R3VQ: Redundancy-Reduced Residual Vector Quantization for Low-Bitrate Neural Speech Coding

Eunkyun Lee, Jongwook Chae, Sooyoung Park, *Graduate Student Member, IEEE*, and Jong Won Shin, *Senior Member, IEEE*

***Abstract*—Neural speech and audio codecs have demonstrated decent quality of the decoded audio at low bitrates. They consist of three parts, an encoder, a decoder, and a quantizer. Residual vector quantization (RVQ) or multi-stage vector quantization in which the residual signal from the previous stage is quantized in the next stage is employed in many neural speech codecs and has exhibited good performance while providing bitrate scalability. In this letter, we propose the redundancy-reduced residual vector quantization (R3VQ) which improves the RVQ by inserting a neural network called a refiner. The role of the refiner is to reduce the power of the residual signal to be quantized by enhancing the estimate of the original speech from the quantized signals in the previous stages. We also present a part-wise (PW) training scheme suitable for the training of the neural speech codec with the R3VQ. Experimental results showed that the proposed R3VQ trained with a PW training scheme outperformed the RVQ in both objective measures for speech quality and subjective MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) test.**

***Index Terms*—Speech coding, Residual Vector Quantization, Vector Quantised-Variational AutoEncoder, SoundStream**

I. Introduction

NEURAL speech and audio codecs [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] get much attention recently because they provide decent quality of the coded-and-decoded speech and audio at low bitrates. Ever since the success of the speech coding using the vector quantized variational autoencoder (VQ-VAE) [13] which adopted the stop gradient operator in the loss function to optimize the quantizer along with neural networks [14], a majority of speech and audio codecs employ a structure consisting of an encoder, a quantizer, and a decoder. Another class of approaches is based on LPCNet [15] which mimics many aspects of traditional linear predictive coding-based speech codecs, and has demonstrated the efficiency in the low-bitrate speech coding [16], [17], [18]. Neural codecs are also used for other applications such as speech synthesis [19], voice conversion [20], speech representation learning [21], and speech enhancement [22]. On the other hand, there have been deep learning-based pre/post-processing methods [23], [24], [25], [26], [27], [28] to improve the quality of speech decoded by legacy speech codecs.

Although most of the researches on neural codecs focus on improving the encoder, decoder, or discriminator [1], [2], [3], [4], [7], several efforts have been made to improve the efficiency of the residual vector quantization (RVQ) [4], [11], [6]. In the Descript Audio Codec (DAC) [4], the input vectors for each stage of an RVQ are projected into low-dimensional vectors before quantization and the decoded code vectors are projected back to the high-dimensional space with learnable projection matrices to ensure that all the code vectors are used regularly. A similar approach to quantize the projected vectors with the scalar quantization and vector quantization was proposed in [11]. In the LMCodec [6], the quantized vectors in the later stages of an RVQ are not transmitted but estimated from the quantized vectors in the early stages of the RVQ and the past information using the fine-level AudioLM.

In this letter, we focus on the possibility to further reduce the dynamic range of the signal to be quantized in each stage of the RVQ by inserting a neural network we call a refiner to predict the original speech from the summation of the estimated speech from the previous refiner and the quantized residual signal in the previous stage and quantize the prediction error instead. To train the neural speech codec (NSC) adopting the proposed scheme named redundancy-reduced residual vector quantization (R3VQ) more effectively, we also present a part-wise (PW) training scheme consisting of three steps each of which focuses on the training of a certain part of the codec with the help of auxiliary modules sequentially and freezes it for the upcoming steps. Experimental results show that a NSC based on the SoundStream [1] with the proposed R3VQ trained with a PW training scheme outperformed the same codec with the original RVQ, the AudioDec [3], and the LPCNet-based neural vocoder [17] in terms of the ViSQOL [29], [30], wideband perceptual evaluation of speech quality (PESQ) [31] scores and the subjective MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) test [32] scores. Some samples are available on the demo page[1].

