UFC: UNIVERSAL FOCALCODEC FOR ULTRA-LOW BITRATE AUDIO COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural audio codecs (NACs) provide compact discrete representations for audio language models (ALMs). However, many NACs rely on multiple codebooks, increasing token rates and often requiring a two-stage autoregressive and non-autoregressive decoding procedure, and are limited to speech. We propose Universal FocalCodec (UFC), a family of single-codebook NACs for 16 kHz speech, music, and sound. UFC builds on the Qwen3-Omni Audio Transformer (AuT) encoder and quantizes its intermediate representations into a single discrete token stream. We introduce variants operating at frame rates from 13 to 52 Hz, corresponding to bitrates of 0.195 to 0.780 kbps, offering a speed-quality trade-off. Across multiple domains and evaluation metrics, UFC is competitive with state-of-the-art universal codecs. UFC-13 achieves the strongest overall performance among universal codecs operating around 0.195 kbps, leading on most metrics and outperforming several higher-bitrate systems. Our ablations further reveal that early AuT layers provide a favorable balance of semantic and acoustic information for neural audio coding. UFC shows that ALMs can reuse their own audio encoders to generate audio instead of relying on a separate decoding methodology.

1 INTRODUCTION

Neural Audio Codecs (NACs) have become competent compressors that preserve high audio quality at low bit-rates (Zeghidour et al., 2021; Défossez et al., 2023). NACs are typically designed as an autoencoder that maps audio to a compressed quantized latent embedding $\hat{z}$ (Mousavi et al., 2025). The quantized representation $\hat{z}$ can be mapped to a set of discrete tokens from a codec vocabulary V_{codec}. This mapping allows processing audio in a language-modeling fashion with a next-token prediction, making NACs appealing for audio language models (ALMs) (Borsos et al., 2023).

NAC design faces two challenges that limit its efficiency for ALM training and inference: (1) choosing the number of codebooks and their vocabulary size to balance computational efficiency and reconstruction fidelity (Borsos et al., 2023; Tao et al., 2024), and (2) faithfully preserving semantic information for downstream tasks such as audio understanding and text-to-speech (TTS) (Ye et al., 2025a). Multiple codebooks (Zeghidour et al., 2021; Kumar et al., 2023; Hu et al., 2026a) and/or separate acoustic and semantic branches (Hu et al., 2026b; Ye et al., 2025a) can help with faithfully representing audio but at a larger computational burden, *e.g.*, an additional non-autoregressive (NAR) decoding (Borsos et al., 2023; Chen et al., 2025) or multi-branch processing. Recently, FocalCodec (Della Libera et al., 2026) showed that a speech NAC, which is both semantically- and acoustically-aware, can be trained on top of a pretrained self-supervised (SSL) backbone (Chen et al., 2022), without the need for multiple codebooks nor separate acoustic and semantic branches.

In this paper, we extend the FocalCodec concept from speech-only to universal NACs, *i.e.*, operating on music and sound in addition to speech. More specifically, we build a single-codebook universal NAC on top of Qwen3-Omni's Audio Transformer (AuT) encoder (Xu et al., 2025b). The impressive performance of Qwen3-Omni on speech, sound, and music benchmarks, such as MMSU (Wang et al., 2026), MMAU (Sakshi et al., 2025), and MuChoMusic (Weck et al., 2024), suggests that AuT features excel in retaining both acoustic and semantic information.

We refer to our system as *Universal FocalCodec* (UFC), as it is a universal (speech, music, and sound) codec operating at a low 13 Hz frame-rate. Since UFC uses a single implicit codebook and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

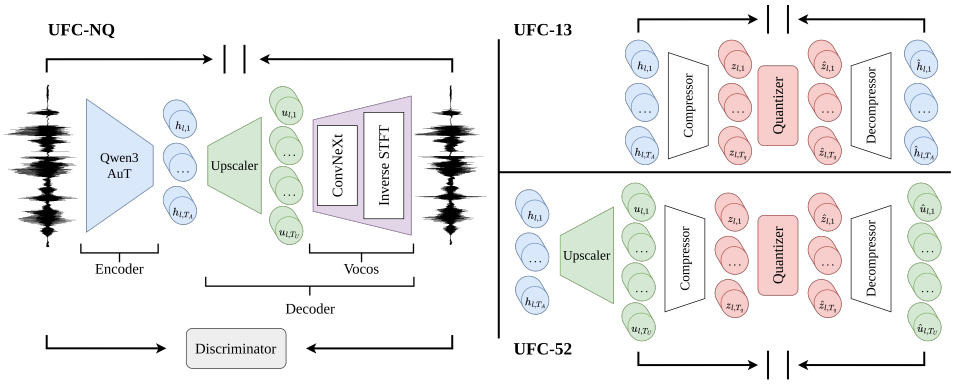


Figure 1: The UFC architecture. (Left) UFC-NQ compresses audio into a continuous latent representation, upsamples it to a higher frame rate, and reconstructs the waveform. (Right, top) UFC quantizes AuT features at 13 Hz. (Right, bottom) An alternative design quantizes the upsampled features, enabling different discrete frame rates through the compressor downsampling factor.

preserves both acoustic and semantic information, it can be used directly with a next-token prediction training/inference style for ALMs, avoiding the need for NAR decoding. Our contributions are as follows:

- We introduce the UFC family, a set of *universal* NACs with a *single codebook* at ultra-low bitrates ($0.195\,\text{kbps}$ to $0.780\,\text{kbps}$). UFC-13 operates at a $2.5\times$ smaller bitrate than the lowest previously reported bitrate of $0.480\,\text{kbps}$ (Ji et al., 2025), while even surpassing some of the higher-bitrate codecs operating at $0.2\,\text{kbps}$ to $1.0\,\text{kbps}$. UFC demonstrates that an ALM's audio generation can be done by building on top of ALM's own audio encoder rather than requiring a separate decoding architecture.
- We perform extensive ablations on the choice of AuT features and demonstrate their semantical and acoustical awareness, thus making them a simpler and cheaper alternative to dual-branch codecs.
- We compare five quantization schemes: Finite Scalar Quantization (FSQ), Binary Spherical Quantization (BSQ), vanilla VQ, SimVQ, and SharedVQ, a domain-aware variant inspired by split and nested codebook designs from prior universal NACs. While FocalCodec favors BSQ and ablates only implicit-codebook quantizers, we show that BSQ, FSQ, and VQ achieve similar overall performance in UFC. We further find that UFC is less susceptible to the domain-disentanglement problem of universal NACs at higher frame rates.
- UFC is a *fully-open* release[1], *i.e.*, open-data, open-weights, open-license, and open training/inference scripts.

2 Related Work

2.1 Addressing Semantic Shortcomings of NACs

Given an input audio signal x, a NAC aims to reconstruct it such that $\hat{x} \simeq x$. Training typically combines reconstruction objectives, such as L_1, L_2, Mel/STFT losses, with adversarial losses that improve perceptual realism (Zeghidour et al., 2021). However, these objectives primarily target acoustic fidelity and do not explicitly encourage preservation of semantic information (namely phonetic and lexical content). As a result, NAC representations can be suboptimal for audio understanding in ALMs and for generation tasks requiring rich phonetic information, such as TTS. Conversely, semantic-only tokenizers based on clusters from pretrained SSL models suffer from poor acoustic reconstruction (Borsos et al., 2023).

To tackle this issue, a dual-branch system consisting of an SSL backbone and a NAC can be created by either (1) concatenating SSL units and acoustic tokens with separate decoders (Borsos et al.,

[1]The link will be provided upon publication, it is hidden for the double-blind review.

2023) or with a joint one (Li et al., 2025; Hu et al., 2026b) or (2) distilling information from an SSL backbone into the first out of multiple acoustic tokens (Zhang et al., 2024; Chen et al., 2026). Nonetheless, these approaches result in multiple tokens per frame, which inevitably leads to the inefficient dual-stage decoding procedure (see Section 2.2). Alternatively, XCodec-2 (Ye et al., 2025b) proposed concatenating semantic and acoustic features in the continuous space and quantizing them together with a single codebook. While achieving high performance, XCodec-2 preserves two branches in the encoding, making the joint NAC system computationally demanding. In contrast, FocalCodec (Della Libera et al., 2026) uses the first layers of an SSL backbone, which are both semantically and acoustically rich, and builds a NAC directly on top of a pretrained SSL system, thus avoiding the dual-branch design. We extend this idea from speech-only NACs to universal models capable of reconstructing music and sound as well.

Semantic information can also be introduced without a separate SSL branch by adding an auxiliary language-modeling loss over acoustic tokens (Gong et al., 2026; Zhang et al., 2025). However, this approach is computationally more demanding and less direct than the FocalCodec design.

2.2 NACs and Language Modeling

As discussed in Section 2.1, preserving both semantic and acoustic knowledge often requires separate semantic and acoustic tokens. Moreover, to avoid excessively large codebooks and difficult optimization, acoustic information is typically represented using multiple codebooks (Hu et al., 2026b; Li et al., 2025; Gong et al., 2026; Boson AI, 2025; Kumar et al., 2023; Zeghidour et al., 2021). Consequently, audio generation with such NACs usually requires a two-stage procedure (Borsos et al., 2023; Chen et al., 2025): an AR model first generates semantic tokens or the first acoustic codebook, followed by a NAR model that predicts the remaining codebooks in parallel. Although more efficient than AR decoding, this additional stage introduces computational and optimization overhead and may require diffusion (Liu et al., 2024; Yang et al., 2025) or flow-matching Transformer/DiT models (Liu et al., 2026; Xu et al., 2025a).

Multiple-codebook NACs are also inefficient for audio understanding, *i.e.*, $(\text{audio}, \text{text}) \rightarrow \text{text}$. Using only the first codebook limits auditory perception due to the acoustic shortcomings of SSL representations (Borsos et al., 2023), while using all codebooks substantially increases sequence length. In particular, even at a low encoder frame rate (FR), the effective number of tokens per second (TPS) processed by the ALM scales as $\mathrm{TPS} = \mathrm{FR} \times N_{\mathrm{codebooks}}$. Thus, multiple-codebook NACs introduce substantial inefficiencies in both audio generation and understanding, motivating a single-codebook representation compatible with conventional next-token prediction.

3 Methodology

Our UFC NAC builds on the FocalCodec methodology (Della Libera et al., 2026). A conventional NAC quantizes the encoder latent space z and reconstructs the waveform directly from the quantized representation $\hat{z}$. FocalCodec, on the other hand, supposes that the decoder can synthesize audio from intermediate encoder features h. As a result, quantization should preserve the information needed to reconstruct h rather than the waveform itself. Following this idea, we first construct UFC-NQ, a non-quantized model that reconstructs audio directly from h. UFC then inserts a compressor–decompressor module into UFC-NQ to quantize these hidden features and represent the input audio as a sequence of discrete tokens. The full pipeline is shown in Figure 1.

3.1 UFC-NQ (non-quantized)

UFC-NQ follows an autoencoder design. Given an audio waveform $x \in \mathbb{R}^T$, we extract hidden features $h_l \in \mathbb{R}^{T_A \times D_h}$ from the l-th intermediate layer of the Qwen3-Omni AuT encoder (Xu et al., 2025b) at 13 Hz frame-rate[2]. Prior work has shown that intermediate layers of pretrained audio models capture different mixtures of semantic and acoustic information (Pasad et al., 2021; 2023). Motivated by the strong audio-understanding performance of Qwen3-Omni (Wang et al., 2026; Sakshi et al., 2025; Weck et al., 2024), we hypothesize that some AuT layers provide representations

[2]The theoretical frame-rate is 12.5 Hz, however, the original Qwen's implementation is effectively 13 Hz.

suitable for both semantic modeling and waveform reconstruction. We evaluate all 32 Transformer layers in Section 5.3.

The decoder reconstructs the waveform as $\hat{x} = D(h_l) = V(U(h_l))$, where $U(\cdot)$ is an upscaler and $V(\cdot)$ is a Vocos decoder (Siuzdak, 2024). Directly decoding the 13 Hz AuT features would require a very large STFT hop size and therefore poor temporal resolution. We instead upsample the representations to 52 Hz before Vocos decoding. $U(\cdot)$ is the FocalDecoder from FocalCodec (Della Libera et al., 2026). We study the impact of this component in Appendix B

3.2 UFC (QUANTIZED)

To obtain a discrete NAC, UFC inserts compressor–quantizer–decompressor modules into the pretrained UFC-NQ model. Given encoder features $h_l \in \mathbb{R}^{T_A \times D_h}$, the compressor maps them to a quantization space, $z_l = E_{\text{comp}}(h_l)$, which is discretized as $\hat{z}_l = Q(z_l)$. The decompressor then reconstructs the encoder representation, $\hat{h}_l = D_{\text{comp}}(\hat{z}_l) \simeq h_l$.

The quantized representation $\hat{z}_l$ can equivalently be viewed as a sequence of discrete codec tokens. We use the FocalEncoder and FocalDecoder from FocalCodec (Della Libera et al., 2026) as the compressor and decompressor. Although AuT already operates at a low frame rate, the compressor is needed to project features in a space that is optimized for quantization.

Our default variant, *UFC-13*, quantizes AuT features directly at 13 Hz without additional temporal downsampling. Since this module reconstructs features from the frozen pretrained AuT encoder and does not depend on the UFC-NQ decoder, it can be trained independently of UFC-NQ. Alternatively, quantization can be applied after the upscaler. Let $u_l = U(h_l) \in \mathbb{R}^{T_u \times D_u}$ denote the 52 Hz upscaled representation. In this case, the codec reconstructs $\hat{u}_l = D_{\text{comp}}\left(Q\left(E_{\text{comp}}(u_l)\right)\right) \simeq u_l$. By varying the temporal compression ratio of the compressor, we obtain three variants: *UFC-52*, which quantizes at 52 Hz; *UFC-52-26*, which reduces the discrete frame rate to 26 Hz; and *UFC-52-13*, which quantizes the upscaled representation at 13 Hz. The latter enables a direct comparison between quantizing AuT features and upscaled features at the same token rate. Unlike UFC-13, these variants depend on the learned upscaler and therefore cannot be trained independently of UFC-NQ.

3.2.1 QUANTIZER

While FocalCodec uses BSQ (Zhao et al., 2025) and ablates only implicit codebook designs, we compare five quantization schemes to use within UFC. BSQ is known for its strong performance with large codebooks and low frame rates (Zhao et al., 2025; Della Libera et al., 2026). BSQ maps each latent vector to a point on a hypersphere, based on sign, corresponding to an implicit codebook of size 2^K. We use $K = 15$, yielding $|V_{\text{codec}}| = 32768$. We refer to this model as *UFC-BSQ*.

FSQ (Mentzer et al., 2024; Ye et al., 2025b) independently quantizes latent dimensions into a finite number of levels and defines an implicit codebook of size $\prod_i K_i$. We use five dimensions with eight levels each, resulting in the same codebook size of 32768. These variants are denoted *UFC-FSQ*.

