Exploring Disentangled Neural Speech Codecs from Self-Supervised Representations

Aihara, Ryo; Masuyama, Yoshiki; Germain, François G; Wichern, Gordon; Le Roux, Jonathan

TR2026-035 March 19, 2026

Abstract

Neural audio codecs (NACs), which use neural net- works to generate compact audio representations, have garnered interest for their applicability to many downstream tasks— especially quantized codecs due to their compatibility with large language models. However, unlike text, speech conveys not only linguistic content but also rich paralinguistic features. Encoding these elements in an entangled fashion may be suboptimal, as it limits flexibility. For instance, voice conversion (VC) aims to convert speaker characteristics while preserving the original linguistic content, which requires a disentangled representation. Inspired by VC methods utilizing k-means quantization with self-supervised features to disentangle phonetic information, we develop a discrete NAC capable of structured disentanglement. Experimental evaluations show that our approach achieves re- construction performance on par with conventional NACs that do not explicitly perform disentanglement, while also matching the effectiveness of conventional VC techniques.

IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) 2026

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Exploring Disentangled Neural Speech Codecs from Self-Supervised Representations

Ryo Aihara*†, Yoshiki Masuyama†, Gordon Wichern†, François G. Germain†, Jonathan Le Roux†

**Mitsubishi Electric Corporation*, Tokyo, Japan,
Aihara.Ryo@dx.MitsubishiElectric.co.jp
†*Mitsubishi Electric Research Laboratories (MERL)*, MA, USA,
{masuyama, germain, wichern, leroux}@merl.com

***Abstract*—Neural audio codecs (NACs), which use neural networks to generate compact audio representations, have garnered interest for their applicability to many downstream tasks—especially quantized codecs due to their compatibility with large language models. However, unlike text, speech conveys not only linguistic content but also rich paralinguistic features. Encoding these elements in an entangled fashion may be suboptimal, as it limits flexibility. For instance, voice conversion (VC) aims to convert speaker characteristics while preserving the original linguistic content, which requires a disentangled representation. Inspired by VC methods utilizing k-means quantization with self-supervised features to disentangle phonetic information, we develop a discrete NAC capable of structured disentanglement. Experimental evaluations show that our approach achieves reconstruction performance on par with conventional NACs that do not explicitly perform disentanglement, while also matching the effectiveness of conventional VC techniques.**

***Index Terms*—neural speech codec, voice conversion, quantization, disentanglement, self-supervised learning**

I. Introduction

The recent success of large language models (LLMs) has renewed interest in neural audio codecs (NACs) within the speech and audio research community. NACs have found use not only as tokenizers for audio and speech language models such as AudioLM [1], but also to produce tokens as outputs in audio-based dialogue systems [2]. NACs exist in both continuous and discrete flavors, and which one is better is an open research question. While continuous representations offer strong reconstruction capabilities [3], discrete audio tokens are often preferred due to their compatibility with LLMs, which operate on discrete text tokens [2].

End-to-end discrete NACs have been extensively studied. They typically leverage a generative adversarial network (GAN) framework to train a convolutional encoder for processing audio waveforms, a quantizer employing residual vector quantization (RVQ), and a deconvolutional decoder for waveform reconstruction [4]–[6].

Another key direction in the development of NACs is disentanglement. Several studies have leveraged distillation from self-supervised learning (SSL) models or integrated SSL to capture "semantic" information, leading to improved ASR performance with these tokens [7]–[9]. Considering ASR and other potential downstream tasks for speech codecs, it is desirable to develop disentangled codecs.

Speech information can be categorized into three types: linguistic (lexical, semantic), paralinguistic (intentional, stylistic), and non-linguistic (physical, emotional) [10]. This suggests that a speech codec be disentangled into these three categories to more effectively capture the distinct aspects of speech. Voice conversion (VC), which aims to modify speaker characteristics while preserving the original linguistic content, also necessitates a disentangled representation. In recent years, SSL has been increasingly employed to enhance this process. In [11], it was shown that the middle layer of WavLM [12] is strongly correlated with speaker identity while retaining prosody and phonetic information, enabling speaker VC by swapping the corresponding WavLM representations between two speakers. This approach has also been applied to text-to-speech (TTS) systems [13]. Similarly, Sim et al. [14] proposed a one-shot VC method that also leverages the middle layer of WavLM, employing k-means quantization.

Inspired by these methods, this study explores a speech codec that disentangles phonetic, prosodic, and speaker characteristics, which correspond to linguistic, paralinguistic, and non-linguistic information. To evaluate the proposed approach, we conduct both reconstruction and one-shot VC experiments. The results are compared with a non-disentangled codec and conventional VC methods.