II. Baseline neural speech codec

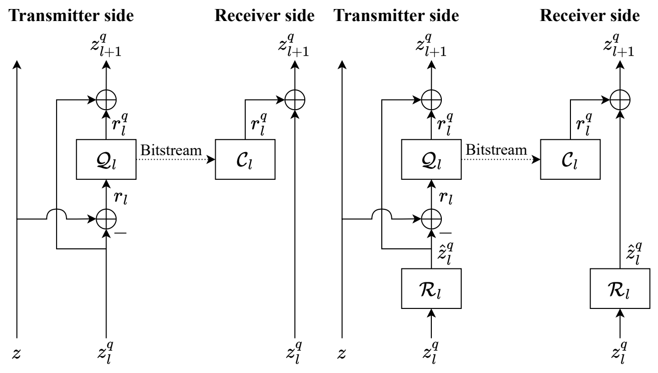
In this work, a modified version of the SoundStream [1] is used as a baseline NSC, which showed better performance for speech coding at 1.6 kbps than the original SoundStream codec in our preliminary experiment. The generator G consists of an encoder $\mathcal{E}$, a quantizer $\mathcal{Q}$, and a decoder $\mathcal{D}$. An input

Manuscript received; revised; accepted. This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [25ZC1100, The research of the basic media·contents technologies]. *(Corresponding author: Jong Won Shin.)*

E. Lee, J. Chae, and J. W. Shin are with Gwangju Institute of Science and Technology, Gwangju 61005, South Korea (e-mail: lek940309@gm.gist.ac.kr; whddnr97@gm.gist.ac.kr; jwshin@gist.ac.kr).

S. Park is with ETRI, Daejeon 34129, South Korea, and Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: sooyoung@etri.re.kr).

[1] https://sapl.gist.ac.kr/demo/R3VQ



(a) (b)

Fig. 1: Block diagrams of the l-th quantization stage of the (a) conventional RVQ and (b) proposed R3VQ.

speech x in the time domain is encoded into $z = \mathcal{E}(x)$, which is then quantized to $z^q = \mathcal{Q}(z)$. The index of the code vector z^q is transmitted to the receiver and the decoder reconstructs speech signal $\hat{x}$ as $\hat{x} = \mathcal{D}(z^q)$.

The RVQ has been employed as $\mathcal{Q}$ and proven to be effective in many neural speech and audio codecs [1], [2], [3], [4], [5], [6], [7], [17], [18]. Fig. 1(a) illustrates an operation of the RVQ at the l-th quantization stage. The quantizer $\mathcal{Q}_l(r_l)$ at the l-th stage quantizes residual signal r_l which is a difference between the original speech z and the speech reconstructed with quantized signals z^q_l from the previous stage. The quantized residual r^q_l is added to z^q_l to form z^q_{l+1}. For the first stage, z^q_1 is set to 0. The codebook index for r^q_l is transmitted to the receiver side and the bitstream is converted back into r^q_l with the codebook lookup $\mathcal{C}_l$. r^q_l for all N stages are summed up to form the reconstructed embedding $z^q = z^q_{N+1} = \sum_{l=1}^{N} r^q_l$, which is fed into the decoder.

As in [1], we train the encoder and decoder via adversarial training [33]. The k-th discriminator D_k out of K discriminators is trained by the hinge loss [1]. As for the generator, the multi-scale spectral reconstruction loss[2] L_{rec} in [34], [1] was used along with the hinge loss L_{adv} and the feature-matching loss L_{fm} in [1], and we have also used the commitment loss [14], $L_{cm}(z^q, z) = ||\mathrm{sg}(z^q) - z||^2_2$ where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation, in the modified version of the SoundStream.