Both BSQ and FSQ avoid maintaining an explicit codebook and are therefore less susceptible to codebook collapse at large vocabulary sizes (Mentzer et al., 2024; Zhao et al., 2025). For comparison with VQ-based universal NACs (Ji et al., 2025; Chen et al., 2026; Jiang et al., 2025), we additionally evaluate vanilla VQ with k-means initialization and EMA updates Zeghidour et al. (2021), denoted as *UFC-VQ*, under the same vocabulary size. SimVQ's (Zhu et al., 2025b) evaluation is in Appendix C, as it is less effective at low frame rates. Furthermore, SharedVQ ablation is in Appendix D.

4 EXPERIMENTAL SETUP

4.1 CORPUS DETAILS

We construct a universal training corpus covering speech, music, and general sound, summarized in Table 1. All audio is resampled to 16 kHz, and we train on random 6-second crops. Shorter samples are padded by replication. To improve domain balance and expose the model to different segments of longer recordings, we oversample MTG (Bogdanov et al., 2019), Suno[3], and AudioSet (Gemmeke

[3] https://huggingface.co/datasets/nyuuzyou/suno

Table 1: The number of elements and audio hours inside our corpus in terms of the amount of original data and the amount corresponding to random 6-seconds-long crops extracted for training.

Type	Dataset	# Elements (Originally)	# Crops	# Hours (Originally)	# Hours (Crops)
Speech	LibriTTS	354780	354828	555.16	591.39
	YODAS-Granary	78226147	78235441	192027.08	130392.40
	total	78580927	78590269	192582.24	130983.79
Music	MTG	49978	1999388	3394.31	3332.31
	Suno	355044	10652613	17634.01	17754.35
	total	405022	12652001	21028.32	21086.66
Sound	AudioSet	1738657	3477730	4787.93	5796.22
All	total	80724606	94720000	218398.49	157866.67

et al., 2017) by factors of 30, 40, and 2, respectively. This results in approximately 94.7M training crops, matching the number of samples processed during codec training.

Our speech data includes LibriTTS (Zen et al., 2019) and YODAS-Granary (Rao Koluguri et al., 2025), covering 23 European languages. Combined with the multilingual Qwen3-Omni AuT backbone (Xu et al., 2025b), pretrained on more than 20M hours and 19 languages, this makes UFC inherently multilingual.

4.2 Training Setup

We separate training of our model into two stages. First, we train UFC-NQ to learn audio reconstruction from the hidden features h_l (we choose $l = 4$, as discussed in Section 5.3). Its weights are updated for 740K steps on $4\times$NVIDIA GH200 with an effective batch size of 128 in BF16 via AdamW. We follow the DAC loss function setup (Kumar et al., 2023): multi-scale MelSpectrogram reconstuction loss, L_1 feature-matching loss, and the Hinge formulation of adversarial loss (Lim & Ye, 2017). Parker et al. (2025) suggested to fine-tune the model with a feature-matching loss from a pretrained SSL backbone to improve perceptual quality. We build on this idea and propose to use two backbones: AuT itself to supervise matching of semantic features and MOSS NAC (Gong et al., 2026) to propagate information about acoustic features. Furthermore, we find it beneficial to add these losses from the beginning to improve convergence. The final perceptual loss is defined as follows:

$$\mathcal{L}_{\text{Perceptual}} = \lambda_{\text{semantic}} \sum_{C_n \in \text{AuT}} \frac{\|C_n(x) - C_n(\hat{x})\|_1}{N_{\text{AuT}} \|C_n(x)\|_1} + \lambda_{\text{acoustic}} \sum_{C_n \in \text{MOSS}} \frac{\|C_n(x) - C_n(\hat{x})\|_1}{N_{\text{MOSS}} \|C_n(x)\|_1}, \quad (1)$$

where C_n is the intermediate layer and N is the total number of layers.

UFC then adds a randomly initialized compressor-decompressor to a pretrained UFC-NQ. $E_{\text{comp}}(\cdot), D_{\text{comp}}(\cdot), Q(\cdot)$ are updated for $740K$ additional steps, while other weights are kept frozen. The model is trained with a mixture of MSE reconstruction loss and a quantization loss. For BSQ, the $Q(\cdot)$ is optimized via entropy following the look-up free quantization design (Yu et al., 2024; Zhao et al., 2025) and, for VQ-based methods, using the commitment loss Zeghidour et al. (2021). FSQ does not need any loss.

4.3 Metrics

For speech, we evaluate perceptual quality with UTMOS (Saeki et al., 2022), SCOREQ (Ragano et al., 2024), PESQ (Rix et al., 2001), and ViSQOL (Chinen et al., 2020); intelligibility with STOI (Taal et al., 2010) and differential WER (dWER) (Wang et al., 2021) using Whisper (Radford et al., 2023); and speaker preservation using WavLM-based speaker similarity (SPK) (Chen et al., 2022). For music and environmental sounds, we report Mel/STFT reconstruction distances, ViSQOL, the audio Aesthetics metrics (Tjandra et al., 2025), *i.e.*, Content Enjoyment (CE), Content Usefulness (CU), Production Quality (PQ), and Production Complexity (PC); and embedding similarity (SIM) using domain-specific pretrained audio encoders (Bolya et al., 2026; Zhu et al., 2025a). No human evaluation is conducted as UTMOS and SCOREQ already highly correlate with listening tests (Mack et al., 2026). We additionally report the codebook size, tokens per second (TPS), and bitrate (BR) to characterize codec efficiency. Given frame rate FR, tokens per frame TPF, and codebook size $|V_{\text{codec}}|$, $\text{TPS} = \text{FR} \cdot \text{TPF}$, $\text{BR} = \text{TPS} \log_2 |V_{\text{codec}}|$.

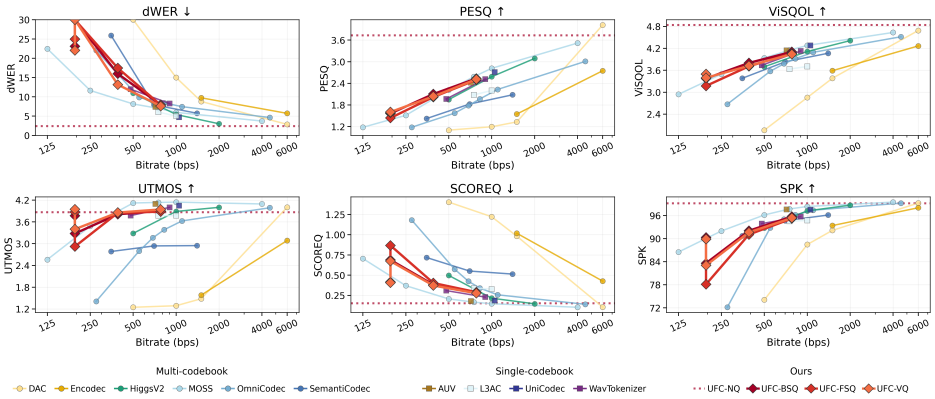


Figure 2: Metrics versus bitrate for different universal NACs and UFC on the test-clean split of LibriSpeech (Panayotov et al., 2015). At 195 bps, the two UFC points correspond to UFC-13 and UFC-52-13, with UFC-13 being the better-performing point.

5 Results

We compare UFC with state-of-the-art single-codebook NACs, including WavTokenizer (Ji et al., 2025), UniCodec (Jiang et al., 2025), AUV (Chen et al., 2026), and L3AC (Zhai et al., 2025), as well as multi-codebook systems such as DAC (Kumar et al., 2023), EnCodec (Défossez et al., 2023), SemantiCodec (Liu et al., 2024), OmniCodec (Hu et al., 2026b), MOSS (Gong et al., 2026), and HiggsV2 (Boson AI, 2025). For RVQ-based codecs, we additionally evaluate lower-bitrate configurations by decoding from the first $n \leq N_q$ codebooks. StableCodec (Parker et al., 2025) is evaluated using the authors' post-hoc single-codebook FSQ variant. Models operating at 24 kHz are evaluated by resampling inputs and outputs accordingly. Figures are used to visualize the main trends (metric vs. bitrate), while exact metric values are reported in Appendix F.

5.1 Speech Data

To evaluate speech reconstruction, we compare UFC with universal NAC baselines on LibriSpeech (Panayotov et al., 2015). Figure 2 visualizes performance as a function of bitrate.

Figure 2 (for exact metric values see Table 8 in Appendix F) highlights the efficiency of single-codebook representations. Multi-codebook NACs degrade significantly as fewer codebooks are used at inference time, and most are less competitive than single-codebook systems at comparable bitrates. MOSS (Gong et al., 2026) is a notable exception and remains competitive at reduced bitrate. One possible contributing factor is its substantially larger model capacity, approximately $5\times$ that of the single-codebook systems considered here (Appendix F).

Across BSQ, FSQ, and VQ, UFC exhibits similar overall performance, suggesting that the proposed architecture is relatively robust to the choice of quantization method. The largest differences appear in UTMOS, where the VQ-based variant performs better. Since FSQ does not require an auxiliary quantization loss and VQ is more difficult to stabilize during training, FSQ provides a favorable practical trade-off. Given the similar overall performance we omit the quantizer suffix below.

Increasing the discrete frame rate from 13 Hz to 52 Hz, *i.e.*, going from 195 bps to 780 bps, consistently improves reconstruction quality, while UFC-52-26 provides an intermediate trade-off between bitrate and fidelity. We originally hypothesized that the upscaled features u_l would also be easier to quantize than the raw AuT latents h_l, since AuT representations are not explicitly optimized for quantization. However, UFC-52-13 performs worse than UFC-13 despite using the same 13 Hz token rate. This indicates that the gains of UFC-52 primarily come from the finer temporal resolution, while compressing the upscaled features back to 13 Hz removes this advantage and does not improve quantizability over the original AuT representation.

Table 2 shows the metrics for the non-quantized UFC-NQ backbone. It achieves superior performance in terms of intelligibility metrics and is among the best systems for other metrics, establish-

Table 2: UFC-NQ performance depending on the perceptual loss coefficients.

$\lambda_{\text{acoustic}}$	$\lambda_{\text{semantic}}$	dWER ↓	STOI ↑	PESQ ↑	ViSQOL ↑	UTMOS ↑	SCOREQ ↓	SPK ↑
10.0	10.0	**2.41**	**0.97**	**3.73**	**4.84**	**3.87**	**0.15**	**99.2**
10.0	0.0	2.86	0.96	3.62	4.79	3.85	0.16	**99.2**
0.0	0.0	2.69	0.96	3.62	4.81	3.75	0.17	99.1

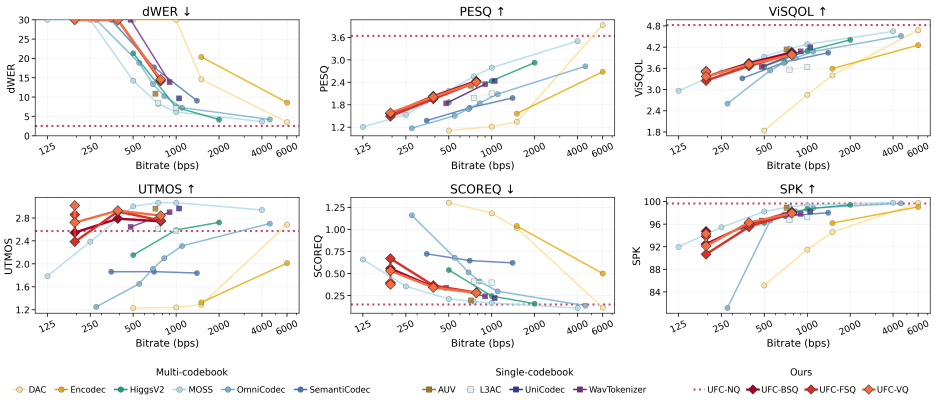


Figure 3: Metrics versus bitrate for different universal NACs and UFC on the non-English partition of Multilingual LibriSpeech (Pratap et al., 2020). At 195 bps, the two UFC points correspond to UFC-13 and UFC-52-13, with UFC-13 being the better-performing point.

ing a strong upper bound before quantization. The acoustic perceptual loss improves reconstruction quality, while adding the semantic perceptual loss improves both quality and intelligibility.

Overall, UFC achieves competitive or superior performance across a wide bitrate range. In particular, UFC-13 remains highly competitive at its ultra-low bitrate of 195 bps and outperforms or matches several higher-bitrate baselines on most metrics, including OmniCodec at 275-550 bps. At higher bitrates, AUV (Chen et al., 2026) and UniCodec (Jiang et al., 2025) remain stronger on some perceptual metrics, while AUV also achieves slightly higher speaker similarity.

We further evaluate multilingual speech on the non-English test partition of Multilingual LibriSpeech (Pratap et al., 2020) (Figure 3). The same overall trends hold: increasing bitrate improves reconstruction quality, while UFC-13 remains competitive with systems operating at substantially higher rates. dWER degrades more strongly in the multilingual setting. However, UFC-NQ largely preserves its dWER, indicating that most of this intelligibility degradation is introduced by the compressor–quantizer–decompressor stage rather than by the AuT backbone or waveform decoder.

5.2 Sound and Music Data

Our codec also supports music and environmental sounds. On SongDescriber (Manco et al., 2023) (Figure 4, Table 10 in Appendix F), UFC-NQ remains competitive across most metrics, although its Mel and STFT distances are weaker. These time-frequency distances are sensitive to small local spectral mismatches and serve only as proxies for perceptual quality, so worse values do not necessarily imply worse subjective reconstruction. Rather, they indicate that some fine-grained spectral details are not reproduced exactly. As this limitation is already present in the non-quantized backbone, it is subsequently inherited by the quantized variants. Nonetheless, UFC performs strongly on perceptual quality and semantic similarity metrics. In particular, UFC-13 surpasses AUV (Chen et al., 2026) in PQ and matches it in CE, while outperforming several multi-codebook NACs operating at substantially higher bitrates.

For general sound, we evaluate on the AudioSet evaluation split (Gemmeke et al., 2017) (Figure 5, Table 11 in Appendix F). UFC remains competitive at comparable bitrates on reconstruction and perceptual-quality metrics, but performs less strongly on semantic similarity. This suggests that the model preserves acoustic structure better than high-level semantic content for this domain. One possible factor is the training distribution, which contains substantially more speech and music than

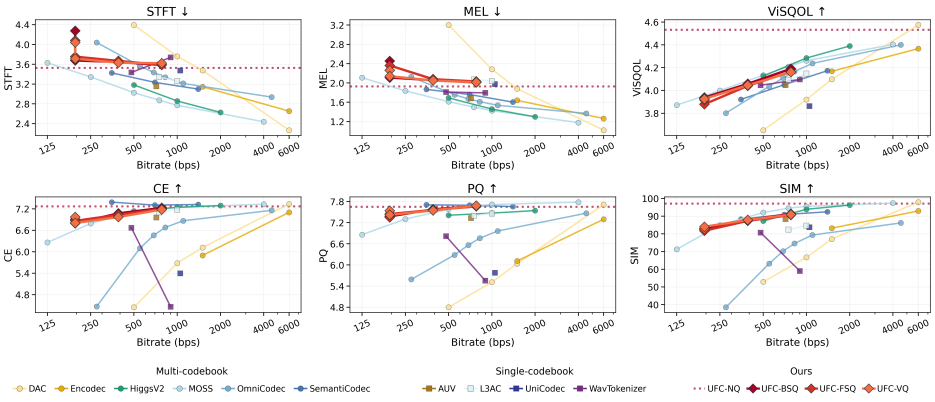


Figure 4: Metrics versus bitrate for different universal NACs and UFC on SongDescriber (Manco et al., 2023). At 195 bps, the two UFC points correspond to UFC-13 and UFC-52-13, with UFC-13 being the better-performing point.

