Interspeech 2024
1-5 September 2024, Kos, Greece

MMM: Multi-Layer Multi-Residual Multi-Stream Discrete Speech Representation from Self-supervised Learning Model

Jiatong Shi[1], *Xutai Ma*[2], *Hirofumi Inaguma*[2], *Anna Sun*[2], *Shinji Watanabe*[1]

[1] Carnegie Mellon University, [2] Meta AI

{jiatongs, swatanab}@cs.cmu.edu

Abstract

Speech discrete representation has proven effective in various downstream applications due to its superior compression rate of the waveform, fast convergence during training, and compatibility with other modalities. Discrete units extracted from self-supervised learning (SSL) models have emerged as a prominent approach for obtaining speech discrete representation. However, while discrete units have shown effectiveness compared to spectral features, they still lag behind continuous SSL representations. In this work, we propose MMM, a multi-layer multi-residual multi-stream discrete units extraction method from SSL. Specifically, we introduce iterative residual vector quantization with K-means for different layers in an SSL model to extract multi-stream speech discrete representation. Through extensive experiments in speech recognition, speech resynthesis, and text-to-speech, we demonstrate the proposed MMM can surpass or on-par with neural codec's performance under various conditions.

Index Terms: discrete speech representation, self-supervised learning, discrete speech unit

1. Introduction

Efficient representation of speech signals is fundamental for a wide array of speech processing tasks, ranging from automatic speech recognition (ASR) to text-to-speech (TTS) synthesis. Traditionally, spectral features such as linear spectrograms or mel spectrograms have been extensively used in speech processing due to their robustness and interpretability [1, 2]. However, with the advent of deep learning, there has been a paradigm shift towards utilizing neural networks as feature extractors, offering improved performance over traditional methods [3, 4].

More recently, self-supervised learning (SSL) approaches have gained prominence in speech representation learning. These methods leverage large amounts of unlabeled speech data to learn powerful representations, surpassing previous state-of-the-art results on various benchmarks [5–13]. However, continuous SSL representations often suffer from scalability issues in terms of storage, computation, and integration with other modalities [14–17]. This has led to a growing interest in discrete speech representation approaches, which offer more efficient and compact representations.

Two prominent methods that have emerged are SSL-based units and neural audio codecs. SSL-based units leverage clustering methods in an unsupervised manner to convert continuous SSL representations into discrete units, initially explored for speech resynthesis and subsequently proven effective in tasks such as speech translation (ST), ASR, TTS, and spoken language understanding [15, 16, 18–25]. However, while SSL-based units offer efficiency and effectiveness benefits, they often fall short of achieving better performance than continuous SSL representations and lack detailed acoustics for speech generation purposes [15, 23, 26].

On the other hand, neural audio codecs focus on audio resynthesis tasks, employing neural networks to learn auto-encoder architectures for discrete codec generation. A key component of recent neural codec methods is the use of residual vector quantization (RVQ) in the discretization process, resulting in multi-stream audio compression that retains subtle audio details with enhanced expressiveness [26–29]. This property has led to extensions of neural audio codecs to text-to-speech and spoken language models, demonstrating robust speech generation capabilities in zero-shot multi-speaker TTS scenarios [30–32]. However, neural codecs optimized for resynthesis tasks often lack semantic information due to their focus on streaming efficiency and short-context representation [16].

Despite their differences, limited comparative studies exist between SSL-based units and neural audio codecs under comparable conditions. Notably, SSL-based units typically operate in a single-stream setting, which offers less information capacity compared to multi-stream codecs.

This study propose a multi-layer multi-residual multi-stream (MMM) framework to extract discrete speech representation from continuous SSL representations. Specifically, we conduct RVQ-style quantization with K-means clustering to enable multi-stream discrete tokens in each single SSL layer. By combining with streams from multiple SSL layers, we further enhance the richness of the SSL-based units. With extensive experiments in ASR, we reveal that the proposed MMM-based discrete speech units elevates the performance of original single-stream SSL by a large margin and almost approaches the top-line performance with continuous SSL representation. While maintaining better ASR performance, we also demonstrate that the MMM-based units could achieve comparable or better performance to the neural codec-based approach in speech resynthesis and TTS.

