ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11462151

UJCODEC: AN END-TO-END UNET-STYLE CODEC FOR JOINT SPEECH COMPRESSION AND ENHANCEMENT

Pincheng Lu[⋆] *Peng Zhou*[⋆] *Xiaojiao Chen*[⋆] *Jing Wang*[⋆] *Zhong-Qiu Wang*[†]

[⋆] Beijing Institute of Technology, Beijing, China
[†] Southern University of Science and Technology, Shenzhen, China

ABSTRACT

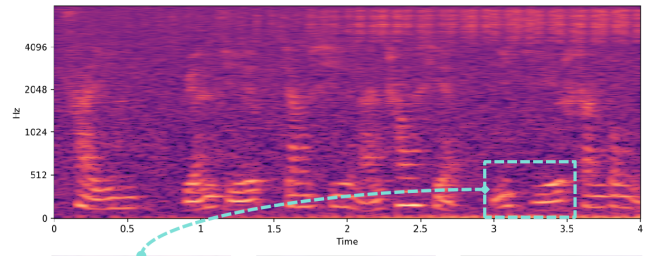
End-to-end speech codecs enable low-bitrate communication but most of them typically lack integrated enhancement, limiting performance in noisy conditions. Although recent efforts have explored integrating speech enhancement into neural codecs, the decoded speech often exhibits noticeable distortions, with word omissions particularly pronounced, leading to significant degradations in intelligibility. To address this, we first propose a UNet-style model with skip connections designed to retain speech details throughout the coding process. We then employ a three-stage training strategy that builds on the codec's compression capability while rapidly enhancing its robustness to noise. To further mitigate word omission distortions, we simulate latent frame corruption during training. Experiments show that UJCodec achieves strong noise suppression and high intelligibility under low-bitrate, noisy scenarios.

Index Terms— UNet, speech codec, speech enhancement

1. INTRODUCTION

A speech codec compresses speech signals while preserving perceptual quality [1]. Recent end-to-end models, such as SoundStream [2], Descript Audio Codec (DAC) [3], and L3AC [4], use encoder-decoder architectures with Residual Vector Quantization (RVQ) or Finite Scalar Quantization (FSQ) [5] to achieve high-quality reconstruction. However, most existing neural speech codecs implicitly assume noise-free input, trained exclusively on clean speech. This limits their robustness, as real-world speech often contains background noise that can severely degrade performance. Therefore, integrating speech enhancement into neural speech codecs is particularly important for real-world applications.

Several approaches, such as Tencent Games Voice[1], use separate modules for enhancement and compression. While this modular design offers flexibility, it can introduce additional latency due to the cascaded structure, and the overall quality is often constrained by the weaker of the two components [6]. The other approach performs end-to-end joint enhancement and compression within one codec, which has emerged as an active area of research. SoundStream and SEStream [7] explored joint compression and enhancement by training the codec directly on paired noisy-clean speech data. SDCodec [8] introduces multiple domain-specific codebooks to model different audio types (e.g., speech, music, sound effects), enabling better enhancement and separation performance. Some methods like [9, 10] adopt masked generative models, predicting clean acoustic tokens from noisy input, while others [11, 12] treat enhancement as latent space regression within a pretrained codec, generating embeddings of clean speech from noisy speech. While these approaches can be effective in specific scenarios, they still suffer from noticeable quality degradation compared to dedicated enhancement models, with **word omissions** being a particularly severe issue. As illustrated in Fig. 1, models exhibiting severe word omissions often misinterpret speech segments as noise and remove them, resulting in extremely weak formant energy in the spectrogram. To listeners, this resembles silent segments, and such distortions severely degrade intelligibility.



(a) noisy input (b) word retaining (c) word omitting

Fig. 1: Illustration of word omission. The top Mel-spectrogram shows the noisy input. Subplot (a) zooms into a region where omissions frequently occur; (b) shows an enhanced output that retains the word; and (c) depicts a case of word omission, where the remaining speech energy is very weak.