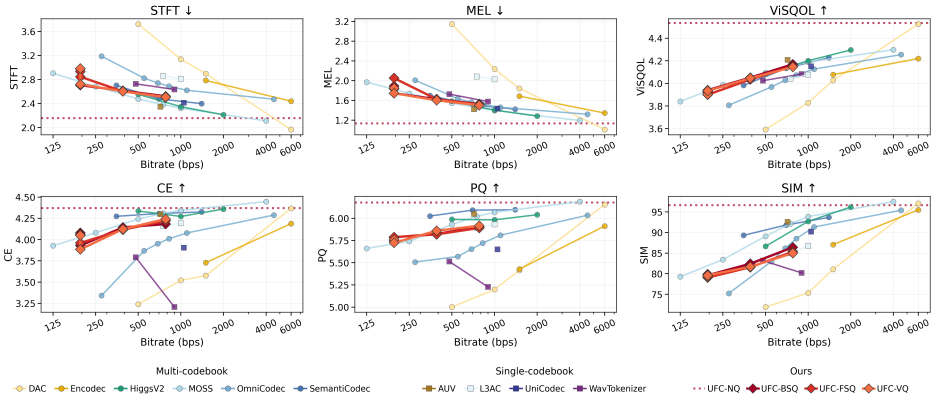


Figure 5: Metrics versus bitrate for different universal NACs and UFC on the evaluation split of AudioSet (Gemmeke et al., 2017). At 195 bps, the two UFC points correspond to UFC-13 and UFC-52-13, with UFC-13 being the better-performing point.

sound data (Section 4.1). Increasing the diversity and amount of sound data may therefore improve semantic preservation. Nevertheless, UFC-13 matches or exceeds several NACs operating at 200–1100 bps on the Aesthetics metrics (Tjandra et al., 2025), including OmniCodec (Hu et al., 2026b).

UFC-NQ again provides a useful upper bound: on AudioSet it achieves very low Mel and STFT distances, while on music these metrics remain comparatively weaker even before quantization. This suggests that the behavior of these proxy reconstruction metrics is driven substantially by the underlying continuous representation and decoder, rather than by quantization alone.

5.3 Intermediate Layer Comparison

Previous studies on SSL backbones report that lower layers preserve more acoustic information, while deeper representations become increasingly semantic (Pasad et al., 2021; 2023). We investigate whether this behavior also holds for the Qwen3-Omni AuT (Xu et al., 2025b) encoder, which is optimized as part of an encoder–decoder system for audio understanding. Despite this semantic training objective, Qwen3-Omni also performs strongly on perceptual tasks, suggesting that AuT retains substantial acoustic information.

We train UFC-NQ using representations from each of the 32 AuT Transformer layers for 90K steps on a single GPU on the train-clean-100 partition of LibriSpeech. As shown in Figure 6, all layers retain enough information for waveform reconstruction, but the first few layers perform best on nearly all metrics. Interestingly, STOI, dWER, and speaker similarity also degrade with layer

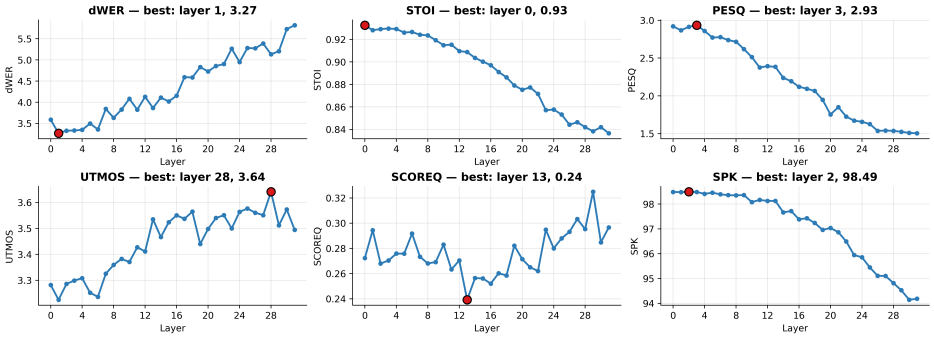


Figure 6: Comparison of UFC-NQ performance trained on different intermediate layers. Results are shown on the dev-clean split of LibriSpeech (Panayotov et al., 2015).

depth, contrary to the commonly observed trend in SSL encoders. One possible explanation is that these metrics are measured on reconstructed waveforms and therefore remain coupled to reconstruction quality: degradation of acoustic detail can also reduce intelligibility and speaker similarity. In addition, deeper AuT representations may become increasingly specialized for the subsequent Qwen3 decoding process rather than for preserving information required for waveform reconstruction. However, the absence of gains from multi-layer fusion (Appendix A) further suggests that later representations do not provide complementary information useful for reconstruction.

UTMOS behaves differently and improves for deeper layers, while SCOREQ peaks around the middle of the network. Since the remaining quality, intelligibility, and speaker metrics consistently favor early layers, we attribute the UTMOS trend at least partly to known biases of the metric (Huang & Toda, 2026). SCOREQ, being a neural-based objective, may also be subject to biases. Overall, layer 4 provides the strongest trade-off and is used in all UFC models.

6 Conclusion

In this work, we introduced UFC, a family of universal single-codebook NACs operating at $13\,\text{Hz}$ to $52\,\text{Hz}$ and $0.195\,\text{kbps}$ to $0.780\,\text{kbps}$. Its low token rate and single-codebook representation enable conventional next-token prediction in ALMs without an additional NAR decoding stage. Across speech, music, and sound, UFC achieves competitive performance with state-of-the-art universal codecs. In particular, UFC-13 operates at only $195\,\text{bps}$, achieves the strongest performance on most metrics among universal codecs around this bitrate, and matches or surpasses several higher-bitrate systems across multiple metrics and domains. Our results demonstrate that representations from a pretrained ALM encoder can serve as the basis of a universal neural codec, enabling generative ALMs to potentially reuse their audio encoder for generation.

In our quantizer comparison, BSQ, FSQ, and VQ achieve similar overall performance. Given the greater optimization complexity of VQ, implicit quantizers such as BSQ and FSQ provide a better practical choice. Speech-only training (Appendix C) further reveals a domain-disentanglement challenge when allocating a global discrete vocabulary across heterogeneous audio domains. Importantly, this effect is concentrated at the lower $13\,\text{Hz}$ and $26\,\text{Hz}$ frame rates, whereas UFC-52 performs similarly under universal and speech-only training. This differs from prior universal NACs (Ji et al., 2025; Jiang et al., 2025; Chen et al., 2026) operating at comparable frame rates, which introduced split or nested codebook mechanisms to reduce the gap between domain-specific and universal training. Thus, UFC can maintain a global single-codebook representation across domains at $52\,\text{Hz}$ without explicit domain-aware codebook partitioning, while improving domain disentanglement at lower frame rates remains a direction for future work (see Appendix C and D).

Finally, our layer analysis shows that early Qwen3-Omni AuT representations retain both semantic and acoustic information. A remaining limitation is that AuT operates on Mel spectrograms and does not explicitly preserve waveform phase, leaving its recovery to the decoder. Future work could incorporate a parallel waveform-based acoustic representation, following XCodec-2 (Ye et al., 2025b) (see Appendix E for preliminary results), at the cost of single-branch efficiency.

AI USE STATEMENT

In this work, generative AI tools have been used to assist in the refactoring of our manuscript to fit the page limit and improve readability. Additionally, we have used coding agents to write scripts for the conversion of metrics tables into figures and parse different dataset formats. Generative AI tools *have not* been used for any other tasks, including: generation of any new datasets, design or development of the methodology, proposal and tests of hypotheses, formulation and proving of any claims and mathematical equations, assistance with translation, or results interpretation. The whole final content of this work and code is solely our responsibility. We have reviewed all AI-aided work by proof-reading and modifying refactored text and code.

REPRODUCIBILITY STATEMENT

We fully open-source UFC to support reproducibility and future research[4]. We release all training and evaluation code, exact model and loss configurations, complete hyperparameters, a Docker environment for reproducing our setup, and checkpoints for all UFC-family models.

REFERENCES

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. In *Machine learning for music discovery workshop, international conference on machine learning (ICML 2019)*, pp. 1–3. Long Beach, CA, United States, 2019.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, et al. Perception encoder: The best visual embeddings are not at the output of the network. *Advances in Neural Information Processing Systems*, 38:60884–60937, 2026.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533, 2023.

Boson AI. Higgs Audio V2: Redefining Expressiveness in Audio Generation. `https://github.com/boson-ai/higgs-audio`, 2025. GitHub repository. Release blog available at `https://www.boson.ai/blog/higgs-audio-v2`.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, 33:705–718, 2025.

Yushen Chen, Kai Hu, Long Zhou, Shulin Feng, Xusheng Yang, Hangting Chen, and Xie Chen. Auv: Teaching audio universal vector quantization with single nested codebook. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 17027–17031. IEEE, 2026.

Michael Chinen, Felicia SC Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines. Visqol v3: An open source production ready objective speech and audio metric. In *2020 twelfth international conference on quality of multimedia experience (QoMEX)*, pp. 1–6. IEEE, 2020.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[4]Hidden URL due to double-blind review.

Luca Della Libera, Francesco Paissan, Cem Subakan, and Mirco Ravanelli. Focalcodec: Low-bitrate speech coding via focal modulation networks. *Advances in Neural Information Processing Systems*, 38:23742–23767, 2026.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Yitian Gong, Kuangwei Chen, Zhaoye Fei, Xiaogui Yang, Ke Chen, Yang Wang, Kexin Huang, Mingshu Chen, Ruixiao Li, Qingyuan Cheng, et al. Moss-audio-tokenizer: Scaling audio tokenizers for future audio foundation models. *arXiv preprint arXiv:2602.10934*, 2026.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*, 2026a.

Jingbin Hu, Haoyu Zhang, Dake Guo, Qirui Zhan, Wenhao Li, Huakang Chen, Guobin Ma, Hanke Xie, Chengyou Wang, Pengyuan Xie, et al. Omnicodec: Low frame rate universal audio codec with semantic-acoustic disentanglement. *arXiv preprint arXiv:2603.20638*, 2026b.

Wen-Chin Huang and Tomoki Toda. Attacking utmos: Probing the robustness of a speech quality assessment model. *arXiv preprint arXiv:2606.31105*, 2026.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Ruiqi Li, Ziang Zhang, et al. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 93809–93826, 2025.

Yidi Jiang, Qian Chen, Shengpeng Ji, Yu Xi, Wen Wang, Chong Zhang, Xianghu Yue, ShiLiang Zhang, and Haizhou Li. Unicodec: Unified audio codec with single domain-adaptive codebook. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19112–19124, 2025.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36:27980–27993, 2023.

Jiaqi Li, Xiaolong Lin, Zhekai Li, Shixi Huang, Yuancheng Wang, Chaoren Wang, Zhenpeng Zhan, and Zhizheng Wu. DualCodec: A Low-Frame-Rate, Semantically-Enhanced Neural Audio Codec for Speech Generation. In *Interspeech 2025*, pp. 4883–4887, 2025. doi: 10.21437/Interspeech.2025-468.

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Alexander H Liu, Alexis Tacnet, Andy Ehrenberg, Andy Lo, Chen-Yo Sun, Guillaume Lample, Henry Lagarde, Jean-Malo Delignon, Jaeyoung Kim, John Harvill, et al. Voxtral tts. *arXiv preprint arXiv:2603.25551*, 2026.

Haohe Liu, Xuenan Xu, Yi Yuan, Mengyue Wu, Wenwu Wang, and Mark D Plumbley. Semanticodec: An ultra low bitrate semantic audio codec for general sound. *IEEE Journal of Selected Topics in Signal Processing*, 18(8):1448–1461, 2024.

Wolfgang Mack, Nezih Topaloglu, Laura Lechler, Ivana Balić, Alexandra Craciun, Mansur Yesilbursa, and Kamil Wojcicki. Assessing speech quality metrics for evaluation of neural audio codecs under clean speech conditions. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 22457–22461. IEEE, 2026.

Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, Elio Quinton, György Fazekas, and Juhan Nam. The song describer dataset: a corpus of audio captions for music-and-language evaluation. In *Machine Learning for Audio Workshop at NeurIPS 2023*, 2023.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *International Conference on Learning Representations*, volume 2024, pp. 51772–51783, 2024.

Pooneh Mousavi, Gallil Maimon, Adel Moumen, Darius Petermann, Jiatong Shi, Haibin Wu, Haici Yang, Anastasia Kuznetsova, Artem Ploujnikov, Ricard Marxer, Bhuvana Ramabhadran, Benjamin Elizalde, Loren Lugosch, Jinyu Li, Cem Subakan, Phil Woodland, Minje Kim, Hung yi Lee, Shinji Watanabe, Yossi Adi, and Mirco Ravanelli. Discrete audio tokens: More than a survey! *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Julian Parker, Anton Smirnov, Jordi Pons, CJ Carr, Zack Zukowski, Zach Evans, and Xubo Liu. Scaling transformers for low-bitrate high-quality speech coding. In *International Conference on Learning Representations*, volume 2025, pp. 51997–52021, 2025.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. Layer-wise analysis of a self-supervised speech representation model. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 914–921. IEEE, 2021.

Ankita Pasad, Bowen Shi, and Karen Livescu. Comparative layer-wise analysis of self-supervised speech models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. MLS: A Large-Scale Multilingual Dataset for Speech Research. In *Interspeech 2020*, pp. 2757–2761, 2020. doi: 10.21437/Interspeech.2020-2826.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

Alessandro Ragano, Jan Skoglund, and Andrew Hines. Scoreq: Speech quality assessment with contrastive regression. *Advances in Neural Information Processing Systems*, 37:105702–105729, 2024.

Nithin Rao Koluguri, Monica Sekoyan, George Zelenfroynd, Sasha Meister, Shuoyang Ding, Sofia Kostandian, He Huang, Nikolay Karpov, Jagadeesh Balam, Vitaly Lavrukhin, Yifan Peng, Sara Papi, Marco Gaido, Alessio Brutti, and Boris Ginsburg. Granary: Speech Recognition and Translation Dataset in 25 European Languages. In *Interspeech 2025*, pp. 3923–3927, 2025. doi: 10.21437/Interspeech.2025-540.

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pp. 749–752. IEEE, 2001.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022. In *Interspeech 2022*, pp. 4521–4525, 2022. doi: 10.21437/Interspeech.2022-439.

Sakshi Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. Mmau: A massive multi-task audio understanding and reasoning benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 84929–84964, 2025.