Since the gradient cannot propagate directly to the encoder in training, we used a gradient approximation approach [14] similar to the straight-through estimator [35], where the gradient resulting from z_q passes to the encoder through z. The loss function for the l-th quantizer is given as $L^{(l)}_{cv}(r^q_l, r_l) = ||r^q_l - \mathrm{sg}(r_l)||^2_2$. The weight updates for the encoder and decoder through the backpropagation of the gradient of the generator loss and the update of the code vectors with the exponential moving average (EMA) method [14] to optimize $L^{(l)}_{cv}$ constitute a generator update step, which is followed by a discriminator update step.

III. Redundancy-Reduced Residual Vector Quantization

In this letter, we propose the R3VQ in which a neural network named refiner is inserted before computing the residual signal to be quantized in each stage to remove components of the residual signal predictable from the quantized signals in the preceding stages. The refiners are expected to decrease the dynamic range of the signals to be quantized, enhancing quantization efficiency. Fig. 1(b) shows the block diagram of the l-th stage of the proposed R3VQ in the transmitter and receiver sides. For the first stage, quantized signals in the previous stages don't exist and thus $z^q_1 = \hat{z}^q_1 = 0$ and a refiner is not needed, which makes the first stage of the R3VQ the same as that for the RVQ. The refiner at the l-th stage, $\mathcal{R}_l$, estimates the clean speech z from z^q_l, which is the summation of r^q_{l-1} and $\hat{z}^q_{l-1}$ from the $(l-1)$-th stage. The refiners at the receiver side are identical to those at the transmitter side, and thus $\hat{z}^q_l$ can be computed from the quantized residual signals at the receiver side. And then the residual signal r_l is computed as a difference between the original speech z and the refiner output $\hat{z}^q_l$, which is quantized into r^q_l by the quantizer at the l-th stage, $\mathcal{Q}_l$. At the receiver side, the same refiner produces $\hat{z}^q_l$ from z^q_l, which is added to r^q_l extracted from the received signal by the codebook lookup $\mathcal{C}_l$ to make the reconstructed speech at the l-th stage, z^q_{l+1}. After the N-th stage, the last refiner $\mathcal{R}_{N+1}$ is applied to z^q_{N+1} to produce the final reconstructed speech $z^q = \hat{z}^q_{N+1}$.

Any neural network architecture can be used as a refiner. In this work, each refiner consists of 4 layers of Long-Short Term Memory (LSTM) [36] with a hidden dimension of 128, followed by a fully connected layer. Additionally, a residual skip connection links the input of the LSTM to the final output.

Like the RVQ, R3VQ is updated sequentially, i.e., code vectors in $\mathcal{Q}_1$ are updated, weights in $\mathcal{R}_2$ are updated, code vectors in $\mathcal{Q}_2$ are updated, and so on. While the codebooks for the quantizers $\{\mathcal{Q}_l\}^N_{l=1}$ are updated based on the EMA method, the refiner at the l-th stage is trained to minimize a loss function $L^{(l)}_{ref}$ defined as $L^{(l)}_{ref}(\hat{z}^q_l, z) = ||\hat{z}^q_l - \mathrm{sg}(z)||^2_2$.

The end-to-end (E2E) training of a neural codec with the R3VQ proceeds in a similar way to that with the RVQ. After the weight updates for the encoder and decoder, the codebooks and refiners are updated. However, the convergence of the E2E training of a neural codec with the R3VQ is slow because the update of the encoder may disrupt the convergence of the refiners by persistently moving targets. To alleviate this difficulty, we propose a PW training scheme for neural codecs with the R3VQ inspired by the training scheme of the AudioDec [3], in which the encoder, quantizer, and decoder are jointly trained without any adversarial loss, and then the encoder and quantizer are kept intact and the decoder used in the encoder training or yet another decoder is trained for the quantizer outputs. The PW training scheme consists of three steps, each of which essentially focuses on the training of the encoder, quantizer, and decoder in order. The training of the encoder utilizes auxiliary modules, and the modules trained in the previous step are frozen. The detailed description of the PW training scheme is as follows:

[2] https://github.com/ViEm-ccy/GEDLoss_pytorch/blob/master/spectral_ops.py

Bitrate (kbps)	Model	PESQ	ViSQOL
0.8	NSC-E2E-RVQ	1.428±0.009	2.200±0.017
	NSC-E2E-R3VQ	1.444±0.005	3.277±0.007
	NSC-PW-RVQ	1.490±0.005	3.433±0.005
	NSC-PW-R3VQ	**1.524**±0.005	**3.492**±0.005
1.6	LPCNet [17]	1.587±0.006	3.831±0.006
	AudioDec [3]	1.879±0.007	4.056±0.004
	NSC-E2E-RVQ	1.905±0.007	3.558±0.006
	NSC-E2E-R3VQ	1.897±0.007	3.616±0.006
	NSC-PW-RVQ	2.095±0.007	3.963±0.004
	NSC-PW-R3VQ	**2.194**±0.008	**4.057**±0.004
3.2	AudioDec [3]	2.276±0.008	**4.350**±0.003
	NSC-E2E-RVQ	2.616±0.009	4.045±0.005
	NSC-E2E-R3VQ	2.443±0.008	3.946±0.005
	NSC-PW-RVQ	2.660±0.009	4.267±0.004
	NSC-PW-R3VQ	**2.783**±0.009	4.314±0.004

TABLE I: Average PESQ and ViSQOL scores for the baseline and proposed models with 95% confidence intervals.

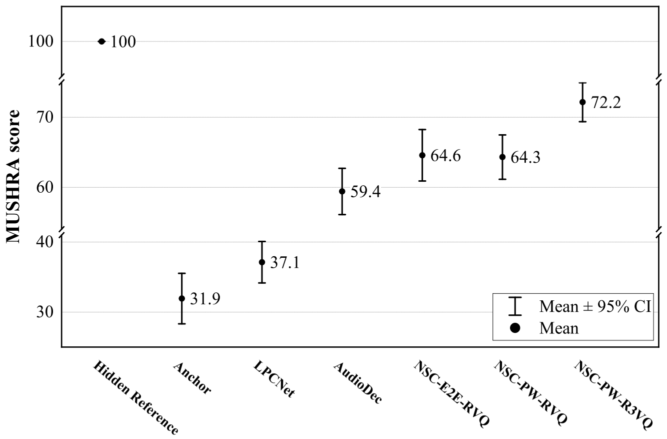


Fig. 2: MUSHRA test results for the LPCNet-based speech codec, AudioDec, and SoundStream-based NSCs operating at 1.6 kbps.

(1) **First step:** The encoder is trained along with an RVQ and a decoder used only for this step with the multi-scale spectral reconstruction loss L_{rec} and the commitment loss L_{cm}

$$\mathcal{L}_{PW}^{\mathcal{E}} = L_{rec}(\hat{x}, x) + \lambda_{cm} L_{cm}(z^q, z). \tag{1}$$

It is noted that the bitrate for the RVQ in this step is set to be higher than the target bitrate as the decoder in this step may not be as well-trained as the final decoder.
(2) **Second step:** Another RVQ or an R3VQ for the final model is fitted to z produced by the encoder which is trained in the first step and then frozen. The respective losses for the RVQ and R3VQ are represented as follows:

$$\mathcal{L}_{PW}^{\mathcal{Q}_{RVQ}} = \sum_{l=1}^{N} L_{cv}^{(l)}(r_l^q, r_l), \tag{2}$$

$$\mathcal{L}_{PW}^{\mathcal{Q}_{R3VQ}} = \sum_{l=1}^{N} L_{cv}^{(l)}(r_l^q, r_l) + \sum_{l=2}^{N+1} L_{ref}^{(l)}(\hat{z}_l^q, z). \tag{3}$$