To address these limitations, we propose a **U**Net-style Codec for **J**oint speech compression and enhancement (UJCodec), a novel end-to-end framework designed to perform both tasks within a unified system.[2] The main contributions of this work are summarized as follows.

- We propose UJCodec, a UNet-style 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.
- To address the issue of word omissions, we further simulate latent frame corruption, encouraging the decoder to reconstruct intelligible speech under degraded latent.

2. METHOD

2.1. Model Architecture

The architecture of our model is shown in Fig.2. Inspired by the UNet−−[13], the encoder progressively downsamples intermediate feature maps to match the frame rate of the final latent representation. These multi-scale features are fused through a dedicated Fusion Module, and the decoder mirrors the encoder with skip connections

[1] https://intl.gcloud.tencent.com/pages/products/gvoice.html

[2] Demo page is available at https://ukitenzai.github.io/UJCodec.demopage

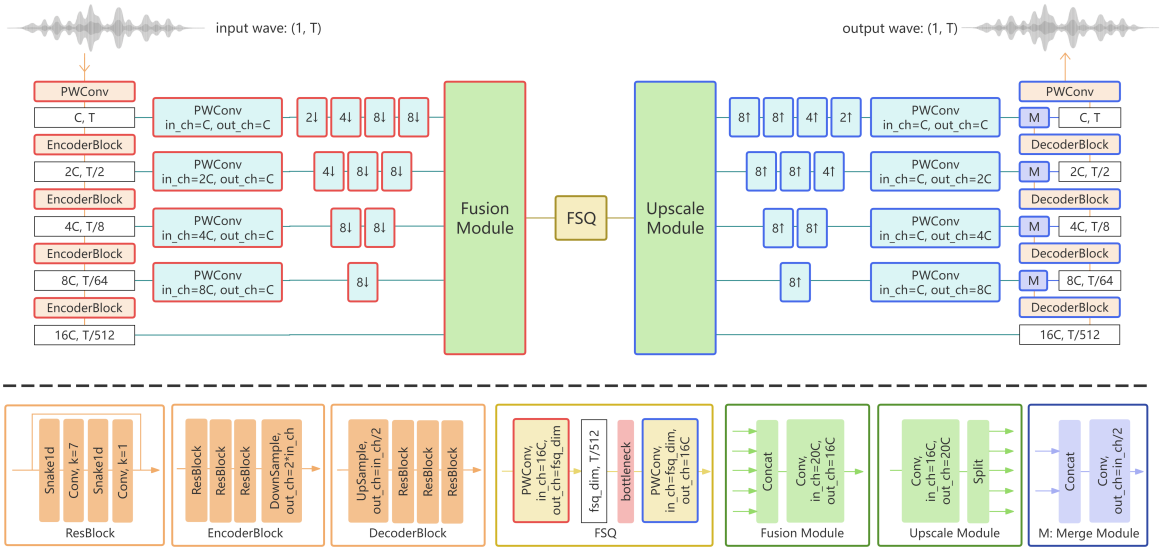


Fig. 2: **Network architecture**. Red-outlined modules are encoder components $\mathcal{E}_N$ optimized in Stage 2, blue-outlined modules are decoder components $\mathcal{D}_N$ optimized in Stage 3, and black-bordered boxes represent intermediate feature maps. Upward arrows (↑) denote upsampling via transposed convolution, while downward arrows (↓) indicate downsampling using strided convolution.

that link corresponding layers. Such skip connections maintain low-level features and high-level information, which is particularly beneficial for speech enhancement. In addition, we adopt FSQ as the discrete bottleneck, which not only improves reconstruction quality but also facilitates alignment objectives in subsequent training stages.

2.2. Stage-wise Training

To improve robustness against noisy speech, we employ a three-stage training strategy (Fig. 3), 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, and adversarial loss, following the same loss setup and adversarial training strategy as DAC [3]. As our model uses FSQ, no commitment loss is needed.

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) [12], which optimizes the entire model on noisy speech, we perform alignment fine-tuning only on the encoder.