Hubert Siuzdak. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pp. 4214–4217. IEEE, 2010.

Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies. *Advances in Neural Information Processing Systems*, 37:114147–114179, 2024.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, et al. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. *arXiv preprint arXiv:2502.05139*, 2025.

Dingdong Wang, Junan Li, Jincenzi Wu, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. Mmsu: A massive multi-task spoken language understanding and reasoning benchmark. In *International Conference on Learning Representations*, volume 2026, pp. 31374–31410, 2026.

Zhong-Qiu Wang, Hakan Erdogan, Scott Wisdom, Kevin Wilson, Desh Raj, Shinji Watanabe, Zhuo Chen, and John R Hershey. Sequential multi-frame neural beamforming for speech separation and enhancement. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 905–911. IEEE, 2021.

Benno Weck, Ilaria Manco, Emmanouil Benetos, Elio Quinton, György Fazekas, and Dmitry Bogdanov. Muchomusic: Evaluating music understanding in multimodal audio-language models. In *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

Detai Xin, Xu Tan, Shinnosuke Takamichi, and Hiroshi Saruwatari. Bigcodec: Pushing the limits of low-bitrate neural speech codec. *arXiv preprint arXiv:2409.05377*, 2024.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025a.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025b.

Yang Yang, Yunpeng Li, George Sung, Shao-Fu Shih, Craig Dooley, Alessio Centazzo, and Ramanan Rajeswaran. Diffsoundstream: Efficient speech tokenization via diffusion decoding. *arXiv preprint arXiv:2506.22362*, 2025.

Zhen Ye, Peiwen Sun, Jiahe Lei, Hongzhan Lin, Xu Tan, Zheqi Dai, Qiuqiang Kong, Jianyi Chen, Jiahao Pan, Qifeng Liu, et al. Codec does matter: Exploring the semantic shortcoming of codec for audio language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25697–25705, 2025a.

Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, et al. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*, 2025b.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *International Conference on Learning Representations*, volume 2024, pp. 765–783, 2024.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

Linwei Zhai, Han Ding, Cui Zhao, Ge Wang, Wang Zhi, Wei Xi, et al. L3ac: Towards a lightweight and lossless audio codec. *arXiv preprint arXiv:2504.04949*, 2025.

Dong Zhang, Gang Wang, Jinlong Xue, Kai Fang, Liang Zhao, Rui Ma, Shuhuai Ren, Shuo Liu, Tao Guo, Weiji Zhuang, et al. Mimo-audio: Audio language models are few-shot learners. *arXiv preprint arXiv:2512.23808*, 2025.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechtokenizer: Unified speech tokenizer for speech language models. In *International Conference on Learning Representations*, volume 2024, pp. 31798–31818, 2024.

Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. In *International Conference on Learning Representations*, volume 2025, pp. 90844–90868, 2025.

Haina Zhu, Yizhi Zhou, Hangting Chen, Jianwei Yu, Ziyang Ma, Rongzhi Gu, Yi Luo, Wei Tan, and Xie Chen. Muq: Self-supervised music representation learning with mel residual vector quantization. *IEEE Transactions on Audio, Speech and Language Processing*, 2025a.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22968–22977, 2025b.

A Layer Fusion

Large pre-trained audio backbones tend to conserve different types of information in their intermediate layers with lower layers typically associated with acoustical information and higher ones – with phonetic knowledge (Pasad et al., 2023; 2021). Therefore, using a combination of AuT layers has potential to capture useful audio features better than a single-layer. In this section, we validate whether layer fusion can further enhance the performance of the UFC codec.

A.1 Approaches

Given a set of layers L, we validate three layer-fusion approaches. As a baseline, we just concatenate the features across the channel dimension:

$$h_{concat} = [h_{l_0}|h_{l_1}|\dots|h_{l_{L-1}}] \in \mathbb{R}^{T_A \times (D_h|L|)}, \tag{2}$$

where l_i are layers from the set L. The resulting representation is passed directly to the decoder $D(\cdot)$ to get the reconstructed waveform. Since different layers might return output of different magnitudes, hurting the balance between features, and, hence, the optimization, we consider a normalized version too:

$$h_{concat,norm} = [\text{LayerNorm}_{l_0}(h_{l_0})|\dots|\text{LayerNorm}_{l_{L-1}}(h_{l_{L-1}})] \in \mathbb{R}^{T_A \times (D_h|L|)}. \tag{3}$$

However, concatenation leads to a large increase in the feature size ($D_h \to D_h|L|$) and corresponding computational burden of the algorithm. To tackle this, the other two methods preserve the size of the single-layer approach.

In "Weighted", we assign a trainable scalar α_l to each layer $l \in L$, so that $\sum_{l \in L} \alpha_l = 1$. Instead of a single feature h_l, the Upsacler $U(\cdot)$ receives the weighted sum h_{weighted}:

$$h_{\text{weighted}} = \sum_{l \in L} \alpha_l h_l. \tag{4}$$

Trainable weights allow UFC to automatically assign the importance of each layer. However, different layers may encode information on different magnitude scales, which can lead to the domination of features with large values even when their α_l is small. To compensate for this, we also explore the normalized variation of the weighted sum:

$$h_{\text{weighted, norm}} = \sum_{l \in L} \alpha_l \text{LayerNorm}_l(h_l). \tag{5}$$

The "Weighted" approach assigns the same value for all feature-channels. In "F-Weighted", we apply a more general method, where the weights $\alpha_{l,f}$ are assigned to each layer $l \in L$ and to each channel $f \in \{0, \ldots, D_h - 1\}$. The weights are normalized between layers and the final representation is obtained as a weighted sum:

$$h_{\text{f-weighted},f} = \sum_{l \in L} \alpha_{l,f} h_{l,f}, \qquad \sum_{l \in L} \alpha_{l,f} = 1 \quad \forall f \in \{0, \ldots, D_h - 1\}. \tag{6}$$

Similarly to previous methods, we also validate a normalized approach, in which h_l is passed through a LayerNorm.

A.2 Results

Table 3: Comparison of UFC-NQ systems that use only a single hidden state versus those, which fuse several layers, on the dev-clean split of LibriSpeech (Panayotov et al., 2015).

Layers	Fusion Type	Norm.	dWER ↓	STOI ↑	PESQ ↑	ViSQOL ↑	UTMOS ↑	SCOREQ ↓	SPK ↑
0	×	×	3.67	0.93	2.89	**4.64**	3.28	0.28	98.5
4	×	×	**3.27**	0.93	2.83	4.59	3.29	0.27	98.4
13	×	×	3.94	0.91	2.42	4.30	3.52	**0.24**	98.1
28	×	×	5.12	0.84	1.50	3.58	3.65	0.31	94.2
4,28	Concatenation	×	3.68	**0.94**	2.89	4.61	3.30	0.28	**98.6**
4,28	Concatenation	✓	3.55	0.93	2.77	4.60	3.22	0.27	98.3
0,4,13,28	Weighted	×	3.38	0.93	2.83	4.60	3.28	0.28	98.5
0,4,13,28	Weighted	✓	3.43	0.93	**2.90**	4.62	3.30	0.27	98.5
0,4,13,28	F-Weighted	×	3.68	0.92	2.81	4.56	**3.32**	0.28	98.3
0,4,13,28	F-Weighted	✓	3.45	0.93	2.80	4.58	3.29	0.26	98.4

In Section 5.3, we observe the decreased semantic performance in later layers. If it is caused by metrics being dependent on speech quality, the fusion of layers from different indexes should excel in both reconstruction and conservation of semantic knowledge. To verify this, we compare layers 0, 4, 13, and 28 with their combination using several approaches described in Section A.1. Layers $0, 13, 28$ achieve the best performance in several metrics and layer 4 gives a good trade-off within the first few indexes. The results of the evaluation are presented in Table 3. For Concatenation, adding normalization adds benefits in terms of dWER in exchange for worse speech quality. In contrast, LayerNorm is clearly helpful for weighted fusion approaches supporting our claim on different magnitudes undesirably impacting importance scores. However, none of the fusion approaches is remarkably better than just using one of the first few layers, while adding computational overhead.

B Upscaler Importance

Table 4: Comparison of different Upscaler FocalDecoder configurations on the dev-clean partition of LibriSpeech (Panayotov et al., 2015). Vocos frame-rate shows the actual frame-rate the Vocos decoder is operating on after upscaling.

Upscaling Factors	Vocos Frame-Rate	dWER ↓	STOI ↑	PESQ ↑	ViSQOL ↑	UTMOS ↑	SCOREQ ↓	SPK ↑
×	13	4.51	0.92	2.38	4.47	2.72	0.45	97.5
2	26	3.65	0.93	2.65	4.58	3.15	0.32	98.3
2×2	52	**3.15**	**0.94**	2.91	**4.64**	3.30	0.27	**98.6**
$1 \times 2 \times 2$	52	3.18	**0.94**	2.96	4.63	**3.35**	0.26	**98.6**
$2 \times 2 \times 1$	52	3.52	**0.94**	2.93	4.63	3.32	0.26	98.5
$1 \times 2 \times 2 \times 1$	52	3.55	**0.94**	2.88	4.59	3.24	0.27	98.2
$2 \times 2 \times 2$	104	3.27	**0.94**	**3.00**	4.61	3.30	**0.25**	98.5
$1 \times 2 \times 2 \times 2$	104	3.48	0.93	2.85	4.56	3.22	0.30	98.4
$1 \times 2 \times 2 \times 2 \times 1$	104	3.32	0.93	2.86	4.55	3.22	0.29	98.4

To validate the necessity of the Upscaler module, we run several configurations on the train-clean-100 split of Librispeech (Panayotov et al., 2015) and evaluate them on the development split of LibriSpeech (Panayotov et al., 2015). More specifically, we either remove the Upscaler completely or change the number of FocalDecoder layers and their upscaling factors. In these experiments, UFC

Table 5: The number of elements and audio hours inside our corpus in terms of the amount of original data and the amount corresponding to random 6-seconds-long crops extracted for training.

Type	Dataset	# Elements (Originally)	# Crops	# Hours (Originally)	# Hours (Crops)
Speech	LibriSpeech	281241	14080000	961.05	23466.67

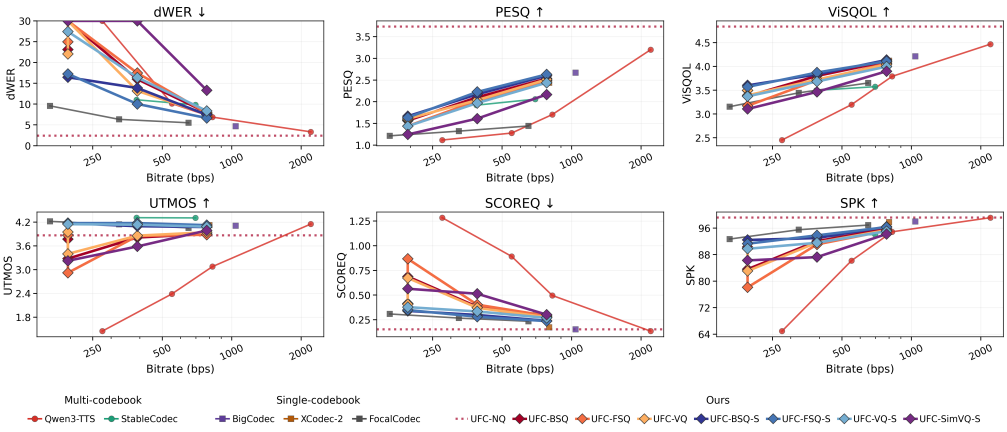


Figure 7: Metrics versus bitrate for different speech-only NACs and UFC on the test-split of LibriSpeech (Panayotov et al., 2015). Additionally, we show results for the speech-only UFC (marked with "-S") and for different quantization algorithms. At 195 bps, the two UFC points correspond to UFC-13 and UFC-52-13, with UFC-13 being the better-performing point.

uses the 4-th intermediate layer. Each model is trained for around 50k steps (12 hours on 1xNVIDIA GH200) with a batch size set to 16 using the same optimization setup described in Section 4. The performance measurements are presented in Table 4. As discussed in Section 3.1, 13 Hz frame-rate leads to a too large hop size for STFT with subsequent poor temporal resolution. Using Upscaler $U(\cdot)$ to increase frame-rate to 26 Hz already boosts reconstruction quality and intelligibility. Further upscaling to 52 Hz pushes the performance even further, however, the highest frame-rate of 104 Hz adds only limited gains in terms of PESQ and SCOREQ, while losing to the 52 Hz in other measurements. Given that the initial frame rate is very low, $\times 8$ upscaling does not add more details compared to $\times 4$, making 104 Hz less efficient. Within different upscaling factor settings, 2×2 and $1 \times 2 \times 2$ achieve the best performance with the latter chosen as our final option.

C Speech-Only Ablation

When trained jointly on speech, music, and sound, a universal NAC must allocate its representation capacity across heterogeneous domains. Prior work (Jiang et al., 2025; Chen et al., 2026) shows that VQ-based systems can struggle with domain disentanglement in the latent space and may require explicit mechanisms such as split or nested codebooks. We therefore test whether the same issue arises in UFC with different quantization schemes.

To this end, we train speech-only variants, denoted UFC-*-S, by optimizing the compressor–decompressor modules only on the LibriSpeech training set (Panayotov et al., 2015). The dataset is oversampled $50\times$ (Table 5), and the models are trained for 110K steps following the setup from Section 4.2.

We compare UFC-S with speech-only single-codebook NACs, including BigCodec (Xin et al., 2024), XCodec-2 (Ye et al., 2025b), and FocalCodec (Della Libera et al., 2026), as well as multi-codebook systems such as Qwen3-TTS-Tokenizer (Hu et al., 2026a) and StableCodec (Parker et al., 2025). Results on LibriSpeech test-clean are shown in Table 12 and Figure 7.

Compared with their universal counterparts, UFC-S variants with BSQ and FSQ improve substantially in dWER, UTMOS, and SCOREQ, indicating better preservation of semantic information

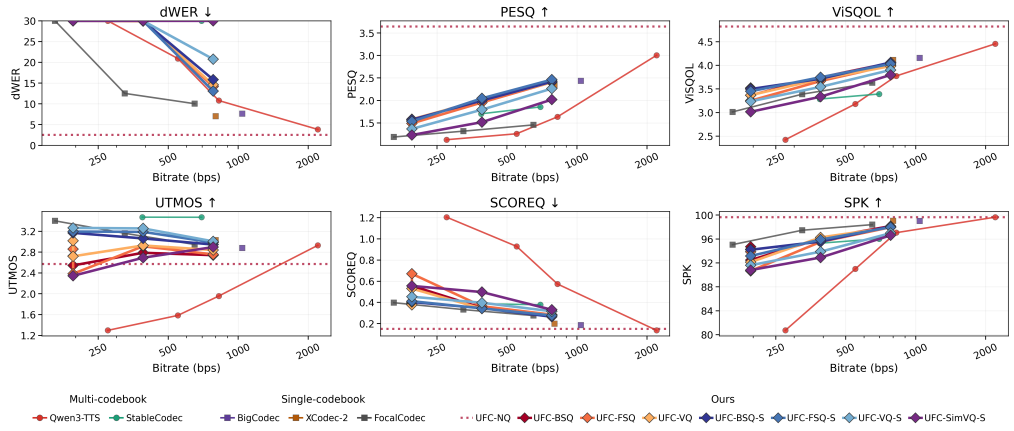


Figure 8: Metrics versus bitrate for different speech-only NACs and UFC on the non-English test split of Multilingual LibriSpeech (Pratap et al., 2020). Additionally, we show results for the speech-only UFC (marked with "-S") and for different quantization algorithms. At 195 bps, the two UFC points correspond to UFC-13 and UFC-52-13, with UFC-13 being the better-performing point.

and perceptual quality. PESQ and ViSQOL also improve, although generally by smaller margins. Nevertheless, speech-only training still does not recover the UFC-NQ upper bound, confirming that quantization remains a major source of information loss. The universal-to-speech-only gap is most pronounced at the lower 13 Hz and 26 Hz frame rates, whereas UFC-52 is affected only marginally by universal training. This suggests that the domain-disentanglement issue previously reported for VQ-based universal codecs also affects the implicit BSQ and FSQ quantizers at highly compressed temporal representations.