As in the E2E training of the RVQ and R3VQ, the code vectors in $\mathcal{Q}_l$ for both the RVQ and R3VQ are updated with the EMA method to minimize $L_{cv}^{(l)}(r_l^q, r_l)$, and the weights in $\mathcal{R}_l$ for the R3VQ are adapted to minimize $L_{ref}^{(l)}(\hat{z}_l^q, z)$.
(3) **Final step:** The decoder for the final model is trained with the adversarial training with the encoder and quantizer frozen. The loss function for the decoder is defined as

$$\mathcal{L}_{PW}^{\mathcal{D}} = L_{adv}(\hat{x}, x) + \lambda_{fm} L_{fm}(\hat{x}, x) + \lambda_{rec} L_{rec}(\hat{x}, x). \tag{4}$$

IV. Experiments

A. Experimental settings

We have compared the performances of the SoundStream-based NSCs with RVQ and R3VQ trained by the E2E training and the PW training. The performance of the LPCNet-based speech codec [17] and AudioDec[3] [3] was also compared. We used the training sets of the LibriTTS speech corpus [37] to train all models except the LPCNet-based speech codec and `test-clean` subset to evaluate them[4]. As for the LPCNet-based speech codec, the pre-trained model[5] was used as it performed better than a model trained with the LibriTTS dataset . All the waveforms to train and test the models were resampled to 16 kHz. We measured the ViSQOL [29] [30] and wideband PESQ [31] scores for an objective quality evaluation of the reconstructed speech using the baseline and proposed models. We also performed the MUSHRA test [32] for the subjective quality evaluation of the speech signals coded by the models. 10 audio samples with the length of 2-5 seconds selected from the test set were assessed by 10 expert listeners. The unprocessed original signals with 24 kHz sampling rate were used as the reference signals.

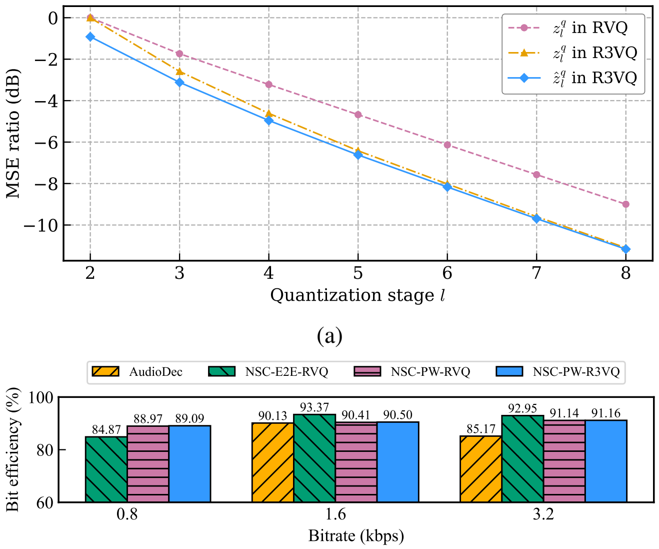
We adopted the encoder and decoder of the streaming SEANet [38] as in the SoundStream [1]. The dimensions of the feature map after the first convolution layer in the encoder and that before the last convolution layer in the decoder were set to 16 and 32, respectively. The kernel size of all dilated convolution layers was 3. The strides for the encoder blocks were set to [2, 4, 5, 8] which became in reverse order for the decoder blocks, and therefore the length of each frame was 20 ms. The numbers of parameters for the encoder, decoder, and each refiner were 1.08M, 4.46M, and 627k, respectively. The strides for the encoder and decoder blocks and bit allocations for AudioDec were set to be the same as those for the SoundStream-based NSCs. The bitrate for the main experiment was set to 1.6 kbps, and thus the RVQ and R3VQ had 4 codebooks with 256 codewords resulting in 32 bits per frame. We have also tested for 0.8 and 3.2 kbps with 2 and 8 codebooks, respectively. The RVQ to train the encoder in the first step of the PW training scheme was composed of 12 quantization stages with 256 codewords. $(\lambda_{cm}, \lambda_{fm}, \lambda_{rec})$ were $(\frac{1}{TK}, 100, 1)$ in which T is the number of frames and K is the dimension of z, which was 256.

