Interspeech 2024
1-5 September 2024, Kos, Greece

The Interspeech 2024 Challenge on Speech Processing Using Discrete Units

Xuankai Chang[1], *Jiatong Shi*[1], *Jinchuan Tian*[1], *Yuning Wu*[4], *Yuxun Tang*[4], *Yihan Wu*[4], *Shinji Watanabe*[1], *Yossi Adi*[2], *Xie Chen*[3], *Qin Jin*[4]

[1]Carnegie Mellon University, USA [2]The Hebrew University of Jerusalem, Israel
[3]Shanghai Jiao Tong University, China [4]Renmin University of China, China

xuankaic@andrew.cmu.edu, jiatongs@andrew.cmu.edu

Abstract

Representing speech and audio signals in discrete units has become a compelling alternative to traditional high-dimensional feature vectors. Numerous studies have highlighted the efficacy of discrete units in various applications such as speech compression and restoration, speech recognition, and speech generation. To foster exploration in this domain, we introduce the Interspeech 2024 Challenge, which focuses on new speech processing benchmarks using discrete units. It encompasses three pivotal tasks, namely multilingual automatic speech recognition, text-to-speech, and singing voice synthesis, and aims to assess the potential applicability of discrete units in these tasks. This paper outlines the challenge designs and baseline descriptions. We also collate baseline and selected submission systems, along with preliminary findings, offering valuable contributions to future research in this evolving field.

Index Terms: discrete speech units, speech recognition, text-to-speech, singing voice synthesis

1. Introduction

In the realm of automatic speech recognition (ASR), considerable advancements have unfolded in the past few decades, propelled by the emergence of deep neural networks [1, 2]. Recently, the predominant approach has shifted towards end-to-end (E2E) ASR models [3–5], gaining popularity and witnessing performance enhancements through a spectrum of robust architectures [6–9]. Noteworthy strides have also been made in training methodologies, with self-supervised learning (SSL) models [10–12] and large-scale supervised training, such as Whisper [13], demonstrating improved performance and generalization. Traditionally, high-dimensional features are derived from raw waveforms in most endeavors. Spectral speech features, like Mel Frequency Cepstral Coefficients (MFCC) or log Mel filter banks (FBANK), conventionally stem from fixed-length temporal windows. Recently, learnt features based on deep neural networks through data-driven methods have become mainstream [10, 11, 14]. Despite these innovations, the data storage and transmission efficiency remain comparable between raw waveforms and speech features in many cases [15]. The challenge persists in enhancing computational efficiency without compromising performance integrity.

Recently, there has been a surge in the adoption of discrete speech representations, with notable developments in the Generative Spoken Language Model (GSLM) for textless Natural Language Processing (NLP) exemplified by [16, 17]. GSLM leverages techniques akin to those used in language modeling to address speech processing tasks through discrete speech representations. The representation of speech as discrete tokens presents a unique advantage, allowing for the unified modeling of both speech and text data within a streamlined framework. Several previous studies have highlighted the efficacy of jointly modeling speech-text data, showcasing improved performance in tasks related to speech and text generation [18–20]. Moreover, employing manipulation methods on discrete tokens enables the reduction of sequence length, resulting in more efficient computation [15, 16].

To encourage further exploration in this field, we propose the challenge of "Speech Processing Using Discrete Speech Units". The significance of this topic lies in its transformative potential across various applications within the community of speech and natural language processing [21–28]. The primary goal of this challenge is to advance innovation and investigation in the domain of discrete speech units, a field that has recently showcased remarkable potential but still lacks unified evaluation platforms to benchmark these methods. To fulfill this objective, we outline three core tasks:

1. The ASR task focuses on the multilingual aspect by incorporating data from the ML-SUPERB challenge [29].
2. The TTS task is divided into two tracks: a *single-speaker TTS track*, which focuses on synthesizing speech from text using a single voice, and a *vocoder track*, which concentrates on the resynthesis of expressive, multi-speaker speech.
3. The SVS task focuses on synthesizing single-singer singing from musical score information.

We chose these tasks due to their broad applicability and established benchmarks, which ensure clear evaluation metrics and significant real-world impact. These tasks cover the complete speech processing pipeline, encouraging holistic innovation in discrete unit processing. Additionally, they reflect current research trends and present diverse challenges that thoroughly test the capabilities of discrete unit representations, driving meaningful advancements in the field. This paper details the challenge designs, baselines, and evaluation metrics with ranking, which consist of ASR/TTS/SVS performance measures and compression rates. In addition, we provide preliminary analyses, including both baselines and selected results submitted at this juncture, to help us find new research directions.

2. Challenge Details

2.1. Formulation of discretization and bitrate

We denote an input waveform with T sampled data points with a sampling rate S as $\mathbf{x} \in \mathbb{R}^T$. This challenge defines discretization $f(\cdot)$ as a function to project $\mathbf{x}$ into a set of discrete sequence streams $\mathbf{U} = \{U^1, \ldots, U^M\}$, where we allow M streams and U^m denotes the m^{th} stream of discrete tokens. The U^m is defined as $U^m = (u_i^m \in \mathcal{V}_m | 1 \leq i \leq N_m)$, where N_m and $\mathcal{V}_m$ are the sequence length and the vocabulary/codebook of the m^{th}

10.21437/Interspeech.2024-1878

stream, respectively.