Interestingly, VQ does not exhibit the same systematic benefit from speech-only training: several objective metrics deteriorate for UFC-VQ-S, despite improvements in UTMOS and SCOREQ. We attribute this less consistent behavior to the optimization sensitivity of VQ with large codebooks. Notably, the small universal-to-speech-only gap of UFC-52 contrasts with previous universal NACs (Chen et al., 2026; Jiang et al., 2025; Ji et al., 2025) operating at comparable frame rates, which introduced split or nested codebooks to reduce domain interference. This suggests that domain disentanglement is determined not only by the quantization algorithm, but also by the underlying model architecture and training setup. We leave improved domain-aware code allocation at low frame rates as future work and provide preliminary results in Appendix D.

SimVQ struggles to converge across all frame rates, with the issue being especially pronounced at 13 Hz, where perceptual quality and intelligibility degrade substantially. Vanilla VQ commonly relies on stabilization techniques such as k-means initialization and EMA codebook updates (Zeghidour et al., 2021), while SimVQ provides an alternative mechanism intended to improve optimization stability. Its poor convergence in our setting highlights the sensitivity of VQ-based quantization to the optimization procedure and the need for careful training design. Since we observe no consistent metric advantage from VQ-based methods over BSQ or FSQ, these results further motivate the use of implicit-codebook quantizers.

Compared with other speech-only NACs, UFC-S is superior in PESQ and ViSQOL and on par on most remaining metrics at comparable bitrates. FocalCodec (Della Libera et al., 2026) is substantially superior in dWER at low frame rates, although the gap becomes small at 52 Hz, while XCodec-2 (Ye et al., 2025b) is superior in SCOREQ. At low bitrates, UFC-S is also superior to the multi-codebook Qwen3-TTS tokenizer (Hu et al., 2026a) on several metrics.

The same overall trends extend to multilingual speech, as shown in Figure 8 and Table 13, although the effect of speech-only training differs from the English case. At 52 Hz, the speech-only compressor–decompressor generalizes well to non-English speech despite being trained only on English data, with only small differences from the universal variants across most metrics. At lower frame rates, however, dWER degrades substantially, indicating weaker cross-lingual preservation of semantic information. Since the UFC-NQ backbone and AuT encoder are multilingual, these

results suggest that their intermediate representations h_l and u_l support cross-lingual quantization, but aggressive temporal compression makes this generalization more difficult.

D SHARED QUANTIZER ABLATION

In Appendix C, we observe a domain-disentanglement problem at low frame rates: universal training degrades speech reconstruction compared with speech-only training, suggesting that the quantizer struggles to allocate codes across heterogeneous audio domains. Here, we investigate whether structured codebook designs, previously used by universal NACs (Chen et al., 2026; Jiang et al., 2025), can reduce this gap for UFC.

Split and nested codebooks restrict each audio domain to a subset of the available codes. UniCodec (Jiang et al., 2025), for example, partitions its VQ codebook into disjoint speech, music, and sound regions. AUV (Chen et al., 2026) instead uses a nested design motivated by the fact that audio can contain mixtures of domains, such as vocals in music. However, its sharing structure does not cover all possible domain mixtures; for example, sound cannot access speech-specific codes. Since speech may contain music and general sound as background noise, having such a mixture is important. We therefore adopt a fully symmetric sharing scheme in which every pair of domains, as well as all three domains jointly, is assigned shared capacity.

We propose *SharedVQ*, which combines domain-specific, pairwise-shared, and globally shared codebook regions. Given a codebook $\{c_0, \ldots, c_{|V_{\text{codec}}|-1}\}$ with $|V_{\text{codec}}| = 32768$, we partition it into seven groups:

- **Domain-specific**: 4096 codes for each of speech, music, and sound.
- **Pairwise-shared**: 4096 codes for each speech–music, speech–sound, and music–sound pair.
- **Globally shared**: 8192 codes accessible to all three domains.

Consequently, each domain can access $4096 + 2 \cdot 4096 + 8192 = 20480$ codes, while the full vocabulary remains 32768.

During training, each sample is assigned a domain according to its source dataset (Table 1), and nearest-neighbor search is restricted to all codebook regions containing that domain. For example, a speech sample can use the speech-specific, speech–music, speech–sound, and globally shared regions. Thus, unlike strictly split codebooks, SharedVQ allows every domain to share codes with every other domain while retaining dedicated capacity for domain-specific information.

We hypothesize that clean, domain-specific signals can benefit from dedicated codes, whereas mixed or noisy signals can exploit pairwise or globally shared regions. This inductive bias is intended to improve domain disentanglement and reduce the universal-to-speech-only performance gap observed with unconstrained VQ.

We train UFC-SharedVQ on the universal corpus (shown in Table 1) following Section 4.2 and compare it with universal UFC-VQ and speech-only UFC-VQ-S.

As shown in Table 6, SharedVQ largely preserves the performance of vanilla VQ across acoustic and perceptual metrics, while its effect on dWER depends strongly on the frame rate. At $52\,\text{Hz}$, SharedVQ improves dWER from 7.60 to 6.58 on LibriSpeech and from 14.59 to 10.62 on multilingual speech, while also improving several reconstruction metrics. At $26\,\text{Hz}$, the two quantizers perform similarly, whereas at $13\,\text{Hz}$ SharedVQ slightly degrades English dWER and more noticeably degrades multilingual dWER.

One possible explanation is that domain-aware code allocation becomes less effective as the frame rate decreases. At $52\,\text{Hz}$, each token represents a relatively short temporal region, allowing the quantizer to choose more precisely between domain-specific, pairwise-shared, and globally shared codebook regions as the local audio content changes. At lower frame rates, each token must summarize a substantially longer segment that is more likely to contain heterogeneous content, making its assignment to a single codebook region less clear. Moreover, fewer tokens are available to represent the same audio duration, reducing the quantizer's freedom to alternate between domain-specific and shared codes.

Table 6: Comparison of SharedVQ quantization algorithm performance with vanilla universal VQ and speech-only-trained VQ on speech data.

Codec	TPS	BR	dWER ↓	STOI ↑	PESQ ↑	ViSQOL ↑	UTMOS ↑	SCOREQ ↓	SPK ↑
English Speech, LibriSpeech Panayotov et al. (2015)									
UFC-NQ (ours)	13×1	–	2.41	0.97	3.73	4.84	3.87	0.15	99.2
UFC-13-VQ (ours)	13×1	195	22.06	0.82	1.59	3.50	3.95	0.41	89.9
UFC-13-SharedVQ (ours)	13×1	195	23.03	0.83	1.55	3.50	3.83	0.43	90.4
UFC-13-VQ-S (ours)	13×1	195	27.43	0.81	1.44	3.37	4.15	0.38	89.8
UFC-52-VQ (ours)	52×1	780	7.60	0.90	2.51	4.04	3.94	0.28	95.4
UFC-52-SharedVQ (ours)	52×1	780	6.58	0.90	2.62	4.15	3.90	0.26	96.4
UFC-52-VQ-S (ours)	52×1	780	8.36	0.89	2.44	4.00	4.08	0.27	95.1
UFC-52-26-VQ (ours)	26×1	390	13.17	0.86	2.04	3.72	3.86	0.37	91.6
UFC-52-26-SharedVQ (ours)	26×1	390	13.70	0.86	2.11	3.80	3.80	0.38	92.6
UFC-52-26-VQ-S (ours)	26×1	390	16.38	0.85	1.97	3.68	4.13	0.33	91.6
Multilingual Non-English Speech, Multilingual LibriSpeech Pratap et al. (2020)									
UFC-NQ (ours)	13×1	–	2.52	0.92	3.64	4.82	2.57	0.15	99.6
UFC-13-VQ (ours)	13×1	195	54.66	0.80	1.56	3.51	3.02	0.38	94.3
UFC-13-SharedVQ (ours)	13×1	195	63.15	0.80	1.53	3.49	2.86	0.39	94.1
UFC-13-VQ-S (ours)	13×1	195	132.41	0.76	1.37	3.24	3.27	0.45	91.6
UFC-52-VQ (ours)	52×1	780	14.59	0.86	2.40	3.98	2.84	0.28	97.9
UFC-52-SharedVQ (ours)	52×1	780	10.62	0.86	2.52	4.10	2.72	0.27	98.4
UFC-52-VQ-S (ours)	52×1	780	20.78	0.84	2.26	3.90	3.01	0.32	97.0
UFC-52-26-VQ (ours)	26×1	390	29.91	0.82	2.01	3.71	2.93	0.34	96.3
UFC-52-26-SharedVQ (ours)	26×1	390	30.77	0.82	2.05	3.76	2.74	0.35	96.3
UFC-52-26-VQ-S (ours)	26×1	390	78.71	0.80	1.79	3.54	3.26	0.40	93.9

Table 7: Comparison of SharedVQ quantization algorithm performance with vanilla universal VQ and speech-only-trained VQ on music and sound data.

Codec	TPS	BR	STFT ↓	MEL ↓	ViSQOL ↑	CE ↑	CU ↑	PC ↑	PQ ↑	SIM ↑
Music, SongDesriber Manco et al. (2023)										
UFC-NQ (ours)	13×1	–	3.53	1.93	4.53	7.27	7.59	5.86	7.65	97.1
UFC-13-VQ (ours)	13×1	195	4.04	2.26	3.92	6.97	7.32	5.26	7.54	82.8
UFC-13-SharedVQ (ours)	13×1	195	4.06	2.75	3.85	6.95	7.26	5.25	7.39	80.3
UFC-52-VQ (ours)	52×1	780	3.62	2.01	4.16	7.18	7.47	5.39	7.68	90.9
UFC-52-SharedVQ (ours)	52×1	780	3.59	2.00	4.21	7.22	7.51	5.47	7.69	91.9
UFC-52-26-VQ (ours)	26×1	390	3.63	2.05	4.05	6.97	7.32	5.06	7.55	87.9
UFC-52-26-SharedVQ (ours)	26×1	390	3.66	2.06	4.07	7.06	7.37	5.18	7.58	87.9
Sound, AudioSet Gemmeke et al. (2017)										
UFC-NQ (ours)	13×1	–	2.16	1.13	4.53	4.37	5.54	4.10	6.18	96.6
UFC-13-VQ (ours)	13×1	195	2.98	1.85	3.90	4.05	5.09	3.63	5.76	79.3
UFC-13-SharedVQ (ours)	13×1	195	3.01	2.02	3.88	4.11	5.17	3.68	5.86	79.5
UFC-52-VQ (ours)	52×1	780	2.51	1.50	4.15	4.24	5.24	3.89	5.92	85.0
UFC-52-SharedVQ (ours)	52×1	780	2.48	1.47	4.18	4.25	5.25	3.96	5.94	86.6
UFC-52-26-VQ (ours)	26×1	390	2.60	1.60	4.05	4.12	5.21	3.60	5.86	81.7
UFC-52-26-SharedVQ (ours)	26×1	390	2.61	1.60	4.06	4.14	5.19	3.67	5.85	82.3

Compared with UFC-VQ-S, SharedVQ is substantially more robust on non-English speech due to its multilingual universal training. Nevertheless, neither SharedVQ nor the speech-only variants recover the UFC-NQ upper bound, confirming that quantization remains a major source of information loss.

The same trend extends to music and sound, as shown in Table 7. At 52 Hz, SharedVQ consistently matches or slightly surpasses vanilla VQ across both domains, including semantic similarity. At 26 Hz, the differences are small, while at 13 Hz SharedVQ provides no consistent benefit and slightly degrades music reconstruction.

Overall, SharedVQ is most effective at the higher 52 Hz frame rate, where it improves semantic preservation while maintaining or improving reconstruction quality. At lower frame rates, however, structured domain-aware code allocation alone does not resolve the universal-to-domain-specific gap. This suggests that split or nested codebook strategies that are effective for higher-frame-rate universal NACs may be insufficient for highly compressed temporal representations, motivating alternative approaches for domain disentanglement at ultra-low token rates. The future work may focus on the development of these approaches.

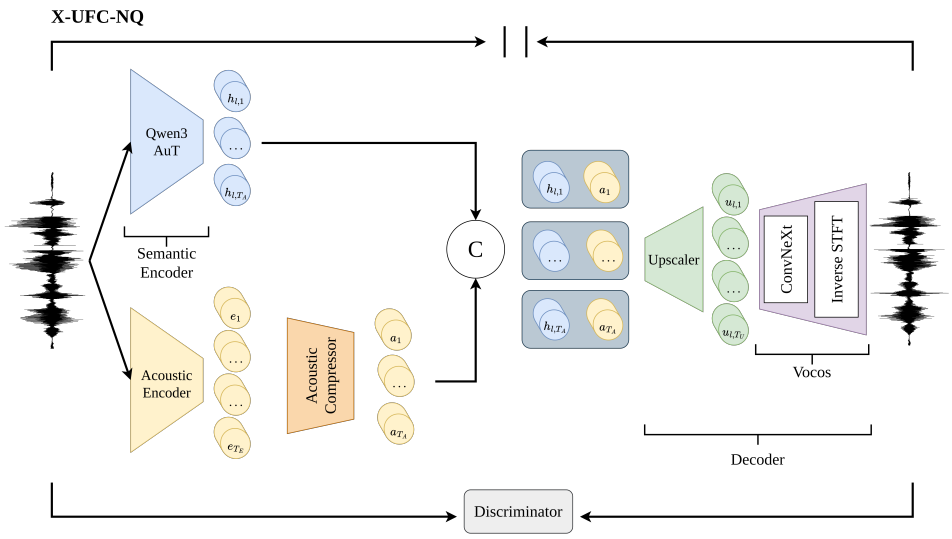


Figure 9: The architecture of the X-UFC-NQ. The circle with a letter C stands for concatenation across the feature dimension. X-UFC follows the original UFC design but adds a separate acoustic branch to restore phase and other missing acoustic information. We visualize only the non-quantized (NQ) variant. The quantized models follow the original UFC methodology and either discretize the concatenated features directly or operate on the upscaled features.