2. Methodology

SSL discrete units are derived from a single-layer hidden representation within a specific SSL model. Given a speech signal $\boldsymbol{x}$, we represent an SSL model as S, which produces layer-wise representations denoted as $\mathbf{R} = [\boldsymbol{R}^1, \ldots, \boldsymbol{R}^L]$, where L signifies the number of layers in S. For a particular layer l, each element $\boldsymbol{r}^l \in \boldsymbol{R}^l$ comprises a sequence of vectors $[\boldsymbol{r}^l_1, \ldots, \boldsymbol{r}^l_T]$, with T representing the frame count. Upon selecting step t for analysis, we employ a l-th layer K-means model as K^l to determine K^l cluster centroids. With these cluster centroids obtained by K-means training, we can cluster the feature vectors $\boldsymbol{r}^l_t$ by finding the optimal cluster index, which minimizes the

10.21437/Interspeech.2024-2251

Euclidean distance between $\boldsymbol{r}_t^l$ and each cluster centroid $\boldsymbol{c}_k^l$ as:

$$k_t^l = \text{argmin}_{k \in 1,\dots,K^l} ||\boldsymbol{r}_t^l - \boldsymbol{c}_k^l||^2. \quad (1)$$

The final cluster IDs $[k_1^l, \dots, k_T^l]$ serve as discrete units at particularly layer l for subsequent downstream tasks

In this study, we broaden the application from a single-stream scenario to encompass multi-stream scenarios. To achieve this, we propose two complementary strategies for multi-stream modeling, which involve leveraging either a single layer or multiple layers from the SSL model S.

2.1. Multi-stream from a single layer

The first approach concentrates on generating multiple streams from a single layer within the SSL model. Citing [10], it's demonstrated that the single-layer continuous representation, specifically from a HuBERT-base model, contains sufficient detail for audio resynthesis at frame resolutions of both 40ms and 100ms. Nonetheless, the resynthesis audio quality markedly deteriorates following quantization via K-means. This decline highlights a significant limitation: discrete units derived from K-means quantization tend to omit intricate acoustic details originally present in the waveform signals.

To counteract the information loss inherent in quantization, our method involves estimating additional streams of discrete units. This process is aligned with the principle of RVQ, adhering to the unsupervised nature of the original K-means-based approach. For training, in the l-th layer, we iteratively estimate m-th K-means model $\mathrm{K}^{m,l}$ on the residual feature from previous K-means models $[\mathrm{K}^{1,l}, \dots, \mathrm{K}^{m-1,l}]$. During inference, the cluster index $k_t^{m,l}$ in frame t can then be obtained through iterative procedures. The following equations elaborate the detailed inference steps from the first stream to stream m.

$$k_t^{1,l} = \text{argmin}_{k \in 1,,\dots,K^{1,l}} \left|\left|\boldsymbol{r}_t^l - \boldsymbol{c}_k^{1,l}\right|\right|^2, \quad (2)$$

$$k_j^{2,l} = \text{argmin}_{k \in 1,,\dots,K^{2,l}} \left|\left|(\boldsymbol{r}_t^l - \boldsymbol{c}_t^{1,l}) - \boldsymbol{c}_k^{2,l}\right|\right|^2, \quad (3)$$

$$\vdots$$

$$k_j^{m,l} = \text{argmin}_{k \in 1,,\dots,K^{m,l}} \left|\left|\left(\boldsymbol{r}_t^l - \sum_{u=1}^{m-1} \boldsymbol{c}_t^{u,l}\right) - \boldsymbol{c}_k^{m,l}\right|\right|^2, \quad (4)$$

where we define $\boldsymbol{c}_t^{m,l}$ as the selected centroid at t, i.e., $\boldsymbol{c}_t^{m,l} := \boldsymbol{c}_k^{m,l}|_{k=k_t^{m,l}}$. Estimated $k_t^{m,l}$ becomes the discrete unit of the m-th stream at t-th frame.

2.2. Multi-stream from multiple layers

While Section 2.1 elaborated on enhancing discrete representations through a single SSL model layer l, an alternative strategy employs multi-layer representations. Historically, the integration of information across multiple layers in SSL models has proven invaluable, particularly when leveraging frozen SSL representations for downstream tasks. This is exemplified in the Speech Universal PERformance Benchmark (SUPERB), where layer-wise representations are combined through a weighted sum approach, achieving commendable results across a variety of speech processing tasks [12]. Subsequent research further validates this approach, demonstrating that different layers of an SSL model encapsulate distinct facets of speech-related information [7, 10, 11, 13, 33, 34].

