQ could be

*This work was done when Xue Jiang was an intern at MSRA.

10.21437/Interspeech.2022-10084



(a) Network structure

(b) Fuse-VQ-refine module: CSVQ-merge

(c) Fuse-VQ-refine module: CSVQ-residual

Figure 2: *The overall network structure. A convolutional encoder produces a compact latent representation of the input signal along with multi-scale features, which are quantized using a variable number of vector quantizers after fusion with reconstructed features from decoding. The fully convolutional decoder receives each quantized embedding, refines the reconstructed feature from previous bitstreams with it and finally reconstructs the whole signal.*

flexibly applied to any neural audio coding network with a mirrored autoencoder structure.

Taking the low-latency neural speech codec TFNet [15] as our backbone, we introduce the bitrate-scalable TFNet (S-TFNet) with the proposed CSVQ scheme, which can operate across variable bitrates from 3 kbps to 18 kbps with a single model. The S-TFNet is end-to-end trained in a single stage by randomly sampling a bitrate for each minibatch. Experimental results show that CSVQ outperforms the RVQ subjectively and the S-TFNet at 3kbps outperforms Opus [20] at 9kbps and Lyra [6] at 3kbps with a graceful rate-distortion trade-off. Moreover, the scalable S-TFNet optimized for multiple bitrates performs on par with that optimized for a fixed bitrate, showing the efficiency of the one-stage training algorithm.

2. The Proposed Scheme

2.1. Overview

Taking TFNet [15] as the backbone, the proposed S-TFNet consists of an encoder, several vector quantizers $Q_0, Q_1, ..., Q_{N-1}$, and a decoder, as shown in Figure 2(a). The encoder takes complex time-frequency (T-F) spectrum as input and produces a series of features at different scales, which are then quantized selectively according to the target bitrate after a fusion with each reconstructed feature from the decoder. The decoder receives each quantized embedding, refines the reconstructed feature from previous bitstreams with the new embedding and gradually reconstructs the signal. Causal convolutions are used for the whole network so that it could keep a low latency of 20ms when STFT uses a window of 20ms with a 5ms hop length. The following subsections will explain each part in detail.

2.2. Encoder and Decoder

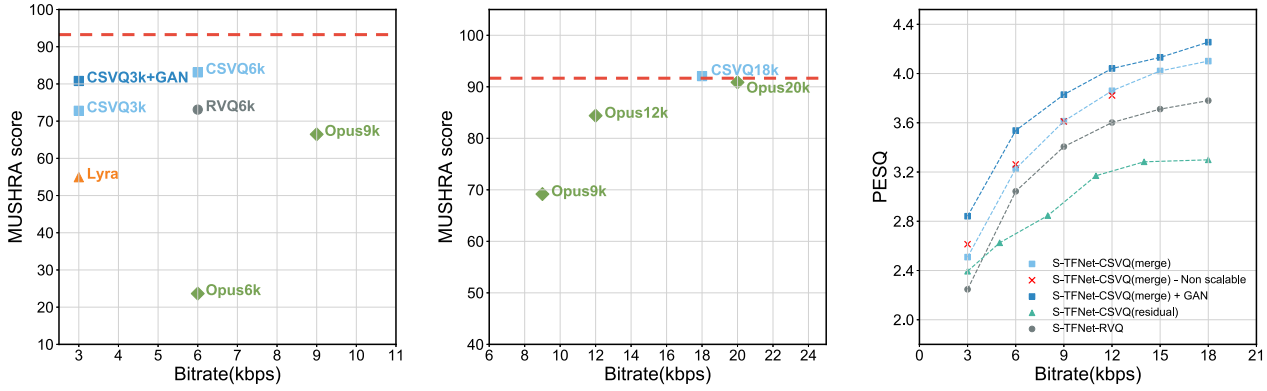
We modify the TFNet structure a little bit to facilitate more bitrate scalability within a single model. The encoder consists of six causal 2-D convolutional layers, followed by a temporal convolution module (TCM) similar to that in [21] and a GRU layer for long-term temporal correlation exploitation. It takes complex spectrum given by short-time Fourier transform (STFT) as the input, denoted by $X^I \in \mathbb{R}^{T\times F\times 2}$, where T is the number of frames and F is the number of frequency bins. The six 2-D convolutional layers successively reduce the size along the frequency dimension with a stride of 2 and finally all frequency information is folded into channels. No down-sampling is performed along the temporal dimension to preserve the temporal resolution. After the convolutional layers, the reshaped feature $X \in \mathbb{R}^{T\times 1\times C}$ is fed into a TCM module followed by a GRU layer which learns a high-level feature by exploring long-range dependencies from the past frames.