The contributions of this paper can be summarized as follows. First, we explore a fully discrete speech codec that disentangles phonetic, prosodic, and speaker information, without any supervision from phonetic labels or fundamental frequency (F_0). We also demonstrate that our disentangled codec achieves one-shot voice conversion performance comparable to conventional methods, using only a single utterance from the target speaker. Experimental results indicate that discretizing speaker vectors tends to diminish speaker-specific information, requiring a sufficiently rich discrete representation to preserve speaker identity.

This approach is valuable on two levels: it enhances interpretability while also enabling disentangled control of a representation compatible with downstream applications like audio and speech language models (LMs).

TABLE I: Comparison of related works on disentangled speech representations

Methods	**Quantize phoneme**	**Quantize prosody**	**Quantize speaker**	**No external label**
Phonetic Codecs [2], [7]–[9], [11], [14]–[20]	✓			✓
Speech Resynthesis [21]	✓	✓		
Phoneme-level Speech Codec [22]	✓	✓		
FACodec [23]	✓	✓	✓	
Ours	✓	✓	✓	✓

II. Related works

Neural audio codecs — SoundStream [4] leveraged a GAN framework to train a convolutional encoder for processing audio waveforms, RVQ, and a deconvolutional decoder for waveform reconstruction. EnCodec [5] extended SoundStream by incorporating a multi-scale discriminator and a loss-balancing mechanism to improve training stability. DAC [6] refined RVQ through L^2-normalization and dimensionality reduction, serving as a strong baseline for subsequent advancements in neural audio coding. SNAC [24] extended DAC with muti-scale RVQ, leading to even more efficient compression. We employ DAC and SNAC as our baseline non-disentangled NACs.

Disentangled speech representations — We summarized the comparison of some related works in Table I. Mimi [2] and SpeechTokenizer [7] introduced knowledge distillation from WavLM to capture phonetic information in codec feature space. Repcodec [8] encodes and decodes directly from the SSL model, and X-Codec [9] concatenates a phonetic codec with an acoustic codec, resulting in enhanced ASR performance. However, the disentanglement of prosodic and speaker information is not investigated in these papers. Polyak et al. [21] proposed a speech resynthesis method where phonetic, prosodic, and speaker information are estimated in parallel. However, their approach utilizes the estimated F_0 from a vocoder as the target for prosodic quantization, which differs from our method. It also does not quantize the speaker vector. We assume that quantized speaker information is likely to be more compatible with downstream applications like speech LM. Ju et al. [23] proposed FACodec, which estimates three streams in parallel for TTS, but their method requires phoneme labels for training, whereas our approach does not impose this constraint. Additionally, they employ a gradient reversal layer (GRL) for disentanglement, which is known to be challenging to train effectively [25]. Guo et al. [15] applied speaker disentanglement in the codec feature space but their method relies on the the waveform similarity overlap-add (WSOLA) technique, which imposes significant computational costs. Similarly, Qian et al. [16] introduced a speaker disentanglement method using contrastive learning between speakers, but their architecture does not incorporate quantization of prosodic information. Ren et al. [17] proposed a model that estimates a time-invariant code, and Della Libera et al. [18] introduced a NAC with a single codebook, which also disentangles speaker information. However, both approaches do not explicitly disentangle prosodic features from phonetic representations. Karapiperis et al. [22] explored prosodic quantization, but their approach relies on externally provided phoneme sequences and speaker vectors.

Voice conversion — VQ is utilized for VC to disentangle speaker information from other information [19]. SSL is also applied to VC to disentangle speaker information from phonetic information. Baas et al. [11] proposed a simple but effective VC method using WavLM, however, the conversion quality degrades when the number of target speaker utterances is small during inference. Kanagawa et al. [20] proposed to perform VC using discrete units from SSL, however prosodic and speaker information is not discretized. SKQVC [14] also uses discrete units from WavLM and we employ this method as our baseline, where only phonetic information is discretized.