In the context of discrete representations, the significance of utilizing multi-layer information becomes even more pronounced. Reliance on a single layer for information extraction may inadvertently prioritize certain features while overlooking others, a discrepancy that becomes especially noticeable following the quantization process (see Section 2.1). Therefore, a multi-layered approach not only diversifies the extracted information but also mitigates the risk of information bias, ensuring a more holistic representation of speech signals.

The formulation of multi-layer multi-stream discrete representations can be easily extracted from continuous representation $\mathbf{R}$. To be specific, we specify L' layers from the multi-layer representation $\mathbf{R}$. Then, for the K-means model $\mathrm{K}^{m,l'}$ of a selected layer l', the designated cluster IDs $[k_1^{m,l'}, \dots, k_T^{m,l'}]$ can be used as the discrete token for downstream tasks. The final MMM-based representation with $M \times L'$ streams can be obtained by combining the multi-stream extraction methods in Section 2.1 and Section 2.2.

2.3. Application of MMM-based discrete units

Building on existing literature using discrete representations [15, 16, 19, 20, 25, 30–32, 35, 36], we explore the application of MMM-based discrete units as either inputs or outputs across various applications. Our investigation encompasses both scenarios, employing a strategic approach to integrate discrete representations into downstream tasks.

Input Scenario: For representations used as inputs, we initially transform these discrete entities into embeddings. Subsequently, we apply a summation across different streams to integrate these embeddings. Specifically, for streams derived from a single layer, a direct summation of embeddings is executed, mirroring the inverse operation of the RVQ technique. Conversely, when dealing with streams from multiple layers, we adopt a learnable weighted summation for their aggregation. This method leverages learnable weights, optimized through a $\mathrm{Softmax}$ function, aligning with strategies from the SUPERB series [12, 13, 37–39]. This approach ensures that the integration of multi-layer streams is both dynamic and informed by the data. In cases where streams are sourced from both RVQ and multiple layers, we first combine embeddings from identical layers before proceeding to merge across different layers.

Output Scenario: For outputs utilizing MMM-based units, we implement a straightforward parallel approach that independently predicts various streams of tokens. This decision is different from prior work on neural audio codecs, where it was noted that models trained on RVQ necessitate auto-regressive modeling to maintain decoder quality [30, 32, 40]. Despite this, our pilot experiments indicate that the impact of such modeling on SSL-based units is minimal, thereby justifying our preference for a simpler, parallel prediction method for MMM-based discrete tokens.

3. Experiments

As discussed in Section 2.3, the discrete token can be applied in both input and output scenarios. For the input scenario, we select the two classical tasks of the proposed method: ASR and speech resynthesis (i.e., vocoder). The two tasks consider both usages in understanding and generation. For the output scenario, we conduct TTS, which serves as the backbone of most other tasks that produce discrete units [23, 41].

Table 1: *ASR performance on discrete speech challenge dataset. "+" indicates single-layer multi-stream setting and "†" stands for multi-layer multi-stream setting. M/L' corresponding to number layers in single/multi-layer multi-stream settings.*

SSL	Streams $M \times L'$	Librispeech WER	ML-SUPERB (1h) CER	Bitrate
WavLM	1 * 1	6.3	22.8	548.3
XLS-R	1 * 1	14.1	21.4	548.3
Encodec	8	15.9	35.9	6000.0
WavLM+	2 * 1	5.9	21.4	1096.6
WavLM†	1 * 4	5.0	20.8	2193.2
XLS-R+	2 * 1	10.5	19.3	1096.6
XLS-R†	1 * 4	7.3	18.0	2193.2
WavLM+†	2 * 4	**4.7**	19.5	4386.3
XLS-R+†	2 * 4	6.8	**17.5**	4386.3