Specifically, we duplicate all modules before the quantization bottleneck (downsampling layers, residual blocks, fusion module, and the FSQ projection) 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 $\mathcal{E}_C$ is quantized by the quantization bottleneck $\mathrm{Q}(\cdot)$ into discrete latent embeddings, which serve as supervision targets for $\mathcal{E}_N$. The $\mathcal{E}_N$ is optimized with a mean squared error loss:

$$\ell_a = \mathbb{E}\left[(\mathcal{E}_N(x_n) - Q(\mathcal{E}_C(x_c)))^2\right]. \tag{1}$$

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.

We further compare two quantization schemes for $\mathrm{Q}(\cdot)$. RVQ uses a codebook $\mathcal{V}$ as the bottleneck: the input vector is matched to its nearest codebook vector v_j, and each output element z_i takes values from the coordinate-wise set formed by the codebook vectors,

$$z_i \in \{v_{j,i} \mid v_j \in \mathcal{V} \subset \mathbb{R}^m\}, \tag{2}$$

which typically spans a broad range. In contrast, FSQ projects the input vector into a lower-dimensional space, applies a scaled mapping $\tilde{y}_i = A\tanh(y_i)$ to restrict values to $(-A, A)$, and uniformly quantizes each element on a fixed grid:

$$z_i \in \{-A + kd \mid k = 0, \ldots, K-1\}, \tag{3}$$

where K is the number of quantization levels (determined by the bitrate), A is a scaling factor, and $d = 2A/(K-1)$ is the quantization interval. As illustrated in Fig. 4 (a) and (b), FSQ yields clearer element-wise optimization targets and more stable alignment compared to RVQ.

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 the decoder to better adapt to these new representations. The encoder $\mathcal{E}_N$ is frozen, and only the decoder $\mathcal{D}_N$ and discriminator are optimized using the same losses as in Stage 1. This improves overall audio quality with minimal overhead.

2.3. Simulating Latent Frames Corruption

In our framework, speech enhancement is mainly performed by the encoder $\mathcal{E}_N$. However, when clean speech components are mistakenly removed as noise, the latent frames output by $\mathcal{E}_N$ become damaged, leading to noticeable distortions. To make the decoder $\mathcal{D}_N$ more robust to such encoder errors, we simulate latent frame corruption during training.

This strategy is applied only during the final phases of Stage 1 and Stage 3. During Stage 1, we randomly replace a portion of the latent frames with either **(a) frames corresponding to silent segments** or **(b) frames randomly sampled from the same clean utterance**. The two types of replacements are sampled with equal probability. This helps the decoder to initially adapt to clean speech

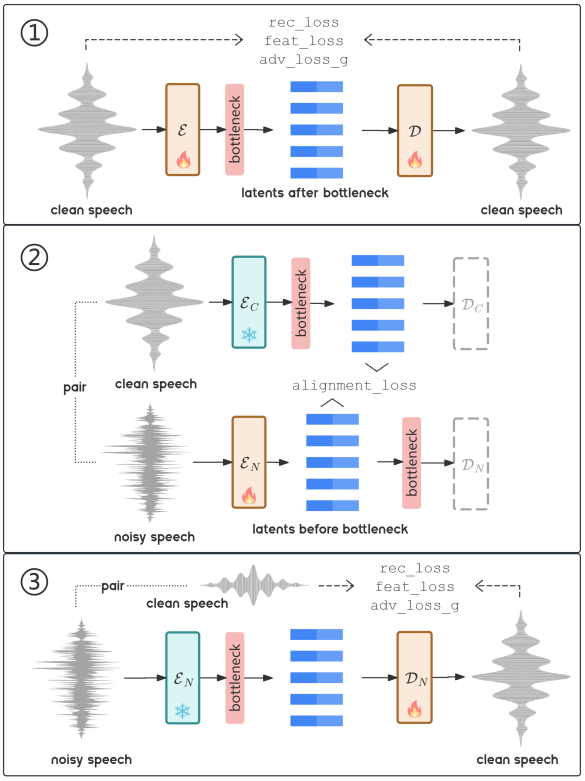


Fig. 3: Proposed stage-wise training strategy.