The E2E training incorporated the Adam [39] optimizer

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

[4] Similar tendencies were observed for the `test-other` subset as shown in our demo page https://sapl.gist.ac.kr/demo/R3VQ.

[5] https://github.com/xiph/LPCNet/tree/v0.1



(a)

(b)

Fig. 3: Plots of (a) MSEs for z_l^q in the RVQ and those for z_l^q and $\hat{z}_l^q$ in the R3VQ normalized by that for z_2^q in the RVQ and (b) bit efficiencies of the AudioDec, NSC-E2E-RVQ, NSC-PW-RVQ, and NSC-PW-R3VQ for 0.8, 1.6, and 3.2 kbps.

with the learning rate of 0.0001 and the momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.9$ for both the generator and discriminators. The generator and discriminators were updated for 1M iterations. In the first training step of the PW training scheme, the encoder was trained with an RVQ and decoder operating at 4.8 kbps for 600k iterations using the Adam optimizer with the learning rate of 0.0001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The quantizers for the final models were updated for 100k iterations in the second step. Each refiner was trained with the Adam optimizer with the learning rate of 0.0003, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The decoder for the final model was trained for 1M iterations using the same optimizer as that of the E2E training. The mini-batch was composed of 16 speech waveforms with the length of 540 ms for the SoundStream-based models. We used the multi-scale discriminators used in [38] and [1] and the short-time Fourier transform-based discriminator (STFTD) [1] for the adversarial training of the SoundStream-based models. We used the layer normalizations [40] into the STFTD as in [2].

B. Experimental results

In Table I, the average PESQ and ViSQOL scores are shown for the SoundStream-based NSCs with the RVQ and proposed R3VQ trained with the E2E training and the PW training scheme, as well as the LPCNet-based speech codec [17] and AudioDec [3]. The NSC with the proposed R3VQ did not show good performances when trained by an E2E training scheme, but outperformed the conventional NSC-E2E-RVQ and the NSC-PW-RVQ in both PESQ and ViSQOL scores for all bitrates when trained by the PW training. The LPCNet achieved the average ViSQOL score higher than NSC-E2E-RVQ but lower than AudioDec and NSC-PW-R3VQ, but the PESQ scores were the lowest which aligned with the subjective test result. The ViSQOL scores for the AudioDec were the highest in 3.2 kbps and close to the highest in 1.6 kbps, but the PESQ scores were lower than NSC-E2E-RVQ, not to mention NSC-PW-R3VQ.

	LPCNet	AudioDec	NSC with RVQ	NSC with R3VQ
Encoding	**0.0128**	0.0888	0.0348	0.0385
Decoding	0.4010	0.4954	**0.0959**	0.1014
Total	0.4138	0.5842	**0.1307**	0.1399

TABLE II: Real-time factors for the LPCNet-based speech codec, AudioDec, and SoundStream-based NSCs with RVQ and R3VQ operating at 1.6 kbps.

The result of the MUSHRA test is presented in Fig. 2 with the average score and the 95% confidence intervals. It was observed that the MUSHRA test scores exhibited similar tendency to that of the PESQ scores; the LPCNet achieved poorer scores than other models, the AudioDec, NSC-E2E-RVQ, and NSC-PW-RVQ did not show significant differences, and the NSC-PW-R3VQ outperformed other codecs with a statistical significance.

To verify that the refiners in the R3VQ indeed reduced the dynamic range of the residual signal to be quantized, we have measured the mean square errors (MSEs) of z_l^q and $\hat{z}_l^q$ in the R3VQ, which are the powers of the residuals to be quantized without and with the refiner, along with those of z_l^q in the RVQ for $l = 2, \ldots, 8$. Fig. 3(a) shows the MSEs normalized by the MSE for z_2^q in the RVQ in a dB scale. It is clear that the refiners decreased the power of the signals to be quantized, enabling better coding. We can also see that the effect of the refiner was larger in the earlier stages. Fig. 3(b) illustrates the bit efficiencies [41] of the four compared codecs. It can be seen that the bit efficiency of the proposed NSC-PW-R3VQ was higher than 89%, which was similar to or better than the NSC-E2E-RVQ and AudioDec.