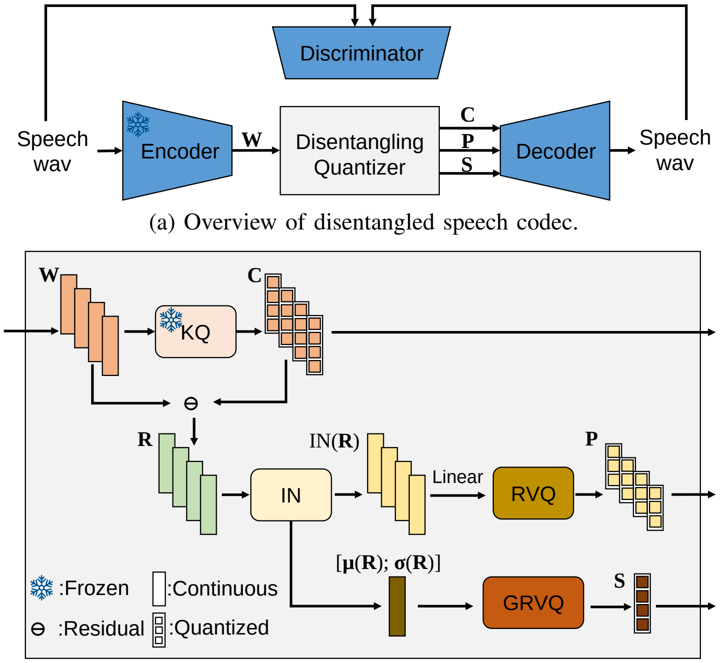
III. Disentangled neural speech codec

A. Overview

Figure 1 depicts the model architecture for our proposed disentangled speech codec, which is comprised of an encoder, disentangling quantizer, decoder, and discriminator. Here, the key distinction from conventional SKQVC lies in the use of a disentangling quantizer. We use WavLM-Large as the encoder, outputting its 6th-layer hidden vectors as representation. We denote them as $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T\} \in \mathbb{R}^{D \times T}$, where D denotes the number of dimensions and T denotes the number of frames. We use a pre-trained WavLM [12] and its parameters are frozen during the training of the codec.

The disentangling quantizer converts $\mathbf{W}$ into three components. As shown in Fig. 1, $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_T\} \in \mathbb{R}^{D \times T}$, $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T\} \in \mathbb{R}^{F \times T}$, and $\mathbf{S} \in \mathbb{R}^{2D}$ represent the quantized content vectors, quantized prosody vectors, and quantized speaker vector, respectively. F denotes the dimension of the quantized prosody vectors. The decoder combines these components to reconstruct the speech waveform. Note that the decoder of SKQVC accepts these three components but only the phonetic components are quantized while the other vectors remain continuous.

The discriminator consists of a multi-period discriminator [26] and a complex multi-scale short-time Fourier transform (STFT) discriminator [27]. The entire model is trained in a generative adversarial manner, as outlined in [6].



(a) Overview of disentangled speech codec.

(b) Disentangling quantizer.

Fig. 1: Model architecture of disentangled speech codec.

B. Disentangling Quantizer

The quantized content vectors $\mathbf{C}$ are estimated from the representation $\mathbf{W}$ using VQ as follows:

$$\mathbf{c}_t = \mathbf{q}_{\hat{j}_t}, \quad \hat{j}_t = \arg\min_j \|\mathbf{w}_t - \mathbf{q}_j\|, \forall t, \tag{1}$$

where $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_J\} \in \mathbb{R}^{D \times J}$ represents the codebook vectors and J denotes the number of codes. We denote this operation as KQ in Fig. 1. We compute $\mathbf{Q}$ prior to training the quantizer, as the result of applying k-means clustering to the representations $\mathbf{W}$ obtained from a separate large dataset [28]. We then freeze $\mathbf{Q}$ for the rest of the training.

Following the content quantization module KQ, we compute the residual $\mathbf{R} = \mathbf{W} - \mathbf{C} \in \mathbb{R}^{D \times T}$. Since the 6th layer of WavLM-Large has been shown to exhibit a high correlation with speaker identity and prosody compared to other layers [11], and the centroids of k-means of SSL hidden features are commonly assumed to capture speaker-independent phonetic information [29], the residual is likely to preserve substantial speaker-related and prosodic information. Next, instance normalization (IN) is applied to $\mathbf{R}$ as follows:

$$\mathrm{IN}(\mathbf{R}) = \frac{\mathbf{R} - \boldsymbol{\mu}(\boldsymbol{R})}{\boldsymbol{\sigma}(\boldsymbol{R})}, \tag{2}$$

where $\boldsymbol{\mu}(\boldsymbol{R}) \in \mathbb{R}^D$ and $\boldsymbol{\sigma}(\boldsymbol{R}) \in \mathbb{R}^D$ denote the mean and standard deviation of $\mathbf{R}$ along the time axis. As a result, $\boldsymbol{\mu}(\boldsymbol{R})$ and $\boldsymbol{\sigma}(\boldsymbol{R})$ are time-invariant, and their concatenated vector is treated as a pre-quantization speaker vector [30]. They are broadcasted appropriately to match the dimension of $\mathbf{R}$ in (2).

Time-variant vectors $\mathrm{IN}(\mathbf{R}) \in \mathbb{R}^{D \times T}$ are then assumed to contain prosodic information. Because an information bottleneck is required to minimize potential interference with phonetic content, these vectors are projected to an F-dimensional space by a linear transformation, and quantized prosody vectors $\mathbf{P}$ are estimated by RVQ. The prosodic quantization codebooks are estimated during training in the L^2-normalized feature space.

Finally, we use group-residual vector quantization (GRVQ) [31] to quantize the pre-quantization speaker vector, and obtain the quantized speaker vector $\mathbf{S}$. In GRVQ, the input feature space is partitioned into several groups along the feature dimension axis, with RVQ performed on each group. During the RVQ process for each group within GRVQ, codes are estimated in the L^2-normalized feature space with the low-dimensional code lookup process as described in [6].

