Interspeech 2024
1-5 September 2024, Kos, Greece

A Low-Bitrate Neural Audio Codec Framework with Bandwidth Reduction and Recovery for High-Sampling-Rate Waveforms

*Yang Ai, Ye-Xin Lu, Xiao-Hang Jiang, Zheng-Yan Sheng, Rui-Chen Zheng, Zhen-Hua Ling**

National Engineering Research Center of Speech and Language Information Processing,
University of Science and Technology of China, Hefei, P. R. China

yangai@ustc.edu.cn, {yxlu0102, jiang_xiaohang, zysheng, zhengruichen}@mail.ustc.edu.cn,
zhling@ustc.edu.cn

Abstract

This paper proposes a novel neural audio codec framework which incorporates bandwidth reduction and recovery, facilitating its application in scenarios with high sampling rates and low bitrates. The proposed framework consists of a two-stage-downsampling-based encoder, a quantizer, and a two-stage-upsampling-based decoder. The encoder initially reduces the bandwidth of the high-sampling-rate waveform before encoding it. Therefore, the discrete tokens outputted by the quantizer are derived from the low-sampling-rate waveform, resulting in a low bitrate. The decoder decodes the low-sampling-rate waveform and ultimately recovers the original high-sampling-rate waveform by bandwidth recovery. Experiments confirm that our proposed framework achieves high-quality audio coding at a sampling rate of 48 kHz and a bitrate of only 1 kbps. The bitrate savings amount to 6 times compared to baseline codecs without bandwidth reduction and recovery.

Index Terms: neural audio codec, bandwidth reduction, bandwidth recovery, low bitrate, high sampling rate

1. Introduction

Audio codec is a compression technology that discretizes audio data for transmission, storage, or playback purposes. Audio codecs are widely used in various fields such as communication [1, 2] and downstream applications [3, 4]. A good audio codec should maintain high fidelity in decoded audio while striving for a low bitrate (i.e., high compression efficiency).

Parametric codecs, utilizing audio characteristic parameters (such as spectral features) for discretization objects, offer the advantage of lower bitrates. Conversely, this type of codecs often compromises on the decoded audio quality, such as traditional linear predictive coding (LPC) [5]. Subsequently, researchers have tried to integrate traditional parametric codecs with neural vocoders [6, 7, 8, 9]. These neural vocoders are employed to convert discrete tokens discretized by traditional parametric codecs into audio waveforms. This approach effectively enhances the quality of decoded audio, although there remains a certain gap compared to natural audio.

With the advancement of deep learning, end-to-end neural audio codecs are emerging, such as SoundStream [10], Encodec [11] and HiFi-Codec [12]. These codecs directly encode, quantize, and decode the audio waveform, and define generative adversarial network (GAN) based losses between the decoded waveform and the natural one to ensure the fidelity of the decoded audio. Therefore, they all belong to waveform codecs. Although traditional waveform codecs like pulse code modulation (PCM) [13] suffer from high bitrates, the neural waveform codec SoundStream and Encodec adopt a strategy of residual vector quantization (RVQ) [14], reducing the bitrate by connecting multiple vector quantizers (VQs) through residual connections. HiFi-Codec also proposes a group RVQ (GRVQ) technique which requires less codebooks used for quantization to further reduce the bitrate. However, directly encoding and decoding waveforms often entails higher model complexity and is not conducive to further audio compression (i.e., it is difficult to reduce the bitrate significantly).

Recently, there has been increasing attention on high-sampling-rate neural audio codecs. However, the increase in audio samples poses greater challenges to waveform codecs for bitrate reduction. Wu *et al.* proposes AudioDec [15] and ScoreDec [16] targeting 48 kHz audio coding. However, as reported in their papers [15, 16], the bitrates of AudioDec and ScoreDec are as high as 12.8 kbps and 24 kbps respectively. Additionally, they require additional integration with vocoder and post-filter, employing a two-stage training mode. Another alternative approach is to construct neural parametric codecs, which involves transforming the audio waveform into the spectral domain before encoding. The MDCTNet [17] encodes, quantizes and decodes the modified discrete cosine transform (MDCT) spectrum of audio waveform, yet still requires a bitrate of 24 kbps at a sampling rate of 48 kHz. The APCodec [18] operates in the short-time Fourier transform (STFT) domain, encoding, quantizing, and decoding both the amplitude and phase spectra of audio. Although APCodec can achieve high-quality audio coding at a bitrate of just 6 kbps for a sampling rate of 48 kHz, it still faces challenges in meeting very low bitrate application scenarios, such as bandwidth-limited audio communication.