The decoder consists of two large TCM modules and one GRU layer in an interleaved manner, followed by a series of 2-D transposed convolutional layers. The interleave TCM and GRU structure captures long-term dependencies from the past to help to reconstruct the orignal signal. The following six 2-D transposed convolutional layers are a mirror-image of convolutional layers at the encoder and each deconvolutional layer is followed by a refinement module in the CSVQ, performing a stepwise refinement. After the decoder, an inverse STFT is applied to reconstruct the waveform audio.

2.3. Cross-Scale Scalable Vector Quantization

Vector Quantizer discretizes the learned features from encoding with a set of trainable codebooks according to the target bitrate. To reduce the codebook size when rate increases, several types of vector quantizers could be used like residual and product. The residual quantizer with multiple stages could also achieve bitrate scalability. However, the quantized features are at a fixed scale, typically the output of the encoder (see Figure 1(a)). Motivated by the spatial scalability in traditional scalable video coding and the U-Net structure, we propose to split bits at different scales of features and propose the cross-scale scalable vector quantization. It leverages a fuse-VQ-refine paradigm for transmitting multiple bitstreams at different scales within a single network.

As shown in Figure 2(a), there are N quantizers. At the lowest bitrate B_0 given by Q_0, only the encoder output is quantized and transmitted to reconstruct a coarse signal. With more bitstreams $B_1, ..., B_i, i = 1, 2, ..., N-1$, more details are recovered. To achieve that, the i-th ($i > 0$) bitstream is generated by fusing the reconstructed feature from previous bitstreams at the decoder $\mathcal{F}^i_{dec}$ with corresponding feature at the encoder $\mathcal{F}^i_{enc}$ and quantizing by Q_i. The quantized feature $\hat{\mathcal{F}}^i_{merge}$ is then used to refine $\mathcal{F}^i_{dec}$ to generate an enhanced one $\mathcal{F}'^i_{dec}$. Specifically, we employ the fuse-VQ-refine module as shown in Figure 2(b). Feature from the encoder $\mathcal{F}^i_{enc}$ is concatenated with that from the decoder $\mathcal{F}^i_{dec}$. They are then fed into two 2-D convolutional layers and downsampled by a 1-D convolutional layer to generate the merge feature $\mathcal{F}^i_{merge}$. During decoding, the quantized feature $\hat{\mathcal{F}}^i_{merge}$ is first upsampled and then concatenated with $\mathcal{F}^i_{dec}$. The following two 2-D convolu-



(a) MUSHRA score (low bitrate) (b) MUSHRA score (high bitrate) (c) PESQ vs. bitrate

Figure 3: *Subjective and objective evaluation results. The red dotted line in (a)(b) represents the score of the reference.*

tional layers are trained to generate the refined feature $\mathcal{F}'^{i}_{dec}$. It should be noted that as partial decoding to get $\mathcal{F}^{i}_{dec}$ needs to be performed during encoding, when there are transmission errors, possible teacher forcing needs to be considered during training. We leave this to future work and don't consider transmission errors here.

As the resolution of feature maps changes in different layers of the encoder, in low-rate scenarios $\mathcal{F}^{i}_{merge}$ fused from deeper encoder layers contains high-level information of the audio and helps to produce a coarse reconstruction. In high-rate scenarios, $\mathcal{F}^{i}_{merge}$ fused from early encoder layers with rich details helps to recover high frequency details.

We adopt a group quantization mechanism similar to [22], but without sharing codebooks across groups. Specifically, each transmitted feature is splitted into G groups and each group is quantized with a separate codebook with K codewords. For the same G and K for all quantizers, each bitstream B_i will consume a constant bit budget given by $R_i = G \log_2 K$ bits. Taking $R = R_0 + R_1$ with two bitstreams as an example, both Q_0 and Q_1 have six codebooks with 1024 codewords in each codebook. Four overlapped frames with 20ms new data are quantized together with the same codebook, so the consumed bitrate could be calculated as $2 \times (6 \times log_2 1024)/0.02 = 6$ kbps. Here no entropy coding is considered nor employed.