C. Decoder

The quantized content vectors $\mathbf{C}$, the quantized prosody vectors $\mathbf{P}$, and the quantized speaker vector $\mathbf{S}$ are passed through the decoder. The quantized content vectors and the quantized prosody vectors are concatenated along the dimension axis and modulated by a feature-wise linear modulation (FiLM) [32] conditioned on the quantized speaker vector as follows:

$$\mathrm{FiLM}([\mathbf{C}; \mathbf{P}] | \mathbf{S}) = f(\mathbf{S}) \odot [\mathbf{C}; \mathbf{P}] + h(\mathbf{S}), \tag{3}$$

where f and h can be trainable arbitrary functions and $[\cdot;\cdot]$ and $\odot$ denote the concatenation operation along the dimension axis and the element-wise product, respectively. In this paper, we employ a simple linear transformation for both functions.

The remainder of the decoder consists of up-sampling deconvolutional blocks that up-sample the modulated features to reconstruct the waveform. Following the approach presented in [24], depth-wise deconvolution is applied to all deconvolutional layers, and a noise block is inserted after each up-sampling layer.

IV. Experiments

A. Experimental conditions

We compare the proposed method with the following unsupervised methods:

- **Vanilla RVQ**: A low bit-rate variant of DAC [6]. The encoder consists of downsampling convolutional blocks with strides, where the downsampling factors are set to (2, 4, 5, 8), resulting in a token rate of 50 Hz for 16 kHz audio, which matches the sampling frame rate used in WavLM. Following the encoder, a two-layer RVQ module with each codebook of size 4096 is applied. Unlike [6], depth-wise convolution is incorporated into each convolutional layer, which is essential for stable training in low bit-rate scenarios [24].
- **SNAC** [24]: This model employs the same encoder architecture as Vanilla RVQ but utilizes a three-layer RVQ with each codebook of size 4096 and strides of (4, 2, 1).
- **SKQ** [14]: The starting point for this study, it is the same as the architecture described in Section III, except for the following. The operation IN is simplified to only estimating and subtracting the mean $\boldsymbol{\mu}(\boldsymbol{R})$. Also, neither $\boldsymbol{\mu}(\boldsymbol{R})$, nor the projection of $\mathrm{IN}(\mathbf{R})$ to an 8-dimensional feature space are quantized by GRVQ and RVQ, respectively, meaning the speaker and prosody information remain continuous. Finally, for decoding, the FiLM operation is replaced by adding $\boldsymbol{\mu}(\boldsymbol{R})$ element-wise to the concatenated content and prosody vectors. Equivalently, this corresponds to $\mathbf{S} = \boldsymbol{\mu}(\boldsymbol{R})$, $f(\mathbf{S}) = 1$ and $h(\mathbf{S}) = \mathbf{S}$ in (3).
- **SKQ+σ**: Same as the architecture described in Section III, except that neither $\boldsymbol{\mu}(\boldsymbol{R})$ and $\boldsymbol{\sigma}(\boldsymbol{R})$ nor the projection of $\mathrm{IN}(\mathbf{R})$ to an 8-dimensional feature space are quantized by GRVQ and RVQ, respectively, meaning the speaker and prosody information remain continuous.
- **SKQ2+σ**: Same as the architecture described in Section III, except that $\boldsymbol{\mu}(\boldsymbol{R})$ and $\boldsymbol{\sigma}(\boldsymbol{R})$ are not quantized by GRVQ, meaning the speaker information remains continuous. $\mathrm{IN}(\mathbf{R})$ is projected to an 8-dimensional feature space before being quantized by a two-layer RVQ, where each VQ layer has a codebook size of 1000.
- **SKQ3+σ**: The architecture described in Section III. $\mathrm{IN}(\mathbf{R})$ is projected to an 8-dimensional feature space before being quantized by a two-layer RVQ, where each VQ layer has a codebook size of 1000. $\boldsymbol{\mu}(\boldsymbol{R})$ and $\boldsymbol{\sigma}(\boldsymbol{R})$ are quantized by GRVQ, where codebook size for each GRVQ layer is fixed at 1024. Other GRVQ hyperparameters are discussed in the ablation study of Section IV-B.
- **kNN-VC** [11]: The 6th layer of WavLM is used as a feature extractor. kNN regression is performed between the source and target utterances, replacing the matched source features with those from the target.

With the exception of kNN-VC, all models are trained on the publicly available LibriSpeech-train-clean-100 dataset and validated on its corresponding validation set, using a sampling rate of 16 kHz. We train with a batch size of 32 using excerpts of duration 3 seconds, and validate using excerpts of duration 5 seconds. The AdamW optimizer is employed with a learning rate of 6×10^{-4}, $\beta_1 = 0.8$, and $\beta_2 = 0.9$. The learning rate is decayed at each step with a decay factor of $\gamma = 0.999994$. The weights for the multiple loss functions are identical to [6], [24], and layer dropout is not applied for RVQ and GRVQ.