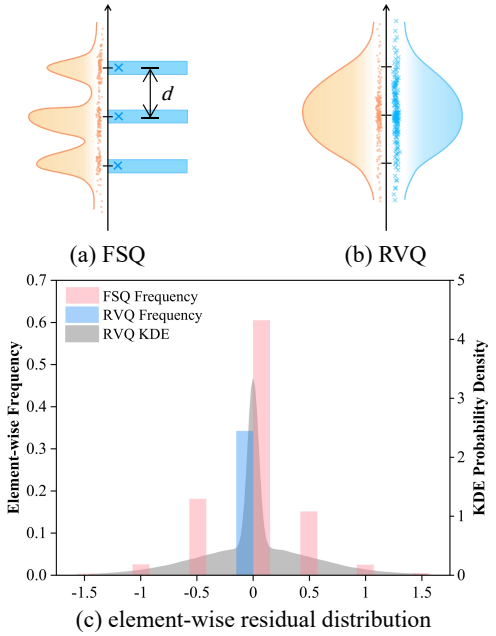


Fig. 4: Comparison of element-wise distributions in FSQ and RVQ. Subplots (a) and (b) show the ideal element-wise distributions of input vectors (left) and output vectors (right) for FSQ and RVQ, respectively. Subplot (c) presents the element-wise residual distribution after quantization with mild perturbation. For clearer comparison, we use blue bars to highlight the proportion of elements in RVQ that remain unchanged under perturbation (i.e., residuals are zero).

contexts with mild latent damage, reducing training difficulty and serving as a preparatory step for more severe distortions encountered in noisy environments. During Stage 3, we introduce more challenging replacements using pure noise frames or frames randomly sampled from noisy speech.

To ensure stable training, corruption is applied only in the last $20,000$ iterations of each stage, with the replacement ratio gradually increased to 5% over the first $10,000$ iterations and fixed thereafter.

3. EXPERIMENT

3.1. Datasets

We used a combination of multiple speech datasets for the different stages of training and evaluation of the proposed coding method. For **Stage 1**, we used the entire training sets from LibriTTS [16], VCTK [17], and AISHELL-3 [18], which cover both English and Mandarin clean speech. For **Stage 2** and **Stage 3**, we used the VoiceBank+DEMAND [19] and DNS-Challenge [20] datasets. The DNS-Challenge dataset consists of both reading English speech and Mandarin speech, combined with its provided noise clips library to synthesize noisy speech at signal-to-noise ratios uniformly sampled from -5 dB to 20 dB. All corpora were downsampled to 16 kHz sampling rate for training and evaluation. We used the official validation and test set provided by each dataset for evaluation 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 16. The number of iterations for Stage 1, Stage 2, and Stage 3 are set to 150k, 50k, and 50k, respectively. Latent Frames Corruption is applied only during the final 20k iterations of Stage 1 and 3.

Evaluation Metrics: For clean speech evaluation, we adopt PESQ [21] for objective quality evaluation and the MUSHRA methodology [22] for subjective assessment.

For noisy speech evaluation, we additionally report the Word Error Rate (WER), computed using Whisper-tiny [23], to better capture word-level intelligibility degradation caused by word omissions. We conduct subjective evaluations using the DNS-Challenge blind test set. Since clean references are unavailable, we directly conduct reference-free evaluation on the enhanced speech. Listeners are asked to rate each sample based on noise suppression, intelligibility, and naturalness, resulting in an overall MOS score [24].

A group of ten listeners, including five females and five males aged between 22 and 29, participated in the subjective listening tests. For clean speech, 20 utterances were randomly selected from the test set. For noisy speech, we selected 20 utterances from the DNS-Challenge noisy blind test set, including 10 synthetic and 10 real-recorded samples. We report the mean scores along with 95% confidence intervals.

For model efficiency, we report the parameter counts and real-time factor (RTF). The RTF is measured on a mobile-grade CPU: Intel(R) Core(TM) i5-10300H @ 2.50GHz.