Therefore, to achieve audio coding at high sampling rates and very low bitrates, we propose a novel neural audio codec framework which incorporates the bandwidth extension (BWE) model AP-BWE [19] into the APCodec. The two-stage-downsampling-based encoder, quantizer, and two-stage-upsampling-based decoder constitute the proposed codec framework. As the name suggests, this encoder implies two downsampling processes, converting the high-sampling-rate waveform into codes with extremely low temporal resolution. This results in discrete tokens outputted by the quantizer with very low bitrates. Similarly, the decoder includes two upsampling processes, utilizing the BWE model for bandwidth recovery. Objective and subjective experiments have both confirmed that, at a sampling rate of 48 kHz, APCodec incorporating AP-BWE at 1 kbps achieves decoded audio quality comparable to that of the baseline SoundStream, Encodec and HiFi-Codec at 6 kbps. Therefore, the bitrate savings are approximately 6 times.

* Corresponding author. This work was funded by the National Nature Science Foundation of China under Grant 62301521 and U23B2053, the Anhui Provincial Natural Science Foundation under Grant 2308085QF200, and the Fundamental Research Funds for the Central Universities under Grant WK2100000033.

10.21437/Interspeech.2024-108

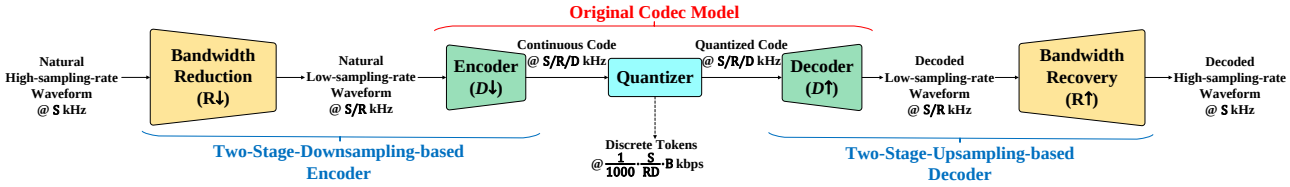

Figure 1: *The overview of the proposed codec framework. Here, ↓ and ↑ represents the downsampling and upsampling, respectively.*

Additionally, the generation speed of the framework is also the fastest, reaching up to 20.5 times real-time speed on CPU.

2. Proposed Methods

2.1. Overview

An overview of the proposed neural audio codec framework is illustrated in Figure 1. The proposed codec framework encodes and quantizes a high-sampling-rate waveform $\boldsymbol{x}_H \in \mathbb{R}^T$ at S kHz, generating discrete tokens for transmission, storage, or downstream tasks, where T is the number of audio waveform samples. Ultimately, the quantized codes acquired by querying the codebooks with these tokens are decoded into a waveform $\hat{\boldsymbol{x}}_H \in \mathbb{R}^T$ at S kHz. The proposed framework consists of a two-stage-downsampling-based encoder, a quantizer and a two-stage-upsampling-based decoder. Specifically, the two-stage-downsampling-based encoder first reduces the bandwidth of $\boldsymbol{x}_H$ by a factor of R, generating a low-sampling-rate waveform $\boldsymbol{x}_L \in \mathbb{R}^{T/R}$ at S/R kHz. It then encodes $\boldsymbol{x}_L$ into a continuous code $\boldsymbol{C} \in \mathbb{R}^{(T/R/D)\times K}$ with a sampling rate of $S/R/D$, implying downsampling by a factor of D, where K is the dimensionality of the code. The quantizer discretizes $\boldsymbol{C}$, generating discrete tokens, with the following bitrate in kbps (kilobits per second):