3.1. Speech recognition

Dataset: Our ASR experiments align with the discrete speech challenge at Interspeech2024.[1] We utilize a dataset comprising Librispeech's train-clean-100h [42] combined with the ML-SUPERB multilingual 1-hour set [13]. This blend enables the examination of both clean English read speech and multilingual ASR tasks. The total dataset spans 310.4 hours and encompasses 143 languages, offering a broad spectrum for evaluation.
SSL Models: Informed by insights from [13, 15, 39] and the challenge's baseline, we recognize the distinct performance capabilities of WavLM [7] and XLS-R [8] across different corpora. WavLM-large exhibits notable effectiveness in English datasets, whereas XLS-R (300M version) is better suited for multilingual datasets. To comprehensively assess performance across both English and multilingual corpora, we employ these models as our primary SSL candidates.
Clustering and Downstream Settings: Consistent with the challenge's baseline, we opt for a random subsampling of 30% of the training set utterances for K-means clustering, setting the cluster size at 500 for each stream, i.e., $K^{m,l} = 500$. Deviating from the baseline, our methodology eschews additional byte-pair encoding and deduplication, simplifying the alignment of different streams. The downstream model leverages the same encoder-decoder architecture within ESPnet [43] as the challenge baseline.
Proposed Method: Following the methodology outlined in Section 2, we extract multi-stream tokens from both a single layer and multiple layers. For the single-layer approach, two ($M = 2$) streams of discrete representations are extracted from the 21st layer of both WavLM and XLS-R. For multi-layer scenarios, we select four ($L' = 4$) layers ({9, 15, 21, 22}) to balance compression efficiency and performance. As in Section 2.2, the two methods can be combined to yield eight streams of tokens.
Baseline and Ablation Studies: Our baseline comparison uses the single-layer, one-stream SSL units as defined in the challenge. Additionally, we use the publicly-availble Encodec-24kHz model [28] with 8 streams as another baseline. Ablation studies are conducted for both single-layer (with $M = 2, 4, 8$ streams) and multi-layer configurations, including a variety of layer selections. Layer selection for multi-layer scenarios is optimized using a model with all layers, employing learnable weighted summation (see Section 2.3), and then selecting the top four weighted layers for further analysis.

[1] https://www.wavlab.org/activities/2024/Interspeech2024-Discrete-Speech-Unit-Challenge/

Table 2: *ASR ablation studies on the single-layer setting over discrete speech challenge dataset.*

SSL	Streams M	Librispeech WER	ML-SUPERB (1h) CER	Bitrate
WavLM+	1	6.3	22.8	548.3
WavLM+	4	6.1	21.5	2193.2
WavLM+	8	6.4	21.7	4386.3
WavLM+	2	**5.9**	**21.4**	1096.6

Table 3: *ASR ablation studies on the multi-layer setting over discrete speech challenge dataset. Detailed layer indexes are shown in the "Layers" column.*

SSL	Layers L'	Librispeech WER	ML-SUPERB (1h) CER	Bitrate
WavLM†	21	6.3	22.8	548.3
WavLM†	1-4	6.8	27.7	2193.2
WavLM†	11-14	6.1	21.9	2193.2
WavLM†	21-24	5.5	21.5	2193.2
WavLM†	0-24	**4.9**	**19.9**	13707.2
WavLM†	9, 15, 21, 22	5.0	20.8	2193.2

Evaluation Metrics: We follow the setting in the discrete challenge for evaluation metrics, including average word error rate (WER) for Librispeech test sets and weighted average character error rate (CER) for ML-SUPERB test sets (i.e., normal test set and few-shot test set). Bitrate is also reported, following the discrete speech challenge guidelines.
Results and Discussion: The main results, as depicted in Table 1, illustrate that our proposed method uniformly enhances performance across both Librispeech and ML-SUPERB datasets for both SSL models. Additionally, the performance improvements yielded by the two proposed approaches appear to be complementary. Compared to Encodec tokens, even the single-stream discrete token sequences have better performances. The finding is aligned with the observations in [16] where SSL-based discrete representations outperform the neural codec-based method, specifically Encodec.

Our detailed ablation studies focusing on the number of streams are summarized in Tables 2 and 3. In the context of the single-layer multi-stream approach, augmenting the number of streams does not always lead to performance enhancement. As highlighted in Table 2, the ASR performance drops when the stream count is increased to 4 or 8, compared to the two-stream scenario. This deterioration could be attributed to the potential instability associated with employing K-means for higher-order residual information extraction. Conversely, for the multi-layer multi-stream scenario, optimal performance is attained when all layers are utilized at the cost of a higher bitrate. By selecting the layers with the top four weights, we show that it's possible to maintain performance levels with only a slight degradation while achieving a significant reduction in the bitrate.