Let n denote the number of bitstreams needed to achieve the desired bitrate. During training, we randomly sample $n \in \{1, ..., N\}$ with uniform distribution and only use the first n quantizers $\{Q_0,...,Q_{n-1}\}$ at each iteration for a minibatch. This enables a single model to operate at several target bitrates. We train the codebooks of each quantizer with exponential moving average, following the method proposed in [9]. During inference, when the i-th bitstream is transmitted, the $\hat{\mathcal{F}}^{i}_{merge}$ is used to refine corresponding decoder feature; otherwise, $\hat{\mathcal{F}}^{i}_{merge}$ is set to zero for decoding.

2.4. Training Objective

For good recovery quality, we use several objective terms during training, which is shown below

$$\mathcal{L} = \mathcal{L}_{Comp} + \lambda_1 \mathcal{L}_{Mel} + \lambda_2 \mathcal{L}_{VQ}. \quad (1)$$

The first term $\mathcal{L}_{Comp}$ is the mean-square-error (MSE) loss on the power-law compressed STFT spectrum [23]. To keep STFT consistency [24], the reconstructed spectrum is first transformed to time domain and then to the frequency domain to calculate the loss. The second term $\mathcal{L}_{Mel}$ is the multi-scale mel-spectrogram loss given by

$$\mathcal{L}_{Mel} = \mathbb{E}_s[\sum_{n=1}^{W} ||\phi^n(s) - \phi^n(\hat{s})||_1], \quad (2)$$

where $\phi^n(\cdot)$ is the function that transforms a waveform into the mel-spectrogram using n-th window size. Following [25], we calculate the mel spectra over a sequence of window-lengths between 64 and 2048. The last term $\mathcal{L}_{VQ}$ is the commitment loss similar to that in [9], forcing the encoder to generate a representation approaching the selected codeword. The scalars λ_1 and λ_2 are weights to balance the three terms.

3. Experiments

3.1. Dataset and Settings

We take 150 hours of 16kHz multilingual clean speech from the DNS Challenge at ICASSP 2021 dataset [26]. For training, we use 180000 clips, each of 3 seconds in duration. For evaluation, we use 1158 clips of 10s without any overlapping with the training data. Hanning window is used in STFT with a window length of 20 ms and a hop length of 5 ms.

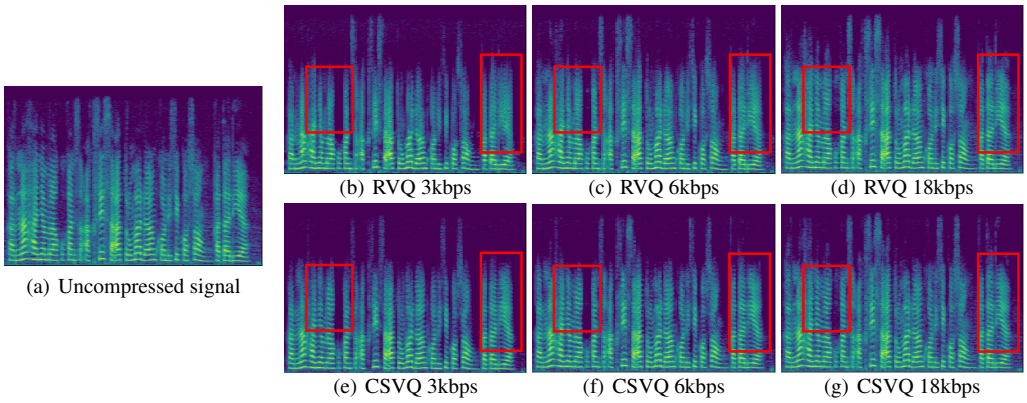
During training, we use adam optimizer with a learning rate of 0.0003. The network is trained for 200 epochs with a batch size of 80.

3.2. Subjective Quality Evaluation

To evaluate the reconstructed signal, we conduct a subjective listening test with a MUSHRA-inspired crowd-sourced methodology [27], where 8 participants evaluate 30 samples. In MUSHRA evaluations, the listener is presented with a hidden reference and a set of test samples by different codecs. The anchor based on low-pass filter is not used in our experiment.