$$Bitrate = \frac{1}{1000} \cdot \frac{S}{RD} \cdot B, \tag{1}$$

where B represents the number of bits required by the quantizer to discretize one frame of the code. Then, the quantizer converts discrete tokens into quantized code $\hat{\boldsymbol{C}} \in \mathbb{R}^{(T/R/D)\times K}$ for the subsequent decoding process by querying the codebook. In the two-stage-upsampling-based decoder, the quantized code $\hat{\boldsymbol{C}}$ is first decoded into a low-sampling-rate waveform $\hat{\boldsymbol{x}}_L \in \mathbb{R}^{T/R}$ at S/R kHz, implying an upsampling process by a factor of D. To restore the sampling rate, bandwidth recovery operation is utilized to expand the frequency band of the low-sampling-rate waveform $\hat{\boldsymbol{x}}_L$, generating the high-sampling-rate waveform $\hat{\boldsymbol{x}}_H \in \mathbb{R}^T$ at the original sampling rate of S kHz. Therefore, in our proposed codec framework, we actually quantize the low-sampling-rate waveform. Compared to directly manipulating high-sampling-rate waveform, this approach reduces the bitrate by a factor of R, showcasing its advantages in very low bitrate application scenarios.

In our implementation of this paper, we adopt the APCodec [18] for encoding, quantizing, and decoding, and AP-BWE [19] for bandwidth recovery in the framework. Specific details are listed in the following subsections.

2.2. Two-Stage-Downsampling-based Encoder

The two-stage-downsampling-based encoder consists of a bandwidth reduction operation implemented by an R-fold signal-processing-based downsampling operation and an encoder from APCodec [18]. The APCodec is a parametric codec designed to encode, quantize, and decode audio amplitude and phase spectra, rather than the raw waveform. Hence, the low-sampling-rate waveform $\boldsymbol{x}_L \in \mathbb{R}^{T/R}$ at S/R kHz is initially subjected to an STFT with a frame shift of W and FFT point number of N to extract the amplitude spectrum $\boldsymbol{A} \in \mathbb{R}^{(T/R/W)\times(N/2+1)}$ and phase spectrum $\boldsymbol{P} \in \mathbb{R}^{(T/R/W)\times(N/2+1)}$. This spectral extraction process involves W-fold downsampling. Then, parallel amplitude and phase encoding streams separately encode $\boldsymbol{A}$ and $\boldsymbol{P}$ into more compact amplitude code and phase code. Both streams share a similar structure, with the backbone being ConvNeXt v2 network [20] and convolutional layer with a stride of U, involving U-fold downsampling. Therefore, we have $D = W \cdot U$. Finally, after the fusion of the amplitude code and phase code, a dimension reduction layer is employed to generate the continuous code $\boldsymbol{C}$.

2.3. Quantizer

The quantizer adopts RVQ which is commonly used in SoundStream [10], Encodec [11], and APCodec [18], consisting of Q VQs. These Q VQs are cascaded together through residual connections. Each VQ discretizes one frame of the input into an integer token, with values ranging from 1 to M (interval is 1), according to a trainable codebook of size $M \times K$. Therefore, each continuous frame is represented by Q tokens, and in Equation 1, $B = Q \cdot \log_2 M$. Finally, the quantized results obtained by each VQ from querying the codebook with tokens are summed together to obtain the quantized code $\hat{\boldsymbol{C}}$.

2.4. Two-Stage-Upsampling-based Decoder

The two-stage-upsampling-based decoder consists of a decoder from APCodec [18] and a bandwidth recovery operation implemented by the AP-BWE model [19]. The decoder model first restores the dimensions of the quantized code, then generates amplitude spectrum $\hat{\boldsymbol{A}} \in \mathbb{R}^{(T/R/W)\times(N/2+1)}$ and phase spectrum $\hat{\boldsymbol{P}} \in \mathbb{R}^{(T/R/W)\times(N/2+1)}$ through parallel amplitude and phase decoding streams. The structures of the decoder and encoder in APCodec are almost symmetrical. A slight difference lies in that each decoding stream includes a deconvolutional layer with a stride of U, involving U-fold upsampling. In addition, the parallel phase estimation architecture [21] is used to directly and accurately predict the phase. Finally, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{P}}$ are transformed into the decoded low-sampling-rate waveform $\hat{\boldsymbol{x}}_L \in \mathbb{R}^{T/R}$ at S/R kHz through inverse STFT (ISTFT), involving W-fold upsampling.