Baselines: We compare our proposed method with several existing end-to-end joint approaches, including SDCodec and NRVRVQ. We use only the separated speech audio track from SDCodec, and configure NRVRVQ in constant bitrate (CBR) mode to align with the fixed bitrate setting of other codecs. To evaluate joint versus cascaded architectures, we build baselines by cascading either an FSQ-based codec (L3AC) or a commonly used RVQ-based codec (DAC) with two enhancement modules: MP-SENet [15], which provides high quality but lacks real-time efficiency, and GTCRN [14], which supports real-time inference at reduced quality. These cascaded systems are denoted as G-L3AC, G-DAC, M-L3AC, and M-DAC.

Table 1: Objective evaluation of joint speech compression and enhancement performance. The enhancement modules in G-L3AC and M-L3AC are activated only for noisy speech inputs and are disabled for clean speech compression.

Model	Bitrate (bps)	Clean	Noisy	
		PESQ↑	PESQ↑	WER(%)↓
UJCodec	750	**2.093**	**1.793**	13.89
SDCodec [8]	750	1.786	1.626	14.77
NRVRVQ [12]	750	1.927	1.697	14.68
G-L3AC [14, 4]	750	1.894	1.556	16.24
M-L3AC [15, 4]	750	1.894	1.704	**13.61**
G-DAC [14, 3]	750	1.774	1.506	15.37
M-DAC [15, 3]	750	1.774	1.577	14.54
UJCodec	3k	**3.091**	**2.711**	**11.34**
SDCodec	3k	2.892	2.392	12.63
NRVRVQ	3k	2.930	2.480	12.08
G-L3AC	3k	2.853	2.293	13.17
M-L3AC	3k	2.853	2.693	11.44
G-DAC	3k	2.875	2.153	13.09
M-DAC	3k	2.875	2.687	11.55
UJCodec	6k	**3.572**	**3.152**	**9.95**
SDCodec	6k	3.392	2.802	10.75
NRVRVQ	6k	3.440	2.925	10.44
G-L3AC	6k	3.428	2.663	11.04
M-L3AC	6k	3.428	3.063	10.15
G-DAC	6k	3.431	2.887	11.24
M-DAC	6k	3.431	3.051	10.22

3.3. Speech Quality Metrics

Table 1 shows the objective results. On clean speech compression, UJCodec achieves clearly higher perceptual quality than all the baselines. For the joint compression and enhancement task on noisy speech, the G-L3AC vs. M-L3AC comparison confirms the key role of the enhancement front-end—stronger modules yield better quality and intelligibility. Yet, UJCodec surpasses both end-to-end and cascaded systems without any external enhancement. Fig. 5 reports the subjective results: across bitrates and tasks, UJCodec consistently outperforms baselines, with the advantage most pronounced at the challenging 750 bps.

3.4. Model Efficiency

Fig. 6 illustrates the relationship between model quality and runtime efficiency at 6 kbps. Cascaded methods (G-L3AC, G-DAC, M-L3AC, and M-DAC) struggle to balance speed and quality, as their overall performance is heavily limited by the enhancement module. In contrast, end-to-end joint models support real-time processing while maintaining high enhancement quality. UJCodec further improves this trade-off, achieving superior performance with significantly fewer parameters and lower RTF than SDCodec and NRVRVQ.

3.5. Ablation studies

Table 2 presents the impact of key architectural and training components, showing that each contributes notably to UJCodec's performance.

Quantizer: To assess the robustness of $\mathcal{E}_N$ with FSQ or RVQ under mild perturbation, we evaluate 30 noisy utterances with SNRs around 18 dB and compute the proportion of unchanged quantized

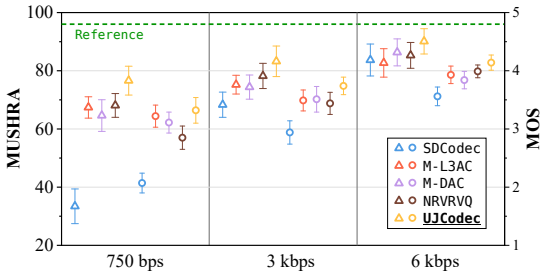


Fig. 5: Subjective scores. △ denote MUSHRA scores (left y-axis), while ◯ denote MOS (right y-axis). Colors indicate different codecs: SDCodec, L3AC, DAC, NRVRVQ and UJCodec.