The pre-trained WavLM-Large model is obtained from its official repository[1], and extracts 1024-dimensional features at 50 Hz for 16 kHz audio from its 6th layer. Codebooks[2] for WavLM-Large are constructed using mini-batch k-means clustering over the full training subset of the LibriSpeech dataset, with the number of centroids fixed at 1000.

To ensure a fair comparison, all models except kNN-VC employ the same decoder architecture, introduced at the end of Section III-C. The upsampling factors of the decoder's upsampling convolutional blocks are set to (8, 5, 4, 2). The discriminator follows the architecture proposed in DAC [6]. For kNN-VC, the decoder utilizes HiFi-GAN [26], which is trained to estimate waveforms from the WavLM-Large 6th-layer representations[3].

We conduct two types of evaluation, both performed on the LibriSpeech-test-clean subset. For reconstruction evaluation, we employ the following objective metrics: mel-spectrum distance (mel-d), STFT distance (STFT-d), and ViSQOL[4] [33] for sound quality, and equal error rate (EER) for speaker verification (SV). For SV, the speaker embedding is estimated by ECAPA-TDNN [34], implemented using the NeMo toolkit[5] [35]. For one-shot VC evaluation, we randomly select two reference utterances: one spoken by a male speaker and the other by a female speaker, and speaker identity is modified by swapping the quantized speaker vector with that of a reference utterance. We assess the word error rate (WER) of the converted speech, EER of SV, UTMOSv2 scores[6] [36], and the Pearson correlation coefficient (PCC) of F_0 between the F_0 contours of the source and converted speech. WER is calculated using ASR with Whisper-turbo[7] [37]. For F_0-PCC, F_0 estimates are obtained using DIO [38] and StoneMask [39] from the WORLD vocoder[8].

[1] https://huggingface.co/microsoft/wavlm-large

[2] https://huggingface.co/speechbrain/SSL_Quantization

[3] https://github.com/bshall/knn-vc

[4] https://github.com/google/visqol

[5] https://github.com/NVIDIA/NeMo

[6] https://github.com/sarulab-speech/UTMOSv2

[7] https://github.com/openai/whisper

[8] https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder

TABLE II: Reconstruction results for each method. Phonetic/prosodic/speaker indicate how each information is represented.

Model	phonetic	prosodic	speaker	Mel-d ↓	STFT-d ↓	ViSQOL ↑	EER [%] ↓
Vanilla RVQ	1.20 kb/s		-	0.82	1.61	4.01	3.01
SNAC	1.05 kb/s		-	0.81	1.60	4.01	2.93
SKQ	0.50 kb/s	1024 dim, 50Hz	1024 dim/u	1.01	1.84	3.97	2.36
SKQ+σ	0.50 kb/s	1024 dim, 50Hz	2048 dim/u	1.04	1.88	3.93	2.48
SKQ2+σ	0.50 kb/s	1.00 kb/s	2048 dim/u	1.06	1.89	3.87	2.55
SKQ3+σ	0.50 kb/s	1.00 kb/s	1.28 kb/u	1.05	1.88	3.88	2.65

TABLE III: One-shot VC results for each method. “c” and “q” indicate continuous and quantized vectors, respectively.

Model	prosodic	speaker	WER [%] ↓	EER [%] ↓	UTMOSv2 ↑	F_0-PCC ↑
kNN-VC	-	-	23.65	3.93	2.24	0.66
SKQ	c	c	2.25	6.88	3.03	0.77
SKQ+σ	c	c	2.57	6.15	2.92	0.76
SKQ2+σ	q	c	2.75	5.61	2.96	0.76
SKQ3+σ	q	q	2.62	5.62	2.92	0.76
Ground truth	-	-	1.92	1.13	3.23	-

TABLE IV: Ablation study on speaker vector quantization for one-shot VC.

Model	#group	#RVQ layer	WER [%] ↓	EER [%] ↓	UTMOSv2 ↑	F_0-PCC ↑
SKQ2+σ	-	-	2.75	5.61	2.96	0.76
SKQ3+σ	4	4	2.61	12.13	3.02	0.76
SKQ3+σ	8	8	2.45	9.68	3.07	0.76
SKQ3+σ	16	8	2.62	5.62	2.92	0.76
SKQ3+σ	16	16	2.58	7.47	2.99	0.76