E X-UFC

Our proposed UFC system builds on the Qwen3-Omni Audio Transformer (AuT) encoder (Xu et al., 2025b). AuT operates on Mel-spectrogram inputs and therefore does not explicitly preserve waveform phase. Although this representation is sufficient for AuT's strong perceptual and semantic performance in Qwen3-Omni, waveform reconstruction in UFC must recover phase information implicitly from magnitude-based features.

We investigate whether providing an additional waveform-based acoustic representation with phase information can further improve reconstruction quality. To this end, we introduce a parallel acoustic branch that operates directly on the input waveform and concatenate its output with the AuT features. Let $A(\cdot)$ denote the acoustic encoder. Given an input waveform $x \in \mathbb{R}^T$, we compute acoustic features with $A(x)$ and pass them through a FocalEncoder-based downsampler $D_{\text{acoustic}}(\cdot)$ to match the temporal resolution of the AuT representation $h_l \in \mathbb{R}^{T_A \times D_h}$. The resulting representation $a = D_{\text{acoustic}}(A(x)) \in \mathbb{R}^{T_A \times D_a}$ is concatenated with h_l along the feature dimension, yielding $h_{\text{X-UFC},l} = [h_l; a]$, which is then passed to the codec decoder. We refer to this dual-branch variant as X-UFC. The X-UFC pipeline is shown in Figure 9.

As in UFC, X-UFC is trained in two stages following Section 4.2. First, we train a non-quantized model in which the decoder reconstructs the waveform directly from the concatenated representation $h_{\text{X-UFC},l}$. In the second stage, we insert the compressor–quantizer–decompressor module and optimize it to discretize the learned latent representation. Since DAC (Kumar et al., 2023) is a state-of-the-art acoustic autoencoder, we use its pretrained encoder as $A(\cdot)$ and keep it frozen throughout training. Only the acoustic downsampler $D_{\text{acoustic}}(\cdot)$ and the UFC-NQ decoder are optimized in the first stage, followed by quantizer training in the second stage.

This design is conceptually related to X-Codec-2 (Ye et al., 2025b), which also combines semantic and waveform-based acoustic representations. However, the optimization procedure differs. X-Codec-2 employs separate waveform and semantic reconstruction objectives from a concatenated representation and jointly optimizes the quantizer together with the decoder. In contrast, X-UFC follows the two-stage training strategy of FocalCodec (Della Libera et al., 2026): the continuous representation and decoder are first trained without quantization, after which the quantization module is optimized separately. This decouples waveform reconstruction from discrete representation learning and allows us to evaluate whether adding explicit waveform-derived acoustic information improves

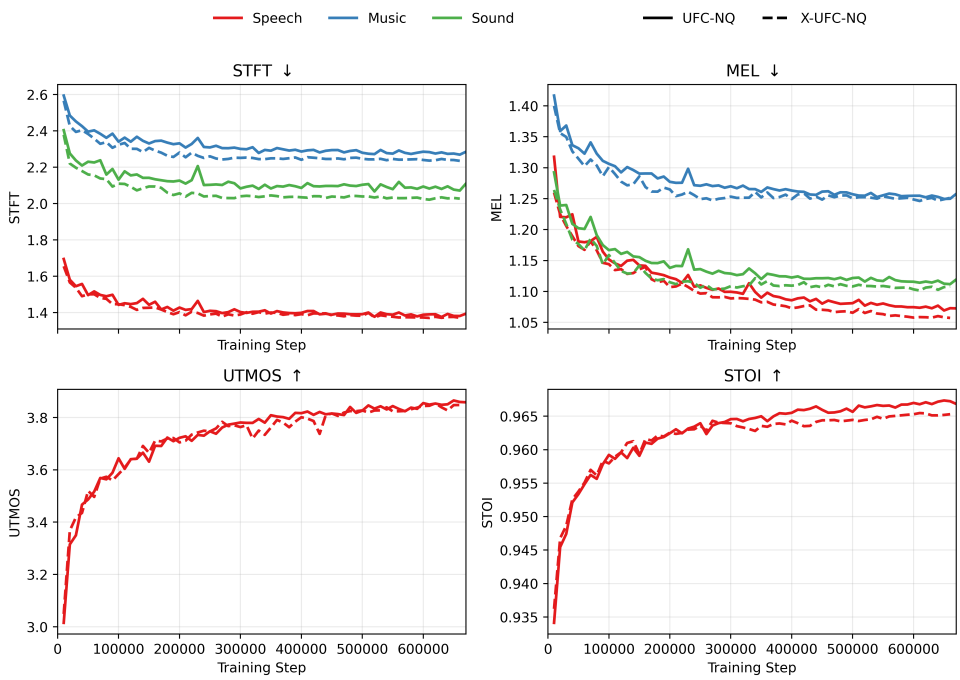


Figure 10: Metric vs step curves for the first 670K steps of UFC-NQ and X-UFC-NQ training.

UFC without introducing an additional semantic reconstruction decoder. Since direct waveform reconstruction is a substantially harder objective than discretizing an already learned compressed latent space, optimizing the decoder first and the quantizer afterwards may provide a more effective training strategy than jointly learning both objectives from scratch, as in X-Codec-2.

Due to time constraints, we train X-UFC-NQ for only 670K out of the 740K training steps used for UFC-NQ, making a direct comparison between the final models imperfect. Nevertheless, the metric trajectories in Figure 10 show that the additional acoustic branch consistently improves the time–frequency reconstruction metrics, reducing both STFT and Mel distances, especially for sound and music. In contrast, UTMOS remains nearly unchanged, while STOI slightly decreases. Since the acoustic branch operates directly on the waveform, it can provide fine-grained acoustic information, including information unavailable from the Mel-spectrogram input alone, which likely explains the improved spectral reconstruction. However, these gains do not translate into improved perceptual quality or intelligibility, suggesting that the original UFC representation already contains sufficient information for high-quality waveform reconstruction and benefits only marginally from the additional waveform-based features. Comparison on the final checkpoint and evaluation of quantized variants is the direction of the future work.

F Metric Tables

In this Section, we report the precise metric values for the results presented in Section 5. Tables 8 and 9 show the comparison of universal NACs on English and multilingual speech data, respectively. The measurements on music and sound are presented in Tables 10 and 11, respectively. For the speech-only ablation, the comparison is reported in Tables 12 and 13 for English and multilingual data, respectively.

Table 8: Results on the test-clean partition of LibriSpeech (Panayotov et al., 2015). NACs marked with * operate on 24 kHz sampling rate. For them, we resample input to 24 kHz and downsample their output back to 16 kHz for a fair comparison with other 16 kHz codecs. Note that the semantic part of OmniCodec operates on 16 kHz as well.

Codec	TPS	Codebook	BR (bps)	# Params	dWER ↓	STOI ↑	PESQ ↑	ViSQOL ↑	UTMOS ↑	SCOREQ ↓	SPK ↑
Universal Multi-Codebook NACs											
DAC Kumar et al. (2023)	50×12	1024	6000	74M	2.86	0.97	4.02	4.69	4.00	0.11	99.3
DAC Kumar et al. (2023)	50×3	1024	1500	74M	8.80	0.79	1.33	3.38	1.48	0.98	92.2
DAC Kumar et al. (2023)	50×2	1024	1000	74M	15.01	0.73	1.20	2.85	1.29	1.22	88.5
DAC Kumar et al. (2023)	50×1	1024	500	74M	131.40	0.62	1.10	1.96	1.25	1.40	74.1
Encodec* Défossez et al. (2023)	75×8	1024	6000	23M	5.72	0.94	2.75	4.27	3.09	0.43	98.0
Encodec* Défossez et al. (2023)	75×2	1024	1500	23M	9.75	0.84	1.55	3.59	1.58	1.02	93.4
SemantiCodec Liu et al. (2024)	100×2	32768/8192	1400	595M	5.75	0.88	2.08	4.06	2.95	0.51	96.1
SemantiCodec Liu et al. (2024)	50×2	32768/8192	700	1033M	7.93	0.86	1.83	3.83	2.94	0.55	94.7
SemantiCodec Liu et al. (2024)	25×2	32768/8192	350	2036M	25.92	0.79	1.42	3.38	2.78	0.72	90.7
OmniCodec* Hu et al. (2026b)	12.5×33	2048	4537	786M	4.63	0.96	3.01	4.52	3.99	0.14	99.2
OmniCodec* Hu et al. (2026b)	12.5×8	2048	1100	786M	7.46	0.91	2.23	4.09	3.63	0.26	97.4
OmniCodec* Hu et al. (2026b)	12.5×6	2048	825	786M	8.42	0.89	1.97	3.92	3.38	0.34	96.2
OmniCodec* Hu et al. (2026b)	12.5×5	2048	687	786M	8.93	0.87	1.78	3.78	3.16	0.42	95.0
OmniCodec* Hu et al. (2026b)	12.5×4	2048	550	786M	9.87	0.84	1.57	3.57	2.80	0.57	92.8
OmniCodec* Hu et al. (2026b)	12.5×2	2048	275	786M	22.05	0.71	1.18	2.67	1.41	1.18	72.1
MOSS* Gong et al. (2026)	12.5×32	1024	4000	1775M	3.72	0.97	3.52	4.63	4.09	0.10	99.5
MOSS* Gong et al. (2026)	12.5×8	1024	1000	1775M	5.97	0.94	2.82	4.28	4.15	0.15	98.4
MOSS* Gong et al. (2026)	12.5×6	1024	750	1775M	7.01	0.93	2.58	4.15	4.14	0.17	97.7
MOSS* Gong et al. (2026)	12.5×4	1024	500	1775M	8.16	0.90	2.22	3.94	4.12	0.21	96.2
MOSS* Gong et al. (2026)	12.5×2	1024	250	1775M	11.65	0.83	1.51	3.49	3.48	0.37	92.0
MOSS* Gong et al. (2026)	12.5×1	1024	125	1775M	22.48	0.74	1.18	2.94	2.56	0.71	86.5
HiggsV2* Boson AI (2025)	25×8	1024	2000	201M	3.01	0.95	3.09	4.41	4.00	0.15	98.7
HiggsV2* Boson AI (2025)	25×4	1024	1000	201M	5.42	0.92	2.58	4.11	3.89	0.22	97.2
HiggsV2* Boson AI (2025)	25×2	1024	500	201M	10.95	0.87	1.95	3.70	3.28	0.50	92.8
Universal Single-Codebook NACs											
WavTokenizer-40* Ji et al. (2025)	40×1	4096	480	85M	11.99	0.87	1.97	3.74	3.78	0.31	93.9
WavTokenizer-75* Ji et al. (2025)	75×1	4096	900	85M	8.27	0.91	2.52	4.13	4.00	0.23	95.7
UniCodec* Jiang et al. (2025)	75×1	16384	1050	211M	4.71	0.93	2.71	4.28	4.05	0.19	97.5
AUV Chen et al. (2026)	50×1	20480	716	133M	7.39	0.91	2.42	4.15	4.09	0.18	97.6
L3AC Zhai et al. (2025)	59.26×1	117649	1000	11M	5.04	0.91	2.20	3.70	3.77	0.33	94.7
L3AC Zhai et al. (2025)	44.44×1	117649	750	11M	6.04	0.90	2.08	3.63	3.77	0.34	94.5
Ours – Universal Single-Codebook NACs											
UFC-NQ	13×1	–	–	275M	2.41	0.97	3.73	4.84	3.87	0.15	99.2
UFC-NQ, $\lambda_{\text{acoustic}} = 10.0$, $\lambda_{\text{semantic}} = 0.0$	13×1	–	–	275M	2.86	0.96	3.62	4.79	3.85	0.16	99.2
UFC-NQ, $\lambda_{\text{acoustic}} = 0.0$, $\lambda_{\text{semantic}} = 0.0$	13×1	–	–	275M	2.69	0.96	3.62	4.81	3.75	0.17	99.1
UFC-13-BSQ	13×1	32768	195	362M	23.11	0.82	1.56	3.49	3.77	0.42	90.3
UFC-52-BSQ	52×1	32768	780	362M	7.95	0.90	2.53	4.10	3.90	0.28	95.7
UFC-52-26-BSQ	26×1	32768	390	362M	15.98	0.86	2.10	3.80	3.81	0.38	92.1
UFC-52-13-BSQ	13×1	32768	195	362M	52.31	0.80	1.56	3.38	3.28	0.69	83.6
UFC-13-FSQ	13×1	32768	195	362M	24.95	0.83	1.57	3.49	3.87	0.41	89.9
UFC-52-FSQ	52×1	32768	780	362M	7.88	0.90	2.50	4.06	3.88	0.29	95.2
UFC-52-26-FSQ	26×1	32768	390	362M	17.46	0.86	2.02	3.71	3.85	0.40	91.1
UFC-52-13-FSQ	13×1	32768	195	362M	71.00	0.77	1.44	3.17	2.92	0.87	78.1
UFC-13-VQ	13×1	32768	195	362M	22.06	0.82	1.59	3.50	3.95	0.41	89.9
UFC-52-VQ	52×1	32768	780	362M	7.60	0.90	2.51	4.04	3.94	0.28	95.4
UFC-52-26-VQ	26×1	32768	390	362M	13.17	0.86	2.04	3.72	3.86	0.37	91.6
UFC-52-13-VQ	13×1	32768	195	362M	50.68	0.80	1.61	3.39	3.40	0.67	83.1

Table 9: Results on the non-English test partition of Multilingual Librispeech (Pratap et al., 2020). NACs marked with * operate on 24 kHz sampling rate. For them, we resample input to 24 kHz and downsample their output back to 16 kHz for a fair comparison with other 16 kHz codecs. Note that the semantic part of OmniCodec operates on 16 kHz as well.