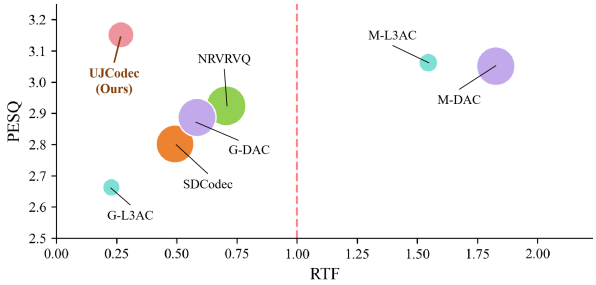


Fig. 6: Comparison of perceptual quality and efficiency **at 6 kbps**. Circle size indicates parameter count. Models on the left of the red line (RTF < 1) support real-time inference.

latent elements. As shown in Fig. 4, 60.54% of the FSQ-based latent elements remain unchanged, notably higher than RVQ's 34.25%, indicating better robustness and alignment stability.

Stage-wise Training: We keep the architecture unchanged and retrain the entire codec on the full dataset described in 3.1 with the alignment loss in Eq. (1). In this setting, the model only reaches baseline-level performance after 350k iterations, about $1.4\times$ that of our proposed approach. This highlights the efficiency of the proposed stage-wise training strategy.

Simulating Latent Frames Corruption: Disabling Latent Frames Corruption causes noticeable word omissions. In contrast, the proposed model performs well without disturbing distortions.

Table 2: Ablation results.

Model	PESQ↑	WER(%)↓
baseline@6kbps	**3.152**	**9.95**
w/o FSQ (replaced by RVQ)	-0.232	+0.27
w/o stage-wise training	-0.126	+0.37
w/o corruption simulation	-0.058	+0.83

4. CONCLUSIONS

We proposed UJCodec, a UNet-style end-to-end speech codec that jointly performs compression and enhancement with low latency. A three-stage training strategy and latent frame corruption simulation were introduced to improve robustness and speech intelligibility. In addition, FSQ was adopted to enhance audio quality and provide a more uniform and explicit latent space, making encoder alignment more stable and effective. Experimental results demonstrate that UJCodec outperforms existing methods in both noise suppression and intelligibility under low-bitrate, noisy conditions, while meeting real-time processing requirements.

5. ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62571037 and U25B2075), the Beijing Natural Science Foundation (Grant Nos. L242089 and L257001), and the National Key Research and Development Program of China (Grant No. 2025YFF0518003).

6. 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] H. Wu and S. Braun, "Ultra-low latency speech enhancement - a comprehensive study," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] L. Yin, W. Tao, D. Zhao, T. Ito, K. Osa, M. Kato, and T.-W. Chen, "Unet−−: Memory-efficient and feature-enhanced network architecture based on u-net with reduced skip-connections," in *Proceedings of the Asian Conference on Computer Vision*, 2024, pp. 4085–4099.

[14] X. Rong, T. Sun, X. Zhang, Y. Hu, C. Zhu, and J. Lu, "Gtcrn: A speech enhancement model requiring ultralow computational resources," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 971–975.

[15] Y.-X. Lu, Y. Ai, and Z.-H. Ling, "Mp-senet: A speech enhancement model with parallel denoising of magnitude and phase spectra," *arXiv preprint arXiv:2305.13686*, 2023.

[16] 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.

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

[18] Y. Shi, H. Bu, X. Xu, S. Zhang, and M. Li, "Aishell-3: A multi-speaker mandarin tts corpus and the baselines," *arXiv preprint arXiv:2010.11567*, 2020.

[19] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating rnn-based speech enhancement methods for noise-robust text-to-speech," in *9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, 2016, pp. 146–152.

[20] 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," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6623–6627.

[21] 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.

[22] ITU-R, *Recommendation BS.1534-1: Method for the subjective assessment of intermediate quality level of coding systems*, International Telecommunications Union, 2001.

[23] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[24] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.