3.2. Speech resynthesis (vocoder)

Dataset: Our speech resynthesis experiments are anchored in the TTS (vocoder) track of the discrete speech challenge at Interspeech 2024. We utilize a curated subset of the Expresso benchmark [26], focusing on a single-speaker dataset and excluding segments with singing, overlapping speech, and long-form content. The experiments adhere to the official train-dev-test partitioning provided by the challenge organizers.[2] To align

[2] https://github.com/ftshijt/Interspeech2024_DiscreteSpeechChallenge

Table 4: *Speech resynthesis performance on discrete speech challenge dataset (filtered Expresso). "+" indicates single-layer multi-stream setting and "†" stands for multi-layer multi-stream setting. M/L' corresponding to number layers in single/multi-layer multi-stream settings.*

SSL	Streams ($M \times L'$)	MCD	F0 RMSE	UTMOS	Bitrate
HuBERT	1 * 1	7.19	0.42	2.27	448.3
Encodec	8	**3.91**	0.21	3.18	3586.4
HuBERT (*S*)	2 * 1	6.79	0.32	2.89	896.6
HuBERT (*M*)	1 * 4	5.12	0.22	3.10	1793.2
HuBERT (*S*+*M*)	2 * 4	4.54	**0.20**	**3.22**	3586.4

Table 5: *TTS performance on discrete speech challenge dataset (LJSpeech).*

SSL	Streams ($M \times L'$)	MCD	F0 RMSE	WER	UTMOS	Bitrate
HuBERT	1 * 1	7.19	0.26	8.1	3.73	448.3
Encodec	8	**7.01**	0.29	7.8	4.01	3586.4
HuBERT (*S*)	2 * 1	7.11	0.29	8.0	3.79	896.6
HuBERT (*M*)	1 * 4	7.25	**0.24**	**7.7**	4.06	1793.2
HuBERT (*S*+*M*)	2 * 4	7.15	0.25	**7.7**	**4.15**	3586.4

with the original SSL model's specifications, audio samples at 48kHz are downsampled to 16kHz for compatibility.

Experimental Set-ups: Echoing prior studies [19, 20, 24, 26, 41] in discrete-based speech resynthesis that predominantly utilize HuBERT [6], our experiments also employ a pre-trained HuBERT-base model trained on the full Librispeech dataset via S3PRL[12]. Clustering is performed on the respective training sets with a designated cluster size of 500 per stream. The discrete HiFiGAN model serves as our downstream backbone, configured in accordance with [24]. For our baseline, the 9th layer is selected for extracting single-stream SSL units, drawing from the methodology in [19, 20, 24]. In exploring multi-stream capabilities, we investigate two ($M = 2$) streams for single-layer scenarios, four ($L' = 4$) streams for multi-layer configurations[3], and an integrated approach yielding eight streams ($M \times L' = 8$). We referred to these multi-stream configurations from the experimental findings in Section 3.1. Additional Encodec baselines are trained on the same training set based on AudioCraft [40]. We set the Encodec model with 8 streams at a 50Hz frame rate. For each stream, the codebook size is set to be 500, to be aligned with our experiments.

Evaluation Metrics: The evaluation framework prioritizes objective metrics in line with the discrete speech challenge's guideline. These metrics include mel cepstral distortion (MCD), F0 root mean square error (F0 RMSE), UTMOS [44], and bitrate. The MCD and F0 RMSE metrics are calculated using the ESPnet-TTS toolkit [45, 46].

Results and Discussion: Table 4 illustrates that our proposed method outperforms the baselines in UTMOS and F0 RMSE. Compared to the single-stream baseline, both of our proposed approaches not only improve all evaluated metrics, but also exhibit complementary benefits to each other. Notably, even when compared to Encodec, our method demonstrates superior UTMOS and F0 RMSE scores. This is particularly significant given that our discrete tokens are extracted unsupervisedly and are not explicitly optimized for the resynthesis task.