The AP-BWE [19] is selected to recover the bandwidth of the low-sampling-rate waveform $\hat{\boldsymbol{x}}_L$ because it shares a similar conceptual framework with APCodec [18]. The AP-BWE model first performs sinc interpolation on the $\hat{\boldsymbol{x}}_L$ to generate $\tilde{\boldsymbol{x}}_H \in \mathbb{R}^T$ at sampling rate of S kHz, involving R-fold upsampling. Then, the amplitude spectrum $\tilde{\boldsymbol{A}} \in \mathbb{R}^{(T/W)\times(N/2+1)}$ and phase spectrum $\tilde{\boldsymbol{P}} \in \mathbb{R}^{(T/W)\times(N/2+1)}$ extracted from $\tilde{\boldsymbol{x}}_H$

Table 1: *Objective comparison for bitrate saving evaluations. Here, "$a\times$" represents $a\times$ real time.*

	Bitrate	LSD↓	AWPD_{IP}↓	STOI↑	UTMOS↑	RTF (GPU)↓	RTF (CPU)↓
Natural	–	–	–	–	4.04	–	–
APCodec+AP-BWE	1 kbps	0.926	1.79	0.842	**3.93**	0.00588 (170×)	**0.0488 (20.5 ×)**
SoundStream	6 kbps	0.937	1.80	0.794	3.41	0.00833 (120×)	0.0956 (10.5×)
Encodec	6 kbps	1.04	1.80	0.793	3.51	0.00835 (120×)	0.0961 (10.4×)
HiFi-Codec	6 kbps	0.961	1.80	0.816	3.56	0.0100 (100×)	0.364 (2.75 ×)
APCodec	1.5 kbps	0.881	1.81	0.810	3.55	**0.00521 (192×)**	0.0634 (15.8 ×)
APCodec	3 kbps	0.858	1.80	0.833	3.72	0.00542 (185×)	0.0655 (15.3 ×)
APCodec	4.5 kbps	0.832	1.70	0.864	3.92	0.00562 (178×)	0.0704 (14.2 ×)
APCodec	6 kbps	**0.818**	**1.68**	**0.875**	**3.93**	0.00631 (158×)	0.0717 (13.9 ×)

are individually painted for the empty high frequency bands by parallel amplitude and phase extension streams. These two streams interact with each other and are both based on ConvNeXt network [22], without any upsampling/downsampling components. Finally, the high-sampling-rate waveform $\hat{\boldsymbol{x}}_H$ at S kHz is reconstructed via ISTFT.

2.5. Training Process

During the training phase, the encoder, quantizer, and decoder illustrated in Figure 1 are trained in accordance with the training guidelines of APCodec [18]. Meanwhile, the bandwidth recovery model in Figure 1 follows the training guidelines of AP-BWE [19]. Joint training of the codec model and the bandwidth recovery model will also be part of our future work.

3. Experiments

3.1. Experimental conditions

A subset of the VCTK-0.92 speech corpus [23] used in APCodec [18] was adopted in the experiments[1]. We used 40,936 utterances from 100 speakers for training, while the test set comprised 2,937 utterances from 8 unseen speakers. The sampling rate of this corpus was 48 kHz (i.e., $S = 48$). The downsampling and upsampling rates of the two-stage-downsampling-based encoder and two-stage-upsampling-based decoder were both $R = 6$ and $D = 320$. Therefore, the low-sampling-rate waveform used for discretization was at 8 kHz. When extracting the spectra from natural waveforms, the frame size was 320 samples, the frame shift was 40 samples (i.e., $W = 40$), and the FFT point number was 1024 (i.e., $N = 1024$). The stride for downsampling convolutional layer and upsampling deconvolutional layer was $U = 8$. The quantizer consisted of 4 VQs (i.e., $Q = 4$), each with a codebook size of 1024×32 (i.e., $M = 1024$ and $K = 32$). Therefore, according to Equation 1, the bitrate was only 1 kbps. Other configurations remain consistent with APCodec [18] and AP-BWE [19].