Codec	TPS	Codebook	BR (bps)	# Params	dWER ↓	STOI ↑	PESQ ↑	ViSQOL ↑	UTMOS ↑	SCOREQ ↓	SPK ↑
Universal Multi-Codebook NACs											
DAC Kumar et al. (2023)	50×12	1024	6000	74M	3.54	0.97	3.93	4.68	2.69	0.11	99.7
DAC Kumar et al. (2023)	50×3	1024	1500	74M	14.60	0.79	1.34	3.40	1.29	1.02	94.6
DAC Kumar et al. (2023)	50×2	1024	1000	74M	32.26	0.73	1.21	2.85	1.24	1.18	91.5
DAC Kumar et al. (2023)	50×1	1024	500	74M	303.69	0.60	1.11	1.84	1.23	1.30	85.2
Encodec* Défossez et al. (2023)	75×8	1024	6000	23M	8.59	0.94	2.68	4.26	2.01	0.50	99.1
Encodec* Défossez et al. (2023)	75×2	1024	1500	23M	20.40	0.84	1.56	3.59	1.33	1.04	96.2
SemantiCodec Liu et al. (2024)	100×2	32768/8192	1400	595M	9.05	0.87	1.98	4.04	1.84	0.62	98.0
SemantiCodec Liu et al. (2024)	50×2	32768/8192	700	1033M	17.73	0.83	1.72	3.78	1.86	0.65	97.4
SemantiCodec Liu et al. (2024)	25×2	32768/8192	350	2036M	70.77	0.76	1.38	3.32	1.86	0.72	95.4
OmniCodec* Hu et al. (2026b)	12.5×33	2048	4537	786M	4.21	0.95	2.83	4.52	2.70	0.14	99.7
OmniCodec* Hu et al. (2026b)	12.5×8	2048	1100	786M	7.27	0.89	2.08	4.07	2.31	0.30	98.8
OmniCodec* Hu et al. (2026b)	12.5×6	2048	825	786M	10.27	0.86	1.84	3.89	2.10	0.41	98.3
OmniCodec* Hu et al. (2026b)	12.5×5	2048	687	786M	13.38	0.84	1.69	3.75	1.91	0.51	97.7
OmniCodec* Hu et al. (2026b)	12.5×4	2048	550	786M	18.86	0.81	1.51	3.54	1.65	0.68	96.6
OmniCodec* Hu et al. (2026b)	12.5×2	2048	275	786M	69.07	0.68	1.17	2.60	1.25	1.16	81.2
MOSS* Gong et al. (2026)	12.5×32	1024	4000	1775M	3.66	0.97	3.50	4.64	2.94	0.11	99.8
MOSS* Gong et al. (2026)	12.5×8	1024	1000	1775M	6.16	0.93	2.79	4.28	3.07	0.17	99.4
MOSS* Gong et al. (2026)	12.5×6	1024	750	1775M	8.24	0.92	2.56	4.15	3.07	0.18	99.1
MOSS* Gong et al. (2026)	12.5×4	1024	500	1775M	14.26	0.89	2.18	3.93	3.00	0.21	98.2
MOSS* Gong et al. (2026)	12.5×2	1024	250	1775M	39.59	0.81	1.53	3.49	2.38	0.36	95.5
MOSS* Gong et al. (2026)	12.5×1	1024	125	1775M	131.77	0.72	1.21	2.97	1.79	0.66	92.0
HiggsV2* Boson AI (2025)	25×8	1024	2000	201M	4.19	0.94	2.93	4.40	2.72	0.16	99.4
HiggsV2* Boson AI (2025)	25×4	1024	1000	201M	7.36	0.91	2.44	4.09	2.59	0.25	98.7
HiggsV2* Boson AI (2025)	25×2	1024	500	201M	21.34	0.85	1.87	3.66	2.15	0.54	96.2
Universal Single-Codebook NACs											
WavTokenizer-40* Ji et al. (2025)	40×1	4096	480	85M	33.87	0.84	1.84	3.64	2.64	0.34	96.6
WavTokenizer-75* Ji et al. (2025)	75×1	4096	900	85M	13.86	0.90	2.35	4.08	2.90	0.24	97.8
UniCodec* Jiang et al. (2025)	75×1	16384	1050	211M	9.67	0.91	2.44	4.19	2.96	0.22	98.2
AUV Chen et al. (2026)	50×1	20480	716	133M	10.88	0.90	2.31	4.13	2.96	0.20	98.9
L3AC Zhai et al. (2025)	59.26×1	117649	1000	11M	7.18	0.90	2.10	3.64	2.58	0.40	97.2
L3AC Zhai et al. (2025)	44.44×1	117649	750	11M	8.46	0.88	1.98	3.56	2.61	0.41	96.7
Ours – Universal Single-Codebook NACs											
UFC-NQ	13×1	–	–	275M	2.52	0.92	3.64	4.82	2.57	0.15	99.6
UFC-NQ, $\lambda_{\text{acoustic}} = 10.0$, $\lambda_{\text{semantic}} = 0.0$	13×1	–	–	275M	2.73	0.92	3.52	4.78	2.55	0.16	99.6
UFC-NQ, $\lambda_{\text{acoustic}} = 0.0$, $\lambda_{\text{semantic}} = 0.0$	13×1	–	–	275M	2.70	0.92	3.51	4.79	2.47	0.18	99.6
UFC-13-BSQ	13×1	32768	195	362M	52.56	0.80	1.57	3.52	2.73	0.39	94.7
UFC-52-BSQ	52×1	32768	780	362M	14.54	0.85	2.43	4.05	2.74	0.28	98.2
UFC-52-26-BSQ	26×1	32768	390	362M	33.68	0.82	2.03	3.75	2.79	0.35	96.1
UFC-52-13-BSQ	13×1	32768	195	362M	119.07	0.76	1.56	3.37	2.54	0.56	92.5
UFC-13-FSQ	13×1	32768	195	362M	58.19	0.80	1.56	3.50	2.86	0.40	93.9
UFC-52-FSQ	52×1	32768	780	362M	14.16	0.86	2.41	4.00	2.76	0.28	98.0
UFC-52-26-FSQ	26×1	32768	390	362M	37.41	0.82	1.95	3.67	2.91	0.36	95.5
UFC-52-13-FSQ	13×1	32768	195	362M	165.55	0.74	1.49	3.25	2.39	0.67	90.7
UFC-13-VQ	13×1	32768	195	362M	54.66	0.80	1.56	3.51	3.02	0.38	94.3
UFC-52-VQ	52×1	32768	780	362M	14.59	0.86	2.40	3.98	2.84	0.28	97.9
UFC-52-26-VQ	26×1	32768	390	362M	29.91	0.82	2.01	3.71	2.93	0.34	96.3
UFC-52-13-VQ	13×1	32768	195	362M	128.41	0.76	1.58	3.37	2.72	0.53	92.1

Table 10: Results on the SongDescriber (Manco et al., 2023) dataset. NACs marked with * operate on 24 kHz sampling rate. For them, we resample input to 24 kHz and downsample their output back to 16 kHz for a fair comparison with other 16 kHz codecs. Note that the semantic part of OmniCodec operates on 16 kHz as well.

Codec	TPS	BR (bps)	STFT ↓	MEL ↓	ViSQOL ↑	CE ↑	CU ↑	PC ↑	PQ ↑	SIM ↑
Universal Multi-Codebook NACs										
DAC Kumar et al. (2023)	50×12	6000	2.27	1.02	4.57	7.33	7.62	5.86	7.71	98.0
DAC Kumar et al. (2023)	50×3	1500	3.48	1.88	4.10	6.12	5.86	6.00	6.04	77.1
DAC Kumar et al. (2023)	50×2	1000	3.76	2.29	3.92	5.68	5.36	5.85	5.51	66.7
DAC Kumar et al. (2023)	50×1	500	4.39	3.20	3.65	4.45	4.62	5.33	4.80	52.9
Encodec* Défossez et al. (2023)	75×8	6000	2.65	1.26	4.37	7.10	7.33	5.89	7.30	92.9
Encodec* Défossez et al. (2023)	75×2	1500	3.15	1.64	4.17	5.90	6.11	6.01	6.11	83.2
SemantiCodec Liu et al. (2024)	100×2	1400	3.10	1.60	4.17	7.32	7.58	5.93	7.66	92.5
SemantiCodec Liu et al. (2024)	50×2	700	3.23	1.76	4.05	7.30	7.58	5.84	7.69	90.9
SemantiCodec Liu et al. (2024)	25×2	350	3.43	1.87	3.92	7.39	7.60	5.92	7.70	88.3
OmniCodec* Hu et al. (2026b)	12.5×33	4537	2.94	1.37	4.40	7.16	7.44	5.88	7.47	86.2
OmniCodec* Hu et al. (2026b)	12.5×8	1100	3.21	1.54	4.24	6.87	7.00	5.91	6.96	79.3
OmniCodec* Hu et al. (2026b)	12.5×6	825	3.34	1.61	4.17	6.68	6.83	5.91	6.76	74.5
OmniCodec* Hu et al. (2026b)	12.5×5	687	3.43	1.67	4.11	6.46	6.67	5.90	6.57	70.2
OmniCodec* Hu et al. (2026b)	12.5×4	550	3.57	1.75	4.04	6.10	6.42	5.88	6.28	63.2
OmniCodec* Hu et al. (2026b)	12.5×2	275	4.04	2.13	3.80	4.47	5.25	5.16	5.59	38.5
MOSS* Gong et al. (2026)	12.5×32	4000	2.44	1.18	4.40	7.33	7.66	5.73	7.78	97.4
MOSS* Gong et al. (2026)	12.5×8	1000	2.77	1.43	4.26	7.24	7.61	5.68	7.71	95.6
MOSS* Gong et al. (2026)	12.5×6	750	2.87	1.50	4.21	7.20	7.59	5.68	7.68	94.5
MOSS* Gong et al. (2026)	12.5×4	500	3.03	1.61	4.13	7.11	7.53	5.71	7.58	92.0
MOSS* Gong et al. (2026)	12.5×2	250	3.34	1.84	4.00	6.79	7.35	5.76	7.30	85.0
MOSS* Gong et al. (2026)	12.5×1	125	3.63	2.11	3.87	6.26	7.08	5.73	6.86	71.3
HiggsV2* Boson AI (2025)	25×8	2000	2.63	1.30	4.39	7.29	7.50	5.86	7.54	96.3
HiggsV2* Boson AI (2025)	25×4	1000	2.86	1.46	4.28	7.24	7.43	5.89	7.47	93.9
HiggsV2* Boson AI (2025)	25×2	500	3.18	1.69	4.13	7.14	7.40	6.01	7.41	87.3
Universal Single-Codebook NACs										
WavTokenizer-40* Ji et al. (2025)	40×1	480	3.43	1.81	4.05	6.67	6.88	6.15	6.82	80.7
WavTokenizer-75* Ji et al. (2025)	75×1	900	3.74	1.79	4.10	4.46	5.13	5.26	5.56	59.0
UniCodec* Jiang et al. (2025)	75×1	1050	3.47	1.98	3.86	5.39	6.10	5.47	5.78	83.7
AUV Chen et al. (2026)	50×1	716	3.16	1.68	4.05	6.96	7.37	5.84	7.33	88.4
L3AC Zhai et al. (2025)	59.26×1	1000	3.26	2.04	4.15	7.18	7.55	5.74	7.46	84.7
L3AC Zhai et al. (2025)	44.44×1	750	3.34	2.07	4.09	7.10	7.48	5.71	7.40	82.4
Ours – Universal Single-Codebook NACs										
UFC-NQ	13×1	–	3.53	1.93	4.53	7.27	7.59	5.86	7.65	97.1
UFC-NQ, $\lambda_{\text{acoustic}} = 10.0$, $\lambda_{\text{semantic}} = 0.0$	13×1	–	3.57	1.94	4.49	7.20	7.51	5.84	7.54	96.7
UFC-NQ, $\lambda_{\text{acoustic}} = 0.0$, $\lambda_{\text{semantic}} = 0.0$	13×1	–	3.55	1.92	4.49	7.24	7.54	5.82	7.59	97.3
UFC-13-BSQ	13×1	195	4.28	2.45	3.93	6.91	7.22	5.32	7.37	83.0
UFC-52-BSQ	52×1	780	3.59	2.01	4.19	7.23	7.48	5.43	7.67	91.3
UFC-52-26-BSQ	26×1	390	3.64	2.06	4.06	7.07	7.36	5.22	7.58	87.9
UFC-52-13-BSQ	13×1	195	3.68	2.11	3.94	6.79	7.12	4.81	7.38	82.5
UFC-13-FSQ	13×1	195	4.08	2.26	3.93	6.80	7.19	5.27	7.35	82.7
UFC-52-FSQ	52×1	780	3.59	2.03	4.16	7.19	7.48	5.41	7.66	90.6
UFC-52-26-FSQ	26×1	390	3.67	2.08	4.04	7.03	7.33	5.10	7.56	87.4
UFC-52-13-FSQ	13×1	195	3.76	2.36	3.88	6.86	7.22	4.78	7.45	81.8
UFC-13-VQ	13×1	195	4.04	2.26	3.92	6.97	7.32	5.26	7.54	82.8
UFC-52-VQ	52×1	780	3.62	2.01	4.16	7.18	7.47	5.39	7.68	90.9
UFC-52-26-VQ	26×1	390	3.63	2.05	4.05	6.97	7.32	5.06	7.55	87.9
UFC-52-13-VQ	13×1	195	3.71	2.14	3.93	6.80	7.13	4.78	7.42	84.0

Table 11: Results on the evaluation partition of AudioSet (Gemmeke et al., 2017). NACs marked with * operate on 24 kHz sampling rate. For them, we resample input to 24 kHz and downsample their output back to 16 kHz for a fair comparison with other 16 kHz codecs. Note that the semantic part of OmniCodec operates on 16 kHz as well.