B. Experimental results

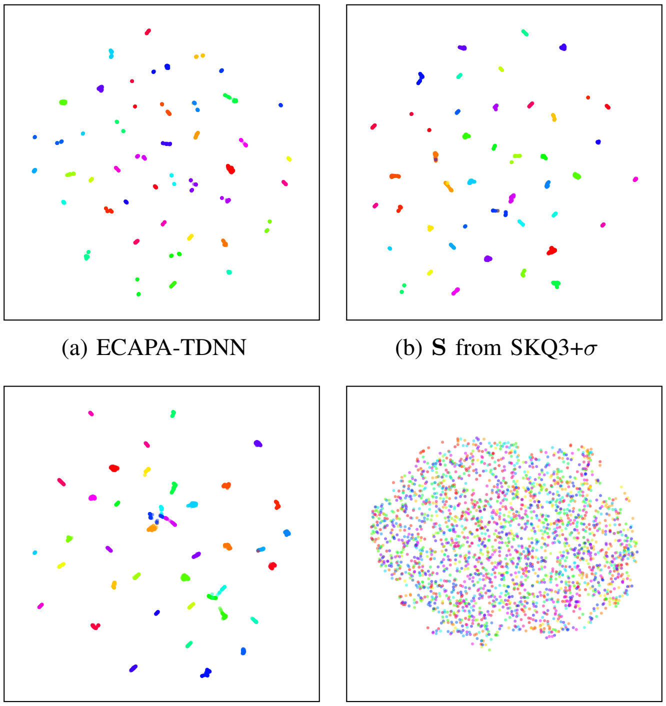
Table II presents the reconstruction evaluation results, where the values for phonetic, prosodic, and speaker indicate the amount of data associated with each (continuous or quantized) vector representing the corresponding information, and kb/s, kb/u, and dim/u stand for kilobits per second, kilobits per utterance (for quantized vectors), and dimension per utterance (for continuous vectors), respectively. The non-disentangled methods, Vanilla RVQ and SNAC, achieve slightly higher reconstruction quality than disentangled methods. However, disentangled methods demonstrate superior SV accuracy. Quantizing both prosody and speaker vectors results in a slight degradation of SV and ViSQOL performance, indicating some loss of information when converting to an LLM-compatible discrete representation.

Table III presents the one-shot VC evaluation results, where “c” and “q” indicate continuous and quantized vectors, respectively. The conventional kNN-VC achieved the highest SV accuracy but performed the worst across other evaluation metrics. This outcome is likely due to the converted speech being generated solely from the 6th-layer representation of WavLM for the target speaker. Incorporating the variance component of the speaker vector improves SV accuracy. Quantizing the prosody vectors further enhances SV performance, presumably by introducing an information bottleneck that facilitates the disentanglement of prosodic and phonetic information. However, quantizing speaker vectors slightly degrades SV performance, posing a trade-off in establishing compatibility with LLMs, while other evaluation metrics remain unaffected.

Table IV presents the results of the ablation study on speaker vector quantization in one-shot VC, where “#group” and “#RVQ layer” denote the number of groups and the number of RVQ layers in the GRVQ block of Fig. 1(b), respectively. These results suggest that increasing the bitrate for speaker vectors may be advantageous for improving SV accuracy without substantially affecting all the other VC performance indicators.

Figure 2 presents the UMAPs[9] of (a) source speaker embeddings (obtained from ECAPA-TDNN), (b) the quantized speaker vectors $\mathbf{S}$ (from SKQ3+σ), (c) $\mathbf{W}$ (from WavLM), and (d) $\mathbf{P}$ (from SKQ3+σ), respectively. Here, SKQ3+σ has a GRVQ with 16 groups and 8 codebooks. We average $\mathbf{W}$ and $\mathbf{P}$ over time for each utterance. Each color represents a distinct speaker. The projections indicate that both the quantized speaker vectors and $\mathbf{W}$ from WavLM form clusters corresponding to the identity of the utterance speaker, consistent with the behavior of speaker verification models

[9] https://github.com/lmcinnes/umap



(a) ECAPA-TDNN

(b) $\mathbf{S}$ from SKQ3+σ

(c) $\mathbf{W}$ from WavLM

(d) $\mathbf{P}$ from SKQ3+σ

Fig. 2: UMAPs for LibriSpeech utterance. Colors represent distinct speakers.

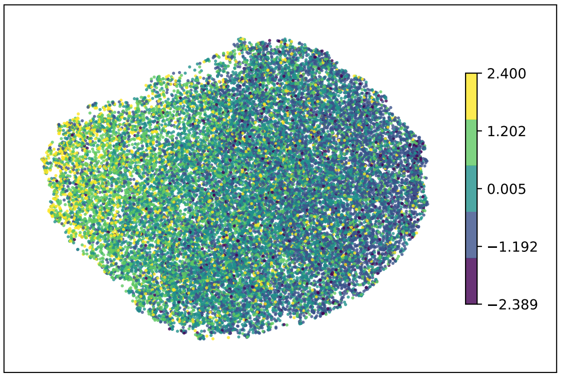


Fig. 3: UMAP of $\mathbf{P}$ from SKQ3+σ. Colors represent F_0 normalized by each speaker's mean and variance.

like ECAPA-TDNN. In contrast, $\mathbf{P}$ does not exhibit distinct speaker clusters, demonstrating the feature disentanglement between S and P in our proposed model.