Figure 3(a)(b) shows the subjective evaluation results, where we compare our bitrate-scalable method with two real-time audio codecs, Opus and Lyra. Opus [20] is a versatile codec that is widely used for real-time communications, supporting narrowband to fullband speech and audio with bitrate from 6kbps to 510kbps. Lyra [6] is an autoregressive generative speech codec proposed recently, which reconstructs high quality speech at 3 kbps.

Low bitrate scenarios Figure 3(a) shows the quality versus bitrate evaluation at low bitrate range. It can be seen that our bitrate-scalable model with the proposed CVSQ at 3 kbps



(a) Uncompressed signal (b) RVQ 3kbps (c) RVQ 6kbps (d) RVQ 18kbps (e) CSVQ 3kbps (f) CSVQ 6kbps (g) CSVQ 18kbps

Figure 4: *Visualization.*

(denoted by CSVQ3k) significantly outperforms both Opus at 6 kbps and Lyra at 3kbps, and also performs better than Opus at 9kbps. To better verify the efficiency of the proposed CSVQ scheme, we also implement a RVQ-based scalable method based on the same backbone as CSVQ. It is shown that CSVQ outperforms RVQ at 6kbps, indicating that the rich information carried by multi-scale features indeed helps the reconstruction quality. Although this paper focuses only on the CSVQ technique, to show its potential when combined with other sophisticated techniques in the literature, we also implement a CSVQ scheme with adversarial training, shown as CSVQ+GAN in Figure 3(a). At 3kbps, the CSVQ+GAN largely outperforms CSVQ and it also exceeds Opus at 9kbps by a large margin.

High bitrate scenarios Figure 3(b) shows the comparison at high bitrates. It is shown that CSVQ at 18kps performs slightly better than Opus at 20kbps, indicating that our model could achieve a high quality close to transparent at 18kbps.

3.3. Ablation Study

For ablation study, we use PESQ [28] for quality evaluation. Although it is not proposed for coding quality evalution, we found that when using the same backbone and loss function, it matches our perceptual quality well. Figure 3(c) shows the quality versus bitrate of various schemes.

3.3.1. Bitrate scalability

We evaluate the CSVQ with the bitrate R ranging from 3kbps to 18 kbps. We investigate two different configurations: a) a scalable solution trained with random bitrate sampling and evaluated at bitrate R (denoted as S-TFNet-CSVQ (merge) in Figure 3(c)); b) a non-scalable model trained and evaluated at a fixed bitrate R (denoted as S-TFNet-CSVQ (merge)-Non scalable).

It can be seen that the S-TFNet with proposed CSVQ provides a good tradeoff between quality and bitrate and achieves an obvious quality improvement with the increasing bitrate. Furthermore, the bitrate-scalable model is on par with the non-scalable one, showing the effectiveness of random bitrate sampling during training.

3.3.2. CSVQ vs. RVQ

We further compare the CSVQ with the common RVQ used in [14] across a wide bitrate range from 3 kbps to 18 kbps. It is shown that when using the same backbone, the proposed CSVQ consistently outperforms RVQ at different bitrates.

Figure 4 illustrates the spectrogram of the signal reconstructed by CSVQ and RVQ at different birates. It can be seen that CSVQ produces an output with clearer harmonics than RVQ when both operating at the same bitrate. Additionally, we can observe that more high-frequency details are recovered with the increasing bitrate, indicating that the additional information by $B_1, B_2, ..., B_{N-1}$ carrying rich details can progressively refine the output quality.

3.3.3. Ablation study on fuse-VQ-refine module

Inspired by RVQ, we also investigate a residual variant of the fuse-VQ-refine module (see Figure 2(c)), denoted as S-TFNet-CSVQ(residual) in Figure 3(c). Q_1,...,Q_{N-1} quantize the feature residual between the encoder and decoder instead of learned fused information. To make the residual sparse, an ℓ_2 feature loss is introduced to enforce the decoder feature $\mathcal{F}^i_{dec}$ be close to the encoder feature $\mathcal{F}^i_{enc}$ of the corresponding layer.

From Figure 3(c) we can see that the residual variant shows a similar trend as the merge scheme S-TFNet-CSVQ(merge) that the quality increases with the bitrate. However, the residual variant is much poorer than the proposed merge scheme for CSVQ and this gap gets larger as the bitrate increases. This demonstrates that the merge scheme automatically learns more meaningful information from the multi-scale concatenated features of the encoder and decoder, which could better refine the decoder features than only using the residual information.

