LOW-COMPLEXITY END-TO-END SPEECH ENHANCEMENT CODEC FOR REAL-TIME COMMUNICATION IN NOISY AND REVERBERANT CONDITIONS

Pincheng Lu *Peng Zhou* *Xiaojiao Chen* *Jing Wang*

Beijing Institute of Technology, Beijing, China

ABSTRACT

End-to-end speech codecs enable efficient low-bitrate communication, but most existing approaches lack integrated enhancement, which limits performance under noisy and reverberant conditions. While recent work has attempted to combine speech enhancement with neural codecs, these methods are often too complex to be practical in low-resource scenarios. In this paper, we present a lightweight speech enhancement codec specifically designed for resource-constrained settings. The proposed system adopts a three-stage training strategy that first establishes strong compression capability and then progressively improves robustness to noise and reverberation. Experimental results demonstrate that our model achieves superior performance in challenging noisy and reverberant environments while meeting strict constraints on computational complexity, latency, and bitrate.

***Index Terms*—** low complexity, speech codec, speech enhancement

1. INTRODUCTION

Speech codecs compress speech signals while preserving perceptual quality [1]. Recent end-to-end models such as SoundStream [2], DAC [3], and L3AC [4] employ encoder–decoder architectures with quantization modules like RVQ or FSQ [5], achieving high-quality reconstruction. However, as most are trained only on clean speech, they lack robustness to real-world noise, making integrated enhancement essential for practical deployment.

Joint enhancement–compression has thus emerged as an active research direction. Early approaches, such as SoundStream and SEStream [6], were trained directly on noisy–clean pairs. More recent methods have explored the use of domain-specific codebooks [7], masked generative models [8, 9], or latent space regression within pretrained codecs [10, 11]. While these approaches have demonstrated promising performance, they often come with high computational complexity, which hinders their applicability in real-time, resource-constrained scenarios.

To address these limitations, we propose a Lightweight Codec for Joint speech compression and enhancement (LJCodec), an end-to-end framework designed to perform both tasks within a unified system. The main contributions of this work are summarized as follows.

- We propose LJCodec, a Lightweight Codec that jointly performs speech compression and enhancement.
- We propose a three-stage training strategy that strengthens noise robustness by training on clean speech, aligning encoder representations from noisy to clean embeddings, and adapting the decoder with the fixed encoder.

2. METHOD

2.1. Model Architecture

The entire model follows the same structure as the baseline. The **encoder** consists of five EncoderBlocks, each composed of several residual convolutional blocks followed by a strided convolution for downsampling. The downsampling factors across the five blocks are 2, 2, 3, 4, and 5, respectively. The **quantizer** employs Residual Vector Quantization (RVQ), where multiple codebooks are cascaded such that each deeper codebook encodes the residual of the previous one. The **decoder** mirrors the encoder architecture and performs upsampling using transposed convolutions with stride equal to the kernel size, thereby reducing the complexity introduced by the upsampling operations. To reduce the computational burden at the receiver side and satisfy LRAC requirements, the convolutional channel width in the decoder is set to about 3/4 of that in the encoder.

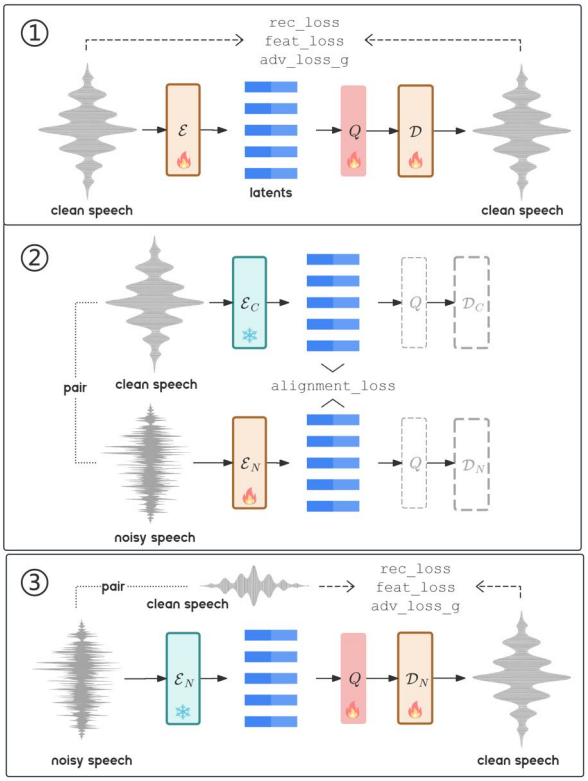

Fig. 1: Proposed stage-wise training strategy.