Figure 3 shows the UMAP of $\mathbf{P}$ from SKQ3+σ, where the color represents F_0 of each given frame normalized by the corresponding speaker's mean and variance. Here, SKQ3+σ employs a GRVQ with 16 groups and 8 codebooks. The projection indicates that quantized prosody vectors align with F_0 deviation, with points corresponding to similar amounts of frequency deviation staying together.

To illustrate the controllability of prosody in SKQ3+σ, we select a sample from the test set and resynthesize it by replacing the prosody codes $\mathbf{P}$ in the latter half of the utterance with a fixed value equal to the code at the midpoint, resulting in a flattened prosody while preserving some variation. The spectrogram of the generated speech and the F_0 contours of the original and generated speech are shown in Fig. 4, where the prosody in the latter half is flattened, maintaining both naturalness and speaker characteristics. The ability of SKQ3+σ to perform this adjustment on quantized embeddings enables fine-grained F_0 (i.e., disentangled prosodic) control in models that require discrete codes, i.e., LLM-compatible models [2].

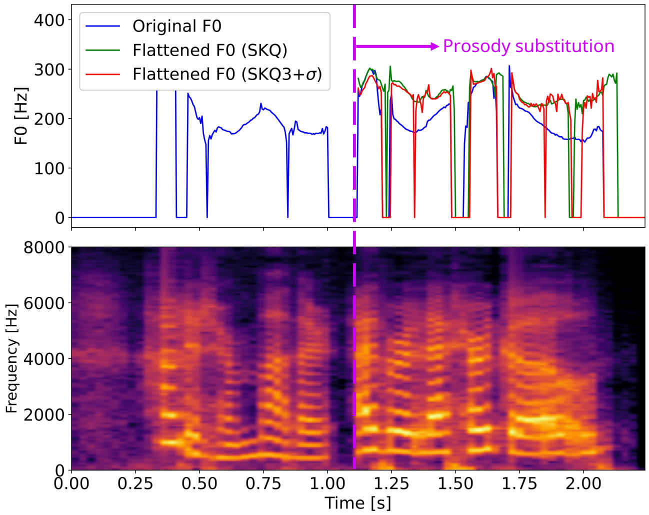


Fig. 4: F_0 contours of the original and flattened samples, along with the F_0-flattened spectrogram (SKQ3+σ).

V. Conclusion and Future Works

This paper explored a neural speech codec designed to proactively disentangle phonetic, prosodic, and speaker characteristics without supervision from textual or speaker information, leveraging the strong correlation between speaker identity and the middle layers of an SSL model. Unlike conventional VC methods, not only the phonetic feature but also prosodic and speaker features are also quantized, which facilitates the integration of these features into LLMs. Experimental results demonstrate that the proposed disentangled speech codec does effectively generate disentangled representations while preserving reconstruction quality comparable to its non-disentangled counterpart, and achieving performance on par with conventional VC methods. An ablation study further indicates that expressivity of the quantized representation of the speaker vector impacts speaker identity control, underscoring the necessity of allocating sufficient bitrate for sufficient speaker representation.

In future work, we will test the learned disentangled representations in relevant downstream tasks (e.g., ASR, TTS). We will also explore representing speech with discrete codec tokens in speech LMs, enabling the prediction of phonetic and prosodic sequences together with a fixed speaker code to maintain speaker consistency in duplex dialogue systems.

References

[1] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "AudioLM: a language modeling approach to audio generation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 2523–2533, 2023.

[2] A. Défossez, L. Mazaré, M. Orsini, A. Royer, P. Pérez, H. Jégou, E. Grave, and N. Zeghidour, "Moshi: A speech-text foundation model for real-time dialogue," *arXiv preprint arXiv:2410.00037*, 2024.

[3] P. Mousavi, L. Della Libera, J. Duret, A. Ploujnikov, C. Subakan, and M. Ravanelli, "DASB - discrete audio and speech benchmark," *arXiv preprint arXiv:2406.14294*, 2024.

[4] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An end-to-end neural audio codec," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 495–507, 2021.

[5] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *TMLR*, 2023.

[6] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved RVQGAN," in *Proc. NeurIPS*, 2023.

[7] X. Zhang, D. Zhang, S. Li, Y. Zhou, and X. Qiu, "SpeechTokenizer: Unified speech tokenizer for speech large language models," in *Proc. ICLR*, 2024.

[8] Z. Huang, C. Meng, and T. Ko, "RepCodec: A speech representation codec for speech tokenization," in *Proc. ACL*, 2024.

[9] Z. Ye, P. Sun, J. Lei, H. Lin, X. Tan, Z. Dai, Q. Kong, J. Chen, J. Pan, Q. Liu, Y. Guo, and W. Xue, "Codec does matter: Exploring the semantic shortcoming of codec for audio language model," *arXiv preprint arXiv:2408.17175*, 2024.