Additionally, we estimated the real-time factors (RTFs) using a single thread of an AMD Ryzen5 5600G processor as shown in Table II. The LPCNet exhibited the fastest encoding but the slowest decoding, which would make it second slowest if used in a voice communication system. The encoder and decoder of the AudioDec showed the slowest operation among those of all the speech codecs. The introduction of the R3VQ to the SoundStream-based NSC increased the RTFs of the same NSC with the RVQ by 10% and 6% for encoding and decoding, respectively, which may be at an allowable level considering the performance improvement.

V. Conclusion

In this letter, we propose the R3VQ in which a neural network called a refiner is inserted before computing the residual to be coded in each stage of the RVQ to further remove predictable components from the residual, and the PW training scheme to train the NSC with the R3VQ effectively. Experimental results on low-bitrate speech coding using a SoundStream-based NSC showed that the adoption of the R3VQ led to the performance improvement when trained with the proposed PW training procedure.

References

[1] 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, 2022.

[2] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *Transactions on Machine Learning Research*, 2023.

[3] Y.-C. Wu, I. D. Gebru, D. Marković, and A. Richard, "AudioDec: An open-source streaming high-fidelity neural audio codec," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[4] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-Fidelity Audio Compression with Improved RVQGAN," in *Proc. NeurIPS*, vol. 36, 2023, pp. 27 980–27 993.

[5] H. Yang, I. Jang, and M. Kim, "Generative de-quantization for neural speech codec via latent diffusion," in *Proc. IEEE ICASSP*, 2024, pp. 1251–1255.

[6] T. Jenrungrot, M. Chinen, W. B. Kleijn, J. Skoglund, Z. Borsos, N. Zeghidour, and M. Tagliasacchi, "LMCodec: A low bitrate speech codec with causal transformer models," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[7] S. Ahn, B. J. Woo, M. H. Han, C. Moon, and N. S. Kim, "HILCodec: High-fidelity and lightweight neural audio codec," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–14, 2024.

[8] D. N. Rim, I. Jang, and H. Choi, "Deep neural networks and end-to-end learning for audio compression," *arXiv:2105.11681*, 2021.

[9] X. Jiang, X. Peng, H. Xue, Y. Zhang, and Y. Lu, "Latent-domain predictive neural speech coding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2111–2123, 2023.

[10] X.-H. Jiang, Y. Ai, R.-C. Zheng, H.-P. Du, Y.-X. Lu, and Z.-H. Ling, "MDCTCodec: A lightweight MDCT-based neural audio codec towards high sampling rate and low bitrate scenarios," in *Proc. IEEE SLT*, 2024, pp. 540–547.

[11] X.-H. Jiang, Y. Ai, R.-C. Zheng, and Z.-H. Ling, "A streamable neural audio codec with residual scalar-vector quantization for real-time communication," *IEEE Signal Processing Letters*, vol. 32, pp. 1645–1649, 2025.

[12] W. Liu, Z. Guo, J. Xu, Y. Lv, Y. Chu, Z. Liu, and J. Lin, "Analyzing and mitigating inconsistency in discrete speech tokens for neural codec language models," in *Proc. ACL*, 2025, pp. 31 035–31 046.

[13] C. Gârbacea, A. v. 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 *Proc. IEEE ICASSP*, 2019, pp. 735–739.

[14] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. NIPS*, vol. 30, 2017, pp. 1–10.

[15] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. IEEE ICASSP*, 2019, pp. 5891–5895.

[16] H. Yang, W. Lim, and M. Kim, "Neural feature predictor and discriminative residual coding for low-bitrate speech coding," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[17] 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.