3.2. Evaluation Results

To evaluate the bitrate savings of the proposed codec framework, at a sampling rate of 48 kHz, we compared the framework at 1 kbps using APCodec and AP-BWE (denoted as APCodec+AP-BWE) with baseline SoundStream [10], Encodec [11] and HiFi-Codec [12] at 6 kbps. The baselines were reproduced by source codes[2]. The log-spectral distance (LSD), anti-wrapping phase distance of instantaneous phase (AWPD_{IP})

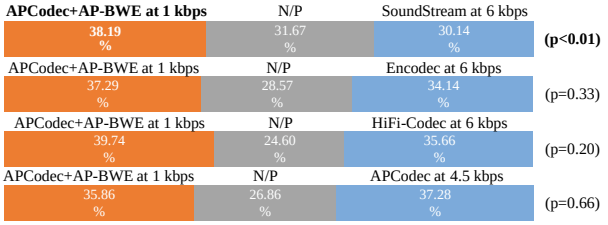

Figure 2: *Average preference scores (%) of ABX tests on audio quality, where N/P stands for "no preference" and p denotes the p-value of a t-test between two models.*

and short-time objective intelligibility (STOI) used in [18] were used as objective metrics for evaluating amplitude quality, phase quality and intelligibility, respectively. UTMOS[3] [24], an objective mean opinion score (MOS) prediction system, was used to evaluate the overall audio quality. We also introduced real-time factor (RTF) [18], an efficiency evaluation metric which indicates the required seconds to generate one second of audio on a NVIDIA GeForce RTX 4090 GPU or a Intel(R) Xeon(R) Silver 4310 CPU. In terms of the subjective evaluation, we conducted ABX preference tests on the Amazon Mechanical Turk platform[4] to compare the differences between APCodec+AP-BWE and three baseline codecs. In each ABX test, 20 utterances were randomly selected from the test set decoded by two comparative models and evaluated by at least 30 native English listeners. The listeners were asked to judge which utterance in each pair had better quality or whether there was no preference. In addition to calculating the average preference scores, the p-value of a t-test was used to measure the significance of the difference between two models.

The objective experimental results are shown in Table 1. The overall amplitude quality of the APCodec+AP-BWE was significantly better than that of SoundStream, Encodec and HiFi-Codec, according to the results of LSD. This indicates that the high-frequency amplitude spectrum extended by AP-BWE was satisfactory compared to these baselines. Although APCodec+AP-BWE was similar to other baselines in phase quality, it exhibited significant advantages in intelligibility (i.e., STOI) and overall audio quality (i.e., UTMOS). The APCodec+AP-BWE also demonstrated outstanding efficiency. Particularly, on a CPU without parallel acceleration, its generation speed was twice that of SoundStream and Encodec, and 7.5 times that of HiFi-Codec. This reflects the advantages of spec-

[1]Audio samples of the proposed framework can be accessed at https://yangai520.github.io/APCodec_APBWE.

[2]https://github.com/yangdongchao/AcademiCodec.

[3]https://github.com/sarulab-speech/UTMOS22.

[4]https://www.mturk.com.

Table 2: *Objective comparison between 8 kHz audios decoded by APCodec+AP-BWE at 1 kbps and APCodec at 6 kbps.*

	LSD↓	AWPD_{IP}↓	STOI↑	UTMOS↑
Natural	–	–	–	3.75
APCodec+AP-BWE at 1 kbps	**0.786**	**1.48**	**0.843**	**3.71**
APCodec at 6 kbps	0.823	1.60	0.843	3.64

tral modeling over direct waveform modeling. As shown in the ABX test results in Figure 2, in terms of human perception, the APCodec+AP-BWE outperformed SoundStream significantly ($p < 0.01$), but showed no significant difference compared to Encodec ($p = 0.33$) and HiFi-Codec ($p = 0.20$), despite its better objective results. This indicates that our proposed codec framework saved bitrates by at least 6 times and improved generation efficiency compared to these baseline codecs.