3.3. Text-to-speech

Dataset: For TTS, our examination leverages the LJSpeech dataset, a single-speaker female TTS corpus, in alignment with the discrete speech challenge recommendations. We rigorously adhere to the official dataset partitioning for the training set, as detailed by the challenge guidelines.

Experimental Set-ups: The TTS modeling experiments inherit several configurations from the speech resynthesis task. This includes utilizing the HuBERT-base SSL model, adopting a cluster size of 500 for K-means clustering, employing the entire training set for clustering, and integrating the discrete HiFi-GAN model. Distinctively, our downstream TTS model employs a modified VITS architecture, substituting its adversarial decoder with a transformer network designed to directly predict discrete units. This modification is informed based on the ESPnet VITS LJSpeech recipe and draws inspiration from [18].[4]

Evaluation Metrics: The evaluation framework utilize MCD, F0 RMSE, UTMOS and bitrate. In addition, we also report CER as assessed by a pre-trained Whisper-large-V2 model [47].

Results and Discussion: The outcomes of the TTS experiments, as displayed in Table 5, echo the observations made in the vocoder experiments detailed in Section 3.2. Compared to the Encodec model, we observe significant enhanced naturalness (reflected in improved UTMOS scores) with the TTS system trained on our proposed multi-stream tokens. This enhancement could be attributed to the SSL tokens possessing a richer semantic content, offering extra advantages in acoustic modeling when used as output. This semantic richness in SSL tokens likely facilitates a more effective acoustic representation for acoustic modeling, thereby enhancing the overall naturalness of the synthesized speech.

4. Conclusion

In this study, we reexamine the extraction of SSL-based discrete tokens, focusing on multi-stream modeling. We focus on two approaches: single-layer and multi-layer modeling. With extensive experiments across ASR, speech resynthesis, and TTS, we showcase that multi-stream tokens from SSL models can usually improve upon the single-stream SSL tokens. Moreover, we also find the proposed representations attain performance levels that are either superior to or on par with those achieved by neural codec methods.

[3]Layer 6, 9, 11, 12 are used, following the same selection strategy as the ASR track. We use the same setting for TTS experiments.

[4]Hyperparameters are in line with ESPnet VITS LJSpeech recipe.

5. Acknowledgements

This work was supported by a Meta AI SRA grant. Meta served in an advisory capacity here and no processing occurred on Meta servers. Jiatong Shi and Shinji Watanabe are funded in part of the Bridges2 system at PSC and Delta system at NCSA through allocations CIS210014 and IRI120008P from the ACCESS program, supported by NSF grants #2138259, #2138286, #2138307, #2137603, and #2138296.

6. References

[1] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, 1990.

[2] X. Huang *et al.*, *Spoken language processing: A guide to theory, algorithm, and system development*. 2001.

[3] T. N. Sainath *et al.*, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. Interspeech*, 2015.

[4] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, vol. 3, 2000.

[5] A. Baevski *et al.*, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, 2020.

[6] W.-N. Hsu *et al.*, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *TASLP*, vol. 29, 2021.

[7] S. Chen *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *JSTSP*, vol. 16, no. 6, 2022.

[8] A. Babu *et al.*, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," in *Proc. Interspeech*, 2022.

[9] C.-C. Chiu *et al.*, "Self-supervised learning with random-projection quantizer for speech recognition," in *Proc. ICML*, 2022.

[10] J. Shi *et al.*, "Exploration on HuBERT with Multiple Resolution," in *Proc. Interspeech*, 2023.

[11] J. Shi *et al.*, "Multi-resolution HuBERT: Multi-resolution speech self-supervised learning with masked unit prediction," in *Proc. ICLR*, 2024.

[12] S.-W. Yang *et al.*, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech*, 2021.

[13] J. Shi *et al.*, "ML-SUPERB: Multilingual Speech Universal PERformance Benchmark," in *Proc. Interspeech*, 2023.

[14] H.-J. Chang, S.-w. Yang, and H.-y. Lee, "DistillHuBERT: Speech representation learning by layer-wise distillation of hidden-unit bert," in *Proc. ICASSP*, 2022.

[15] X. Chang *et al.*, "Exploration of Efficient End-to-End ASR using Discretized Input from Self-Supervised Learning," in *Proc. Interspeech*, 2023.