2.2. Stage-wise Training

To improve robustness against noisy speech, we employ a three-stage training strategy (Fig. 1), starting with clean speech training and followed by independent fine-tuning of the encoder and decoder.

Stage 1. Base Model Training on Clean Speech. In the first stage, we train the codec model exclusively on clean speech using a combination of reconstruction loss, feature loss, commitment loss, and adversarial loss, following the same loss setup and adversarial training strategy as EnCodec.

Let x be the speech to be encoded, and $\hat{x}$ be the speech generated by the decoder. Reconstruction loss is used to measure the difference between $\hat{x}$ and x in both the time domain and the time-frequency domain. The loss in the time and time-frequency domains can be expressed as

$$\ell_t = ||x - \hat{x}||_2^2, \tag{1}$$

and

$$\ell_f = \sum_{s\in\{2^6,\ldots,2^{11}\}} \sum_t ||S_t^s(x) - S_t^s(\hat{x})||_1 + ||\log S_t^s(x) - \log S_t^s(\hat{x})||_2, \tag{2}$$

respectively, where S_t^s represents the t-th frame in the 64-bin mel-spectrogram with window length s and hop length $s/4$. The reconstruction loss ℓ_{rec} is the sum of the time domain loss and the time-frequency domain loss:

$$\ell_{rec} = 100\ell_t(x, \hat{x}) + \ell_f(x, \hat{x}). \tag{3}$$

Feature loss ℓ_{feat} measures the difference between x and $\hat{x}$ in the feature space defined by the discriminators. It is calculated by taking the mean absolute difference between the inner layer output feature maps of the discriminators for the generated speech and the corresponding target speech.

$$\ell_{feat} = E_x \left[\frac{1}{KL} \sum_{k,l} |\mathcal{D}_{k,l}(x) - \mathcal{D}_{k,l}(\hat{x})| \right], \tag{4}$$

where L is the number of intermediate layers, and $\mathcal{D}_{k,l}$ ($l \in \{1, \ldots, L\}$) denotes the output of the l-th layer of discriminator k.

Quantizer commitment loss ℓ_q describes the difference between the input and output of the quantizer. It is used to reduce the discrepancy between the quantizer's embedding space and the encoder's output, which can be expressed by:

$$\ell_q = \sum_{c=1}^{C} ||z_c - q_c(z_c)||_2^2, \tag{5}$$

where q_c represents the c-th vector quantizer.

In adversarial training, the following two adversarial losses are used to optimize the codec and the discriminators. The adversarial loss ℓ_{adv_g} for codec is

$$\ell_{adv\_g} = E_x \left[(1 - \mathcal{D}(\hat{x}))^2 \right], \tag{6}$$

while ℓ_{adv_d} for discriminators is

$$\ell_{adv\_d} = E_x \left[(1 - \mathcal{D}(x))^2 + (1 + \mathcal{D}(\hat{x}))^2 \right]. \tag{7}$$

The total loss for the codec is defined as follows:

$$\ell_g = \lambda_{rec}\ell_{rec} + \lambda_{feat}\ell_{feat} + \lambda_q\ell_q + \lambda_{adv\_g}\ell_{adv\_g}, \tag{8}$$

Table 1: Objective evaluation results at 1 kbps and 6 kbps under clean, noisy, and reverberant conditions.

	ScoreQ	UTMOS	PESQ
1 kbps			
clean	0.39	3.99	1.58
noisy	0.49	3.82	1.44
reverb	0.52	3.61	1.27
6 kbps			
clean	0.27	4.17	2.21
noisy	0.45	3.96	1.77
reverb	0.5	3.63	1.38

Table 2: Computational complexity (MFLOPS) and latency (ms) of different modules.