To assess the bitrate savings of the proposed APCodec+AP-BWE compared to the original APCodec and verify the effects of bandwidth reduction and recovery, we compared the APCodec+AP-BWE at 1 kbps with APCodec directly operated at 1.5 kbps, 3 kbps, 4.5 kbps, and 6 kbps (setting Q as 1, 2, 3 and 4, respectively). The objective experimental results are also shown in Table 1. Unfortunately, the LSD of the proposed APCodec+AP-BWE was the worst. We speculate that this may be influenced by the high-frequency amplitude spectrum extended by AP-BWE, which will be discussed in Section 3.3. Although the APCodec+AP-BWE performed worse than APCodec at $>$ 4.5 kbps on AWPD_{IP} and STOI, it achieved the same highest UTMOS score as APCodec at 6 kbps, trailing natural audio by only 0.11. To gather more evidence regarding the positioning of APCodec+AP-BWE, we conducted a set of ABX preference test between APCodec+AP-BWE at 1 kbps and APCodec at 4.5 kbps. The results are depicted in Figure 2. It can be observed that there was no significant difference ($p = 0.66$) between them. These results indicated that our proposed codec framework achieved the performance previously attained at 4.5 kbps, now at just 1 kbps, resulting in a 4.5-fold savings in bitrate. In terms of efficiency in Table 1, on GPU, the APCodec+AP-BWE was slightly slower compared to APCodec at 4.5 kbps but faster compared to APCodec at 6 kbps. Surprisingly, on CPU, the generation speed of APCodec+AP-BWE was the fastest, achieving 20.5 times real-time speed. A possible reason is that the APCodec operates at 8 kHz, which theoretically leads to a sixfold efficiency improvement compared to operating at 48 kHz, and the AP-BWE [19] is also a CPU-friendly efficient model. Therefore, despite the incorporation of AP-BWE, our proposed codec framework still maintains a high generation efficiency especially on CPU.

3.3. Analysis and Discussion

In this section, we conducted a detailed analysis and discussion of the performance of the proposed codec framework across different frequency bands. We first downsampled the waveform decoded by APCodec+AP-BWE at 1 kbps and APCodec at 6 kbps to sampling rate of 8 kHz for comparing their low-frequency component (0∼4 kHz frequency band). Indeed, the decoded 8 kHz audio from APCodec+AP-BWE is identical to that from a 8 kHz APCodec (i.e., $\hat{\boldsymbol{x}}_L$), serving as the baseband for bandwidth recovery. Objective experimental results are listed in Table 2. We can see that, for all metrics, the 8 kHz audio decoded by APCodec+AP-BWE consistently outperformed that of APCodec at 6 kbps. In addition, we plotted the low-frequency spectrograms of an example utterance in Figure 3. It's evident that for APCodec at 6 kbps, there were artifacts between low-frequency harmonics (e.g., around 0.9 s), and the harmonics appeared more blurred. However, the result of APCodec+AP-BWE was closer to the natural one (the difference in UTMOS was only 0.04). Therefore, the low-frequency amplitude quality of APCodec+AP-BWE was high, but according to the results in Table 1, its overall LSD performance was poor. This suggests that the quality of high-frequency amplitude extended by AP-BWE constrained the LSD metric, but it did not significantly affect subjective listening experience.

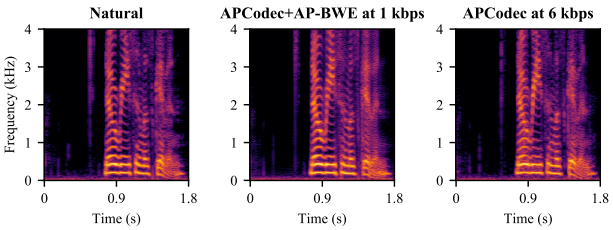

Figure 3: *Spectrogram comparison at the 0∼4 kHz frequency band for natural audio and audios decoded by APCodec+AP-BWE at 1 kbps and APCodec at 6 kbps.*