[16] X. Chang *et al.*, "Exploring speech recognition, translation, and understanding with discrete speech units: A comparative study," in *Proc. ICASSP*, 2023.

[17] J. Shi *et al.*, "Bridging speech and textual pre-trained models with unsupervised ASR," in *Proc. ICASSP*, 2023.

[18] T. Hayashi *et al.*, "Discretalk: Text-to-speech as a machine translation problem," *arXiv preprint arXiv:2005.05525*, 2020.

[19] A. Polyak *et al.*, "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations," in *Proc. Interspeech*, 2021.

[20] A. Lee *et al.*, "Direct speech-to-speech translation with discrete units," in *Proc. ACL*, 2022.

[21] D. Zhang *et al.*, "Dub: Discrete unit back-translation for speech translation," *arXiv preprint arXiv:2305.11411*, 2023.

[22] J. Shi *et al.*, "Enhancing speech-to-speech translation with multiple tts targets," in *Proc. ICASSP*, 2023.

[23] L. Barrault *et al.*, "Seamless: Multilingual expressive and streaming speech translation," *arXiv preprint arXiv:2312.05187*, 2023.

[24] B. Yan *et al.*, "ESPnet-ST-v2: Multipurpose spoken language translation toolkit," in *Proc. ACL*, 2023.

[25] Y. Yang *et al.*, "Towards universal speech discrete tokens: A case study for ASR and TTS," in *Proc. ICASSP*, 2024.

[26] T. A. Nguyen *et al.*, "Expresso: A benchmark and analysis of discrete expressive speech resynthesis," *arXiv preprint arXiv:2308.05725*, 2023.

[27] N. Zeghidour *et al.*, "Soundstream: An end-to-end neural audio codec," *TASLP*, vol. 30, 2021.

[28] A. Défossez *et al.*, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[29] D. Yang *et al.*, "Hifi-codec: Group-residual vector quantization for high fidelity audio codec," *arXiv preprint arXiv:2305.02765*, 2023.

[30] C. Wang *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[31] X. Wang *et al.*, "Speechx: Neural codec language model as a versatile speech transformer," *arXiv preprint arXiv:2308.06873*, 2023.

[32] D. Yang *et al.*, "Uniaudio: An audio foundation model toward universal audio generation," *arXiv preprint arXiv:2310.00704*, 2023.

[33] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *Proc. ASRU*, 2021.

[34] X. Chang *et al.*, "An exploration of self-supervised pretrained representations for end-to-end speech recognition," in *Proc. ASRU*, 2021.

[35] Z. Borsos *et al.*, "Soundstorm: Efficient parallel audio generation," *arXiv preprint arXiv:2305.09636*, 2023.

[36] H. Wu *et al.*, "Codec-SUPERB: An in-depth analysis of sound codec models," *arXiv preprint arXiv:2402.13071*, 2024.

[37] T.-h. Feng *et al.*, "Superb@ SLT 2022: Challenge on generalization and efficiency of self-supervised speech representation learning," in *Proc. SLT*, 2023.

[38] H.-S. Tsai *et al.*, "SUPERB-SG: Enhanced speech processing universal performance benchmark for semantic and generative capabilities," in *Proc. ACL*, 2022.

[39] J. Shi *et al.*, "Findings of the 2023 ML-SUPERB challenge: Pre-training and evaluation over more languages and beyond," in *Proc. ASRU*, 2023.

[40] J. Copet *et al.*, "Simple and controllable music generation," *NeurIPS*, vol. 36, 2024.

[41] S. Maiti *et al.*, "Voxtlm: Unified decoder-only models for consolidating speech recognition/synthesis and speech/text continuation tasks," *arXiv preprint arXiv:2309.07937*, 2023.

[42] V. Panayotov *et al.*, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[43] S. Watanabe *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018.

[44] T. Saeki *et al.*, "UTMOS: Utokyo-sarulab system for voicemos challenge 2022," *arXiv preprint arXiv:2204.02152*, 2022.

[45] T. Hayashi *et al.*, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *Proc. ICASSP*, 2020.

[46] T. Hayashi *et al.*, "ESPnet2-TTS: Extending the edge of TTS research," *arXiv preprint arXiv:2110.07840*, 2021.

[47] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023.