Component	Compute	Latency
Encoder	1946	20
Quantizer	48	0
Decoder	594	20
Buffering latency	–	10
Total	2588	50

and the discriminator loss ℓ_d is

$$\ell_d = \lambda_{adv\_d}\ell_{adv\_d}. \tag{9}$$

where λ are constant weights used to balance each component.

In our experiments, we trained the model with weights $\lambda_{rec} = \lambda_{feat} = \lambda_{adv\_g} = \lambda_{adv\_d} = 1$, and $\lambda_q = 1000$.

Stage 2. Encoder Alignment Fine-tuning. Inspired by SoundStream, we argue that the enhancement task should be performed before quantization, on the encoder side, to minimize the impact of noisy latent representations on both the quantizer and decoder. Unlike NoiseRobustVRVQ (NRVRVQ) [11], which optimizes the entire model on noisy speech, we perform alignment fine-tuning only on the encoder.

Specifically, we duplicate all modules before the quantizer into a trainable encoder, denoted as $\mathcal{E}_N$, and a frozen encoder, denoted as $\mathcal{E}_C$. Noisy speech x_n is fed into $\mathcal{E}_N$, while clean speech x_c is fed into $\mathcal{E}_C$. The output of $\mathcal{E}_C$ serves as the supervision target for $\mathcal{E}_N$. The $\mathcal{E}_N$ is optimized with a mean squared error loss:

$$\ell_a = \mathbb{E}\left[(\mathcal{E}_N(x_n) - \mathcal{E}_C(x_c))^2 \right]. \tag{10}$$

No additional losses (e.g., reconstruction loss) are introduced, as this design forces the encoder to rapidly adapt to the speech enhancement task on top of its established compression capability.

Stage 3. Decoder Adaptive Fine-tuning. Although the latent distribution after Stage 2 is close to that of Stage 1, slight mismatches remain and lead to reconstruction artifacts. We fine-tune both the quantizer and the decoder to better adapt to these new representations. The encoder $\mathcal{E}_N$ is frozen, while the quantizer $\mathcal{Q}$, decoder $\mathcal{D}_N$, and the discriminators are optimized using the same loss functions as in Stage 1. This strategy improves the overall audio quality with minimal overhead.

3. EXPERIMENT

3.1. Datasets

We trained our codec using the datasets specified by LRAC. In Stage 1, the model was trained on clean speech drawn from LibriVox data from the DNS5 Challenge [12], LibriTTS[13], VCTK[14], EARS[15], CommonVoice[16], and Multilingual LibriSpeech[17].

In Stage 2 and Stage 3, we constructed degraded speech by mixing clean utterances with noise and reverberation. The noise sources included Audioset[18] and FreeSound[19] noises from the DNS5 Challenge, WHAM! noise[20], speech-filtered FSD50K[21], and Free Music Archive[22]. Noisy speech was synthesized by mixing clean utterances with these noises at signal-to-noise ratios (SNR) uniformly sampled between -5 dB and 30 dB. Reverberation was simulated using RIR datasets from OpenSLR28, the DNS5 Challenge, and Motus [23]. All corpora were downsampled to 24 kHz for both training and evaluation. For benchmarking, we used the official LRAC validation and test sets to ensure fair and consistent comparisons.

3.2. Training and Evaluation Settings

Training Settings: The entire model is trained on a single RTX 4090 GPU with a batch size of 32. The number of iterations for Stage 1, Stage 2, and Stage 3 are set to 150k, 50k, and 150k, respectively.

Evaluation Metrics: For preliminary offline testing during the development stage, we adopt PESQ[24], UTMOS[25], and ScoreQ[26] as objective quality metrics. For the official benchmark evaluation, we rely on the toolkit provided by the organizers, which reports a more comprehensive set of metrics, including `sheet_ssqa` [27], `scoreq_ref`, `audiobox_AE_CE` [28], `utmos`, and `pesq`. For model efficiency, we report both the computational complexity and the latency of the proposed codec.

3.3. Speech Quality Metrics

Table 1 summarizes the objective evaluation results. On clean speech compression, LJCodec outperforms the baseline at both 1 kbps and 6 kbps. For degraded speech with additive noise and reverberation, LJCodec also demonstrates consistent improvements over the baseline.