Codec	TPS	BR (bps)	STFT ↓	MEL ↓	ViSQOL ↑	CE ↑	CU ↑	PC ↑	PQ ↑	SIM ↑
Universal Multi-Codebook NACs										
DAC Kumar et al. (2023)	50×12	6000	1.97	1.01	4.53	4.37	5.50	4.14	6.15	97.1
DAC Kumar et al. (2023)	50×3	1500	2.90	1.84	4.03	3.58	4.60	4.43	5.42	81.1
DAC Kumar et al. (2023)	50×2	1000	3.14	2.24	3.83	3.52	4.45	4.52	5.20	75.3
DAC Kumar et al. (2023)	50×1	500	3.72	3.15	3.59	3.24	4.43	4.23	5.00	71.9
Encodec* Défossez et al. (2023)	75×8	6000	2.44	1.34	4.22	4.19	5.33	4.15	5.91	95.5
Encodec* Défossez et al. (2023)	75×2	1500	2.78	1.69	4.08	3.73	4.74	4.37	5.43	87.1
SemantiCodec Liu et al. (2024)	100×2	1400	2.40	1.42	4.23	4.33	5.42	4.24	6.10	93.7
SemantiCodec Liu et al. (2024)	50×2	700	2.48	1.51	4.13	4.31	5.40	4.20	6.09	91.8
SemantiCodec Liu et al. (2024)	25×2	350	2.70	1.70	3.98	4.27	5.31	4.26	6.03	89.3
OmniCodec* Hu et al. (2026b)	12.5×33	4537	2.47	1.32	4.25	4.29	5.44	4.07	6.04	95.4
OmniCodec* Hu et al. (2026b)	12.5×8	1100	2.62	1.46	4.13	4.08	5.13	4.12	5.81	91.3
OmniCodec* Hu et al. (2026b)	12.5×6	825	2.69	1.52	4.07	4.01	5.03	4.14	5.72	88.5
OmniCodec* Hu et al. (2026b)	12.5×5	687	2.74	1.56	4.03	3.95	4.95	4.15	5.65	86.1
OmniCodec* Hu et al. (2026b)	12.5×4	550	2.82	1.63	3.97	3.87	4.85	4.17	5.57	83.0
OmniCodec* Hu et al. (2026b)	12.5×2	275	3.19	2.01	3.81	3.34	4.45	3.79	5.51	75.2
MOSS* Gong et al. (2026)	12.5×32	4000	2.11	1.19	4.30	4.45	5.57	4.09	6.19	97.5
MOSS* Gong et al. (2026)	12.5×8	1000	2.32	1.39	4.19	4.34	5.45	4.00	6.07	93.8
MOSS* Gong et al. (2026)	12.5×6	750	2.38	1.45	4.15	4.31	5.40	3.98	6.02	92.1
MOSS* Gong et al. (2026)	12.5×4	500	2.48	1.55	4.10	4.24	5.32	3.95	5.93	89.1
MOSS* Gong et al. (2026)	12.5×2	250	2.68	1.74	3.99	4.08	5.16	3.96	5.74	83.4
MOSS* Gong et al. (2026)	12.5×1	125	2.90	1.97	3.84	3.93	5.18	4.08	5.66	79.3
HiggsV2* Boson AI (2025)	25×8	2000	2.21	1.28	4.30	4.36	5.42	4.15	6.04	96.2
HiggsV2* Boson AI (2025)	25×4	1000	2.35	1.40	4.20	4.27	5.29	4.20	5.98	92.7
HiggsV2* Boson AI (2025)	25×2	500	2.55	1.58	4.08	4.34	5.29	4.41	5.99	86.6
Universal Single-Codebook NACs										
WavTokenizer-40* Ji et al. (2025)	40×1	480	2.73	1.73	4.02	3.79	4.88	4.07	5.51	83.2
WavTokenizer-75* Ji et al. (2025)	75×1	900	2.63	1.58	4.08	3.21	4.28	3.47	5.23	80.2
UniCodec* Jiang et al. (2025)	75×1	1050	2.41	1.43	4.15	3.90	5.00	3.95	5.65	90.2
AUV Chen et al. (2026)	50×1	716	2.35	1.42	4.21	4.30	5.42	4.11	6.05	92.6
L3AC Zhai et al. (2025)	59.26×1	1000	2.81	2.03	4.08	4.19	5.40	4.10	5.93	86.8
L3AC Zhai et al. (2025)	44.44×1	750	2.86	2.08	4.04	4.16	5.38	4.03	5.89	84.9
Ours – Universal Single-Codebook NACs										
UFC-NQ	13×1	–	2.16	1.13	4.53	4.37	5.54	4.10	6.18	96.6
UFC-NQ, $\lambda_{\text{acoustic}} = 10.0$, $\lambda_{\text{semantic}} = 0.0$	13×1	–	2.20	1.16	4.48	4.30	5.49	4.07	6.13	96.1
UFC-NQ, $\lambda_{\text{acoustic}} = 0.0$, $\lambda_{\text{semantic}} = 0.0$	13×1	–	2.14	1.12	4.48	4.34	5.51	4.08	6.16	96.3
UFC-13-BSQ	13×1	195	2.98	1.89	3.93	4.08	5.09	3.77	5.74	79.9
UFC-52-BSQ	52×1	780	2.52	1.53	4.17	4.18	5.21	3.90	5.90	86.4
UFC-52-26-BSQ	26×1	390	2.61	1.62	4.05	4.14	5.16	3.69	5.82	82.4
UFC-52-13-BSQ	13×1	195	2.71	1.75	3.94	3.93	5.11	3.26	5.79	79.6
UFC-13-FSQ	13×1	195	2.93	1.83	3.92	4.06	5.11	3.67	5.78	79.6
UFC-52-FSQ	52×1	780	2.50	1.51	4.15	4.22	5.20	3.91	5.89	85.4
UFC-52-26-FSQ	26×1	390	2.60	1.62	4.03	4.12	5.15	3.60	5.82	81.6
UFC-52-13-FSQ	13×1	195	2.85	2.05	3.90	3.96	5.13	3.26	5.79	79.1
UFC-13-VQ	13×1	195	2.98	1.85	3.90	4.05	5.09	3.63	5.76	79.3
UFC-52-VQ	52×1	780	2.51	1.50	4.15	4.24	5.24	3.89	5.92	85.0
UFC-52-26-VQ	26×1	390	2.60	1.60	4.05	4.12	5.21	3.60	5.86	81.7
UFC-52-13-VQ	13×1	195	2.72	1.75	3.93	3.89	5.04	3.24	5.72	79.6

Table 12: Comparison of speech-only NACs and our systems on the test-clean partition of LibriSpeech (Panayotov et al., 2015). NACs marked with * operate on 24 kHz sampling rate. For them, we resample input to 24 kHz and downsample their output back to 16 kHz for a fair comparison with other 16 kHz codecs.

Codec	TPS	Codebook	BR (bps)	# Params	dWER ↓	STOI ↑	PESQ ↑	ViSQOL ↑	UTMOS ↑	SCOREQ ↓	SPK ↑
Speech Multi-Codebook NACs											
Qwen3-TTS-Tokenizer* Hu et al. (2026a)	12.5×16	2048	2200	171M	3.33	0.96	3.20	4.47	4.15	0.13	99.1
Qwen3-TTS-Tokenizer* Hu et al. (2026a)	12.5×6	2048	825	171M	6.87	0.87	1.70	3.79	3.08	0.50	94.9
Qwen3-TTS-Tokenizer* Hu et al. (2026a)	12.5×4	2048	550	171M	10.12	0.80	1.28	3.20	2.39	0.89	86.2
Qwen3-TTS-Tokenizer* Hu et al. (2026a)	12.5×2	2048	275	171M	54.68	0.71	1.12	2.45	1.45	1.28	64.9
StableCodec Parker et al. (2025)	25×2	15625	696	953M	9.81	0.91	2.06	3.57	4.31	0.31	94.1
StableCodec Parker et al. (2025)	25×1	46656	387	953M	11.04	0.89	1.92	3.50	4.31	0.32	93.3
Ours – Universal Single-Codebook NACs											
UFC-NQ	13×1	–	–	275M	2.41	0.97	3.73	4.84	3.87	0.15	99.2
UFC-13-BSQ	13×1	32768	195	362M	23.11	0.82	1.56	3.49	3.77	0.42	90.3
UFC-52-BSQ	52×1	32768	780	362M	7.95	0.90	2.53	4.10	3.90	0.28	95.7
UFC-52-26-BSQ	26×1	32768	390	362M	15.98	0.86	2.10	3.80	3.81	0.38	92.1
UFC-52-13-BSQ	13×1	32768	195	362M	52.31	0.80	1.56	3.38	3.28	0.69	83.6
UFC-13-FSQ	13×1	32768	195	362M	24.95	0.83	1.57	3.49	3.87	0.41	89.9
UFC-52-FSQ	52×1	32768	780	362M	7.88	0.90	2.50	4.06	3.88	0.29	95.2
UFC-52-26-FSQ	26×1	32768	390	362M	17.46	0.86	2.02	3.71	3.85	0.40	91.1
UFC-52-13-FSQ	13×1	32768	195	362M	71.00	0.77	1.44	3.17	2.92	0.87	78.1
UFC-13-VQ	13×1	32768	195	362M	22.06	0.82	1.59	3.50	3.95	0.41	89.9
UFC-52-VQ	52×1	32768	780	362M	7.60	0.90	2.51	4.04	3.94	0.28	95.4
UFC-52-26-VQ	26×1	32768	390	362M	13.17	0.86	2.04	3.72	3.86	0.37	91.6
UFC-52-13-VQ	13×1	32768	195	362M	50.68	0.80	1.61	3.39	3.40	0.67	83.1
Speech Single-Codebook NACs											
BigCodec Xin et al. (2024)	80×1	8192	1040	159M	4.68	0.94	2.67	4.21	4.11	0.15	98.0
XCodec-2 Ye et al. (2025b)	50×1	65536	800	630M	7.23	0.92	2.44	4.12	4.13	0.17	97.8
FocalCodec-50 Della Libera et al. (2026)	50×1	8192	650	142M	5.51	0.86	1.44	3.65	4.05	0.23	96.9
FocalCodec-25 Della Libera et al. (2026)	25×1	8192	325	144M	6.33	0.83	1.32	3.45	4.14	0.27	95.5
FocalCodec-12.5 Della Libera et al. (2026)	12.5×1	8192	163	145M	9.58	0.78	1.21	3.15	4.22	0.31	92.7
Ours – Speech Single-Codebook NACs											
UFC-13-BSQ-S	13×1	32768	195	362M	16.45	0.85	1.67	3.60	4.17	0.34	92.4
UFC-52-BSQ-S	52×1	32768	780	362M	7.43	0.90	2.60	4.14	4.08	0.24	96.4
UFC-52-26-BSQ-S	26×1	32768	390	362M	13.87	0.87	2.17	3.83	4.10	0.30	93.0
UFC-13-FSQ-S	13×1	32768	195	362M	17.25	0.85	1.64	3.57	4.17	0.35	91.3
UFC-52-FSQ-S	52×1	32768	780	362M	6.65	0.91	2.63	4.11	4.12	0.24	96.3
UFC-52-26-FSQ-S	26×1	32768	390	362M	10.03	0.88	2.23	3.87	4.18	0.28	93.8
UFC-13-VQ-S	13×1	32768	195	362M	27.43	0.81	1.44	3.37	4.15	0.38	89.8
UFC-52-VQ-S	52×1	32768	780	362M	8.36	0.89	2.44	4.00	4.08	0.27	95.1
UFC-52-26-VQ-S	26×1	32768	390	362M	16.38	0.85	1.97	3.68	4.13	0.33	91.6
UFC-13-SimVQ-S	13×1	32768	195	362M	55.64	0.75	1.25	3.11	3.22	0.56	86.2
UFC-52-SimVQ-S	52×1	32768	780	362M	13.31	0.87	2.17	3.90	3.99	0.30	94.2
UFC-52-26-SimVQ-S	26×1	32768	390	362M	43.16	0.81	1.61	3.46	3.59	0.51	87.2

Table 13: Comparison of speech-only NACs and our systems on the non-English test partition of Multilingual LibriSpeech (Pratap et al., 2020). NACs marked with * operate on 24 kHz sampling rate. For them, we resample input to 24 kHz and downsample their output back to 16 kHz for a fair comparison with other 16 kHz codecs.

Codec	TPS	Codebook	BR (bps)	# Params	dWER ↓	STOI ↑	PESQ ↑	ViSQOL ↑	UTMOS ↑	SCOREQ ↓	SPK ↑
Speech Multi-Codebook NACs											
Qwen3-TTS-Tokenizer* Hu et al. (2026a)	12.5×16	2048	2200	171M	3.84	0.95	3.00	4.46	2.93	0.14	99.6
Qwen3-TTS-Tokenizer* Hu et al. (2026a)	12.5×6	2048	825	171M	10.80	0.86	1.64	3.78	1.96	0.57	97.1
Qwen3-TTS-Tokenizer* Hu et al. (2026a)	12.5×4	2048	550	171M	20.93	0.79	1.26	3.18	1.58	0.93	91.0
Qwen3-TTS-Tokenizer* Hu et al. (2026a)	12.5×2	2048	275	171M	134.25	0.69	1.13	2.42	1.30	1.20	80.7
StableCodec Parker et al. (2025)	25×2	15625	696	953M	43.64	0.87	1.86	3.39	3.47	0.38	96.0
StableCodec Parker et al. (2025)	25×1	46656	387	953M	65.71	0.85	1.70	3.29	3.47	0.39	95.3
Ours – Universal Single-Codebook NACs											
UFC-NQ	13×1	–	–	275M	2.52	0.92	3.64	4.82	2.57	0.15	99.6
UFC-13-BSQ	13×1	32768	195	362M	52.56	0.80	1.57	3.52	2.73	0.39	94.7
UFC-52-BSQ	52×1	32768	780	362M	14.54	0.85	2.43	4.05	2.74	0.28	98.2
UFC-52-26-BSQ	26×1	32768	390	362M	33.68	0.82	2.03	3.75	2.79	0.35	96.1
UFC-52-13-BSQ	13×1	32768	195	362M	119.07	0.76	1.56	3.37	2.54	0.56	92.5
UFC-13-FSQ	13×1	32768	195	362M	58.19	0.80	1.56	3.50	2.86	0.40	93.9
UFC-52-FSQ	52×1	32768	780	362M	14.16	0.86	2.41	4.00	2.76	0.28	98.0
UFC-52-26-FSQ	26×1	32768	390	362M	37.41	0.82	1.95	3.67	2.91	0.36	95.5
UFC-52-13-FSQ	13×1	32768	195	362M	165.55	0.74	1.49	3.25	2.39	0.67	90.7
UFC-13-VQ	13×1	32768	195	362M	54.66	0.80	1.56	3.51	3.02	0.38	94.3
UFC-52-VQ	52×1	32768	780	362M	14.59	0.86	2.40	3.98	2.84	0.28	97.9
UFC-52-26-VQ	26×1	32768	390	362M	29.91	0.82	2.01	3.71	2.93	0.34	96.3
UFC-52-13-VQ	13×1	32768	195	362M	128.41	0.76	1.58	3.37	2.72	0.53	92.1
Speech Single-Codebook NACs											
BigCodec Xin et al. (2024)	80×1	8192	1040	159M	7.61	0.92	2.43	4.16	2.88	0.19	99.0
XCodec-2 Ye et al. (2025b)	50×1	65536	800	630M	7.01	0.90	2.33	4.11	3.03	0.20	99.0
FocalCodec-50 Della Libera et al. (2026)	50×1	8192	650	142M	10.04	0.85	1.46	3.63	2.95	0.28	98.4
FocalCodec-25 Della Libera et al. (2026)	25×1	8192	325	144M	12.52	0.81	1.32	3.39	3.16	0.33	97.5
FocalCodec-12.5 Della Libera et al. (2026)	12.5×1	8192	163	145M	46.86	0.74	1.19	3.01	3.40	0.40	95.1
Ours – Speech Single-Codebook NACs											
UFC-13-BSQ-S	13×1	32768	195	362M	81.70	0.81	1.57	3.50	3.17	0.40	94.2
UFC-52-BSQ-S	52×1	32768	780	362M	15.86	0.86	2.42	4.06	2.95	0.26	98.1
UFC-52-26-BSQ-S	26×1	32768	390	362M	46.75	0.82	2.00	3.72	3.06	0.35	95.6
UFC-13-FSQ-S	13×1	32768	195	362M	93.74	0.80	1.53	3.45	3.20	0.41	93.2
UFC-52-FSQ-S	52×1	32768	780	362M	13.08	0.86	2.46	4.03	2.98	0.28	98.0
UFC-52-26-FSQ-S	26×1	32768	390	362M	34.30	0.83	2.05	3.75	3.19	0.34	95.9
UFC-13-VQ-S	13×1	32768	195	362M	132.41	0.76	1.37	3.24	3.27	0.45	91.6
UFC-52-VQ-S	52×1	32768	780	362M	20.78	0.84	2.26	3.90	3.01	0.32	97.0
UFC-52-26-VQ-S	26×1	32768	390	362M	78.71	0.80	1.79	3.54	3.26	0.40	93.9
UFC-13-SimVQ-S	13×1	32768	195	362M	153.60	0.72	1.23	3.02	2.35	0.56	90.8
UFC-52-SimVQ-S	52×1	32768	780	362M	41.09	0.82	2.02	3.80	2.90	0.33	96.6
UFC-52-26-SimVQ-S	26×1	32768	390	362M	139.61	0.76	1.52	3.34	2.69	0.50	92.9