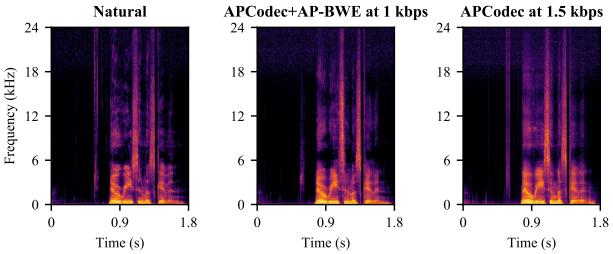

Figure 4: *Spectrogram comparison for natural audio and audios decoded by APCodec+AP-BWE at 1 kbps and APCodec at 1.5 kbps.*

Finally, we compared the full-band spectrograms of APCodec+AP-BWE at 1 kbps and APCodec at 1.5 kbps in Figure 4. Although the latter had a lower LSD than the former, the high-frequency spectrogram of the former was noticeably more natural, while the latter exhibited excessive harmonics. Nevertheless, from Figure 4, we can observe that at certain unvoiced positions (around 0.6 s), the energy of the high-frequency band extended by AP-BWE was relatively low. This could be a contributing factor to the poor LSD performance, but it did not affect the listening experience.

4. Conclusion

In this paper, we have proposed a novel codec framework, which incorporated bandwidth reduction and recovery, aiming to apply to high sampling rate and low bitrate scenarios. The core of this framework lay in discretizing low-sampling-rate waveforms by codec model to significantly reduce the bitrate, and ultimately recovering the missing bandwidth. In the experiment, we combined the codec model APCodec with the BWE model AP-BWE. Objective and subjective experiment results confirmed that this combination could save 6 times the bitrate, achieving audio compression at just 1 kbps under a 48 kHz sampling rate. Applying this framework to downstream tasks and reducing the latency of this framework will be our future work.

5. References

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

[2] R. Salami, C. Laflamme, J.-P. Adoul, and D. Massaloux, "A toll quality 8 kb/s speech codec for the personal communications system (pcs)," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 808–816, 1994.

[3] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[4] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "AudioLM: A language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2523–2533, 2023.

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

[6] 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 *Proc. ICASSP*, 2018, pp. 676–680.

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

[8] 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.

[9] 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*, 2021, pp. 66–70.

[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. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[12] D. Yang, S. Liu, R. Huang, J. Tian, C. Weng, and Y. Zou, "HiFi-Codec: Group-residual vector quantization for high fidelity audio codec," *arXiv preprint arXiv:2305.02765*, 2023.

[13] H. S. Black and J. Edson, "Pulse code modulation," *Transactions of the American Institute of Electrical Engineers*, vol. 66, no. 1, pp. 895–899, 1947.

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

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

[16] Y.-C. Wu, D. Marković, S. Krenn, I. D. Gebru, and A. Richard, "ScoreDec: A phase-preserving high-fidelity audio codec with a generalized score-based diffusion post-filter," *arXiv preprint arXiv:2401.12160*, 2024.

[17] G. Davidson, M. Vinton, P. Ekstrand, C. Zhou, L. Villemoes, and L. Lu, "High quality audio coding with MDCTNet," in *Proc. ICASSP*, 2023, pp. 1–5.

[18] Y. Ai, X.-H. Jiang, Y.-X. Lu, H.-P. Du, and Z.-H. Ling, "APCodec: A neural audio codec with parallel amplitude and phase spectrum encoding and decoding," *arXiv preprint arXiv:2402.10533*, 2024.

[19] Y.-X. Lu, Y. Ai, H.-P. Du, and Z.-H. Ling, "Towards high-quality and efficient speech bandwidth extension with parallel amplitude and phase prediction," *arXiv preprint arXiv:2401.06387*, 2024.

[20] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "ConvNeXt v2: Co-designing and scaling convnets with masked autoencoders," in *Proc. CVPR*, 2023, pp. 16 133–16 142.

[21] Y. Ai and Z.-H. Ling, "Neural speech phase prediction based on parallel estimation architecture and anti-wrapping losses," in *Proc. ICASSP*, 2023, pp. 1–5.

[22] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. CVPR*, 2022, pp. 11 976–11 986.

[23] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2019.

[24] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "UTMOS: Utokyo-sarulab system for voiceMOS Challenge 2022," *arXiv preprint arXiv:2204.02152*, 2022.