[10] H. Fujisaki, "Prosody, information, and modeling — with emphasis on tonal features of speech —," in *Proc. Interspeech*, 2003.

[11] M. Baas, B. van Niekerk, and H. Kamper, "Voice conversion with just nearest neighbors," in *Proc. Interspeech*, 2023.

[12] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 6, pp. 1505—1518, 2022.

[13] K. E. Hajal, A. Kulkarni, E. Hermann, and M. Magimai-Doss, "kNN retrieval for simple and effective zero-shot multi-speaker text-to-speech," in *Proc. NAACL*, 2025.

[14] Y. Sim, J. Yoon, and Y. Suh, "SKQVC: One-shot voice conversion by k-means quantization with self-supervised speech representations," *arXiv preprint arXiv:2411.16147*, 2024.

[15] Y. Guo, Z. Li, C. Du, H. Wang, X. Chen, and K. Yu, "LSCodec: Low-bitrate and speaker-decoupled discrete speech codec," *arXiv preprint arXiv:2410.15764*, 2024.

[16] K. Qian, Y. Zhang, H. Gao, J. Ni, C. Lai, D. Cox, M. Hasegawa-Johnson, and S. Chang, "ContentVec: An improved self-supervised speech representation by disentangling speakers," *arXiv preprint arXiv:2204.09224*, 2022.

[17] Y. Ren, T. Wang, J. Yi, L. Xu, J. Tao, C. Zhang, and J. Zhou, "Fewer-token neural speech codec with time-invariant codes," in *Proc. ICASSP*, 2024.

[18] L. Della Libera, F. Paissan, C. Subakan, and M. Ravanelli, "FocalCodec: Low-bitrate speech coding via focal modulation networks," *arXiv preprint arXiv:2502.04465*, 2025.

[19] D. Wang, L. Deng, Y. Yeung, X. Chen, X. Liu, and H. Meng, "VQMIVC: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion," in *Proc. Interspeech*, 2021.

[20] H. Kanagawa and Y. Ijima, "Knowledge distillation from self-supervised representation learning model with discrete units for any-to-any streaming voice conversion," in *Proc. Interspeech*, 2024.

[21] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech resynthesis from discrete disentangled self-supervised representations," in *Proc. Interspeech*, 2021.

[22] S. Karapiperis, N. Ellinas, A. Vioni, J. Oh, G. Jho, I. Hwang, and S. Raptis, "Investigating disentanglement in a phoneme-level speech codec for prosody modeling," in *Proc. SLT*, 2024.

[23] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, Y. Liu, Y. Leng, K. Song, S. Tang, Z. Wu, T. Qin, X.-Y. Li, W. Ye, S. Zhang, J. Bian, L. He, J. Li, and S. Zhao, "NaturalSpeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," *arXiv preprint arXiv:2403.03100*, 2024.

[24] H. Siuzdak, F. Grötschla, and L. A. Lanzendörfer, "SNAC: Multi-scale neural audio codec," in *Proc. Audio Imagination: NeurIPS Workshop on AI-Driven Speech, Music, and Sound Generation*, 2024.

[25] H. Serieys and A. Cherif, "Adversarial domain adaptation without gradient reversal layer," in *Proc. ESANN*, 2020.

[26] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, 2020.

[27] W. Jang, D. Lim, J. Yoon, B. Kim, and J. Kim, "UnivNet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation," in *Proc. Interspeech*, 2021.

[28] P. Mousavi, J. Duret, S. Zaiem, L. Della Libera, A. Ploujnikov, C. Subakan, and M. Ravanelli, "How should we extract discrete audio tokens from self-supervised models?" in *Proc. Interspeech*, 2024.

[29] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451 – 3460, 2021.

[30] Y. Chen, D. Wu, T. Wu, and H. Lee, "AGAIN-VC: A one-shot voice conversion using activation guidance and adaptive instance normalization," in *Proc. ICASSP*, 2021.

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

[32] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI*, 2018.

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

[34] N. Dawalatabad, M. Ravanelli, F. Grondin, J. Thienpondt, B. Desplanques, and H. Na, "ECAPA-TDNN embeddings for speaker diarization," in *Proc. Interspeech*, 2021.

[35] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook, P. Castonguay, M. Popova, J. Huang, and J. M. Cohen, "NeMo: A toolkit for building AI applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.

[36] K. Baba, W. Nakata, Y. Saito, and H. Saruwatari, "The T05 system for the VoiceMOS challenge 2024: Transfer learning from deep image classifier to naturalness MOS prediction of high-quality synthetic speech," in *Proc. SLT*, 2024.

[37] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[38] M. Morise, H. Kawahara, and H. Katayose, "Fast and reliable F0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech," in *Proc. AES Int. Conf.*, 2009.

[39] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. Inf. & Syst.*, vol. E99.D, no. 7, pp. 1877–1884, 2016.