3.4. Model Efficiency

Table 2 presents the computational complexity and latency of our model. The overall complexity is below 2600 MFLOPS, with the receive-side (decoder) complexity under 600 MFLOPS. The end-to-end latency is less than 50 ms, fully meeting the challenge requirements.

4. CONCLUSIONS

We presented **LJCodec**, a low-complexity end-to-end codec that jointly performs speech compression and enhancement. Through a three-stage training strategy, the model achieves robustness to noise and reverberation while maintaining a low bitrate, low latency ($<$50 ms), and low computational complexity ($<$2600 MFLOPS). Experiments on the LRAC benchmark show consistent improvements over the baseline, demonstrating the practicality of LJCodec for real-world low-resource speech communication.

5. REFERENCES

[1] J. Wang, L. Xu, X. Chen *et al.*, "Research review on low bit rate speech coding technology based on neural networks," *Journal of Signal Processing*, vol. 40, no. 12, pp. 2261–2280, 2024.

[2] 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.

[3] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[4] L. Zhai, H. Ding, C. Zhao, fei wang, G. Wang, W. Zhi, and W. Xi, "L3ac: Towards a lightweight and lossless audio codec," 2025. [Online]. Available: https://arxiv.org/abs/2504.04949

[5] F. Mentzer, D. Minnen, E. Agustsson, and M. Tschannen, "Finite scalar quantization: Vq-vae made simple," *arXiv preprint arXiv:2309.15505*, 2023.

[6] J. Huang, Z. Yan, W. Jiang, and F. Wen, "A two-stage training framework for joint speech compression and enhancement," *arXiv preprint arXiv:2309.04132*, 2023.

[7] X. Bie, X. Liu, and G. Richard, "Learning source disentanglement in neural audio codec," in *IEEE International Conference on Acoustic, Speech and Signal Procssing (ICASSP)*, 2025.

[8] H. Xue, X. Peng, and Y. Lu, "Low-latency speech enhancement via speech token generation," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 661–665.

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

[10] H. Li, J. Q. Yip, T. Fan, and E. S. Chng, "Speech enhancement using continuous embeddings of neural audio codec," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[11] Y. Chae and K. Lee, "Towards bitrate-efficient and noise-robust speech coding with variable bitrate rvq," *arXiv preprint arXiv:2506.16538*, 2025.

[12] H. Dubey, A. Aazami, V. Gopal, B. Naderi, S. Braun, R. Cutler, H. Gamper, M. Golestaneh, and R. Aichner, "Icassp 2023 deep noise suppression challenge," in *ICASSP*, 2023.

[13] 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," *arXiv preprint arXiv:1904.02882*, 2019.

[14] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[15] J. Richter, Y.-C. Wu, S. Krenn, S. Welker, B. Lay, S. Watanabe, A. Richard, and T. Gerkmann, "Ears: An anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation," *arXiv preprint arXiv:2406.06185*, 2024.

[16] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[17] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *arXiv preprint arXiv:2012.03411*, 2020.

[18] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[19] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," 2017.

[20] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "Wham!: Extending speech separation to noisy environments," *arXiv preprint arXiv:1907.01160*, 2019.

[21] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[22] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "Fma: A dataset for music analysis," *arXiv preprint arXiv:1612.01840*, 2016.

[23] G. Götz, S. J. Schlecht, and V. Pulkki, "A dataset of higher-order ambisonic room impulse responses and 3d models measured in a room with varying furniture," in *2021 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*. IEEE, 2021, pp. 1–8.

[24] I.-T. Recommendation, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[25] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "Utmos: Utokyo-sarulab system for voicemos challenge 2022," in *Proc. Interspeech*, 2022, pp. 4521–4525.

[26] A. Ragano, J. Skoglund, and A. Hines, "Scoreq: Speech quality assessment with contrastive regression," *arXiv preprint arXiv:2410.06675*, 2024.

[27] W.-C. Huang, E. Cooper, and T. Toda, "Mos-bench: Benchmarking generalization abilities of subjective speech quality assessment models," *arXiv preprint arXiv:2411.03715*, 2024.

[28] A. Tjandra, Y.-C. Wu, B. Guo, J. Hoffman, B. Ellis, A. Vyas, B. Shi, S. Chen, M. Le, N. Zacharov *et al.*, "Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound," *arXiv preprint arXiv:2502.05139*, 2025.