[18] I. Jang, H. Yang, W. Lim, S. Beack, and M. Kim, "Personalized neural speech codec," in *Proc. IEEE ICASSP*, 2024, pp. 991–995.

[19] S. Chen, C. Wang, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 1–15, 2025.

[20] A. Baade, P. Peng, and D. Harwath, "Neural codec language models for disentangled and textless voice conversion," in *Proc. Interspeech*, 2024, pp. 182–186.

[21] H. Zhou, A. Baevski, and M. Auli, "A comparison of discrete latent variable models for speech representation learning," in *Proc. IEEE ICASSP*, 2021, pp. 3050–3054.

[22] H. Yang, J. Su, M. Kim, and Z. Jin, "GenHancer: High-fidelity speech enhancement via generative modeling on discrete codec tokens," in *Proc. Interspeech*, 2024, pp. 1170–1174.

[23] S. Korse, N. Pia, K. Gupta, and G. Fuchs, "PostGAN: A GAN-based post-processor to enhance the quality of coded speech," in *Proc. IEEE ICASSP*, 2022, pp. 831–835.

[24] S. Hwang, Y. Cheon, S. Han, I. Jang, and J. W. Shin, "Enhancement of coded speech using neural network-based side information," *IEEE Access*, vol. 9, pp. 121 532–121 540, 2021.

[25] J. Büthe, J.-M. Valin, and A. Mustafa, "LACE: A light-weight, causal model for enhancing coded speech through adaptive convolutions," in *Proc. IEEE WASPAA*, 2023, pp. 1–5.

[26] J. Büthe, A. Mustafa, J.-M. Valin, K. Helwani, and M. M. Goodwin, "NOLACE: Improving low-complexity speech codec enhancement through adaptive temporal shaping," in *Proc. IEEE ICASSP*, 2024, pp. 476–480.

[27] S. Hwang, E. Lee, I. Jang, and J. W. Shin, "Alias-and-Separate: Wideband speech coding using sub-Nyquist sampling and speech separation," *IEEE Signal Processing Letters*, vol. 29, pp. 2003–2007, 2022.

[28] E. Lee, S. Beack, and J. W. Shin, "Improved Alias-and-Separate speech coding framework with minimal algorithmic delay," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 8, pp. 1414–1426, 2024.

[29] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, "ViSQOL: The virtual speech quality objective listener," in *Proc. IWAENC*, 2012, pp. 1–4.

[30] M. Chinen, F. S. C. Lim, J. Skoglund, N. Gureev, F. O'Gorman, and A. Hines, "ViSQOL v3: An open source production ready objective speech and audio metric," in *Proc. QoMEX*, 2020, pp. 1–6.

[31] ITU-T Rec. P.862.2, "Wideband extension to recommendation P.862 for the assessment of wideband telephone networks and speech codecs," 2007.

[32] ITU-R Rec. BS.1534-3, "Method for the subjective assessment of intermediate quality level of audio systems," 2015.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, vol. 27, 2014, pp. 1–9.

[34] A. Gritsenko, T. Salimans, R. van den Berg, J. Snoek, and N. Kalchbrenner, "A spectral energy distance for parallel speech synthesis," in *Proc. NeurIPS*, vol. 33, 2020, pp. 13 062–13 072.

[35] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv:1308.3432*, 2013.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in *Proc. Interspeech*, 2019, pp. 1526–1530.

[38] Y. Li, M. Tagliasacchi, O. Rybakov, V. Ungureanu, and D. Roblek, "Real-time speech frequency bandwidth extension," in *Proc. IEEE ICASSP*, 2021, pp. 691–695.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR (Conference Track)*, 2015.

[40] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv:1607.06450*, 2016.

[41] R.-C. Zheng, H.-P. Du, X.-H. Jiang, Y. Ai, and Z.-H. Ling, "ERVQ: Enhanced residual vector quantization with intra-and-inter-codebook optimization for neural audio codecs," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 33, pp. 2539–2550, 2025.