3.3.4. Combination with adversarial training

In Figure 3(a) we show the quality boost when combining CSVQ with GAN for training at 3kbps. Here we show the comparison across the wide bitrate range. For discriminators, we use STFT spectrum as the input and four 2D convolutional layers with reduced resolutions in both time and frequency dimensions for extracting features, followed by a fully-connected layer and temporal pooling to produce logits. We use the least-square loss as the GAN objective (LSGAN[29]). Figure 3(c) shows that the adversarial training consistently boosts the quality at various bitrates, indicating the potential of the proposed CSVQ to be combined with other sophisticated techniques for audio/speech coding.

4. Conclusions

We propose the CSVQ, a cross-scale scalable vector quantization that can achieve bitrate scalability with good rate-distortion performance. Both subjective and objective experiments demonstrate its coding efficiency and great potential to be combined with other sophisticated techniques. Although we use speech coding based on TFNet backbone as an example in this paper, it could be applied to general audio coding and any other mirror-like auto-encoder based neural audio coding as well.

5. References

[1] W. B. Kleijn, F. S. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. 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.

[2] J. Klejsa, P. Hedelin, C. Zhou, R. Fejgin, and L. Villemoes, "High-quality speech coding with sample RNN," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7155–7159.

[3] J. Skoglund and J.-M. Valin, "Improving Opus low bit rate quality with neural speech synthesis," *arXiv preprint arXiv:1905.04628*, 2019.

[4] R. Fejgin, J. Klejsa, L. Villemoes, and C. Zhou, "Source coding of audio signals with a generative model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 341–345.

[5] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[6] W. B. Kleijn, A. Storus, M. Chinen, T. Denton, F. S. Lim, A. Luebs, J. Skoglund, and H. 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.

[7] J.-M. Valin and J. 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] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *arXiv preprint arXiv:1612.07837*, 2016.

[9] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

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

[11] C. Gârbacea, A. van den Oord, Y. Li, F. S. Lim, A. Luebs, O. Vinyals, and T. 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.

[12] K. Zhen, J. Sung, M. Lee, S. Beack, and M. Kim, "Cascaded cross-module residual learning towards lightweight end-to-end speech coding," in *Proceedings of the Annual Conference of the International Speech and Communication Association (Interspeech)*, 2019.

[13] J. Casebeer, V. Vale, U. Isik, J.-M. Valin, R. Giri, and A. 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.

[14] 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*, 2021.

[15] X. Jiang, X. Peng, C. Zheng, H. Xue, Y. Zhang, and Y. Lu, "End-to-end neural speech coding for real-time communications," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 866–870.

[16] A. Vasuki and P. Vanathi, "A review of vector quantization techniques," *IEEE Potentials*, vol. 25, no. 4, pp. 39–47, 2006.

[17] K. Zhen, M. S. Lee, J. Sung, S. Beack, and M. Kim, "Efficient and scalable neural residual waveform coding with collaborative quantization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 361–365.

[18] K. Zhen, J. Sung, M. S. Lee, S. Beak, and M. Kim, "Scalable and efficient neural speech coding," *arXiv preprint arXiv:2103.14776*, 2021.

[19] D. Petermann, S. Beack, and M. Kim, "HARP-Net: 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.

[20] J.-M. Valin, K. Vos, and T. Terriberry, "Definition of the Opus audio codec," *IETF, September*, 2012.

[21] A. Pandey and D. Wang, "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6875–6879.

[22] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[23] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," *arXiv preprint arXiv:1804.03619*, 2018.

[24] S. Wisdom, J. R. Hershey, K. Wilson, J. Thorpe, M. Chinen, B. Patton, and R. A. Saurous, "Differentiable consistency constraints for improved deep speech enhancement," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 900–904.

[25] A. Gritsenko, T. Salimans, R. van den Berg, J. Snoek, and N. Kalchbrenner, "A spectral energy distance for parallel speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 062–13 072, 2020.

[26] C. K. A. Reddy, H. Dubey, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, "ICASSP 2021 deep noise suppression challenge," *arXiv preprint arXiv:2009.06122*, 2020.

[27] ITU-R, "Recommendation BS.1534-1: Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2001.

[28] I. Rec, "P.862.2: Wideband extension to recommendation p.862 for the assessment of wideband telephone networks and speech codecs," *International Telecommunication Union, CH–Geneva*, 2005.

[29] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.