Based on this formulation, we define the bitrate B (bit/second) of the discrete representation $\mathbf{U}$ given the original waveform samples length T and its sampling rate S:

$$B = \sum_{m=1}^{M} \left(\frac{N_m \cdot \log_2(|\mathcal{V}_m)|}{T/S} \right), \quad (1)$$

which corresponds to the sum of the bitrates from all levels. Bitrate is an important metric in our challenge to measure the efficiency of discrete representation.

2.2. ASR task

To assess the fidelity of semantic information, we incorporate the ASR track in the challenge.

Task Definition and Baseline: The target of ASR is to transcribe speech signals into text. Traditionally, feature extraction is applied to an audio segment of length W ($W \leq T$), represented as $\mathbf{x}_i = \mathbf{x}[t_i : t_i + W]$, undergoes conversion to a D-dimensional vector of real or complex values, denoted as $\mathbf{X}_i \in \mathbb{C}^D$. Here, $\mathbf{X}_i$ signifies the feature of that segment, commonly referred to as a frame. In the context of ASR tasks utilizing discrete units, the feature of a frame, $\mathbf{X}_i$, is represented as $U_i = \{u_i^1, \ldots, u_i^M\}$. Notably, in certain instances, as seen in [15], $M = 1$ is employed. In such cases, the sizes of $\mathbf{X}_i$ and u_i are $32 \times D$ and $\log_2(|\mathcal{V}|)$ bits, respectively, under the assumption that $\mathbf{X}_i$ is stored in 32-bit float value and $|\mathcal{V}|$ denotes the size of the codebook.

In the study conducted by Chang *et al.* [15], it was established that discrete units-based ASR systems exhibit proficient performance on the majority of mono-lingual datasets. However, challenges arise in the context of multi-lingual scenarios, as evidenced by the ML-SUPERB dataset [29]. Consequently, this challenge deliberately emphasizes and promotes the multi-lingual dimension of discrete units-based ASR.

Data: To stress the multi-lingual aspect mentioned above, in addition to the widely-used LibriSpeech [30] 100-hour subset (LibriSpeech-100), we also adopt the ML-SUPERB 1-hour public benchmark [29] in the ASR task. LibriSpeech-100 comprises a clean, read English corpus, effectively addressed by the discrete units-based ASR [31]. In contrast, ML-SUPERB presents a more formidable challenge, given the complexities of language families with 143 languages and the limited volume available for each language. Notably, the 1-hour track from ML-SUPERB encompasses approximately 220 hours of speech. The training sets of both corpora are combined to train the ASR model, with the inclusion of LibriSpeech-100 aimed at easing training complexities and showcasing performance on a data-rich resource. As for the evaluation, we employ all test sets from LibriSpeech (dev-clean, dev-other, test-clean, and test-other) and ML-SUPERB (test_1h). The data preparation scripts are included in the baseline by following conventional methods of LibriSpeech-100 and ML-SUPERB. It is important to highlight that there are no constraints on the data used for obtaining discrete tokens in this challenge, including pre-training and k-means training.

Evaluation Metrics: Two evaluation metrics are employed: Character Error Rate (CER) and bitrate.

- CER: The test sets are categorized into two groups, encompassing English and multi-lingual content. Consequently, two CERs are computed: CER_{EN} and CER_{ML}. CER_{EN} is calculated across all utterances in the LibriSpeech test sets, while CER_{ML} is computed on the ML-SUPERB test set. The adoption of CER for LibriSpeech ensures consistency with ML-SUPERB.
- Bitrate: The calculation follows Eq. (1). We compute the bitrate on the whole test sets, i.e., all librispeech evaluation sets and ML-SUPERB test sets.

Ranking: The overall ranking is based on the average of all three ranking positions:

- R_1: micro average CER_{EN} on all LibriSpeech test sets;
- R_2: CER_{ML} on the ML-SUPERB test set;
- R_3: the bitrate of the overall test sets.

The overall ranking position is $\hat{R} = \frac{(R_1+R_2+R_3)}{3}$. In cases where multiple systems share the same ranking, the tiebreaker is determined by the order $R_2 > R_1 > R_3$.

2.3. TTS (Vocoder) task

In the TTS (vocoder) track of the challenge, the focus is on the conversion of discrete speech units into waveforms, assessing the acoustic information within these units.

Task Definition: The core objective of vocoder modeling (speech resynthesis) is to develop a reverse function $f^{-1}(\cdot)$ capable of transforming discrete speech units $\mathbf{U}$ into an audible waveform $\hat{\mathbf{x}}$. No restrictions are placed on the type or size of the model used for the vocoder.

Data: The dataset for this task is sourced from the Expresso benchmark [32], focusing solely on single-speaker scenarios to avoid complications with multi-speaker conversions and long-form speech. The data is partitioned into training (9.7 hours), development (0.6 hour), and test (0.6 hour) sets, and while discrete unit learning can utilize external data, vocoder training is restricted to the provided training dataset.

Evaluation metrics: Four metrics are employed for evaluation: Mel cepstral distortion (MCD), F0 root mean square error (F0 RMSE), UTMOS [33], and bitrate. UTMOS is calculated using the winner model from the VoiceMOS 2022 challenge [34], and the bitrate calculation is standardized as Eq. (1). The evaluation process is facilitated by ESPnet-TTS [35, 36].

Ranking: Similar to the ASR task, the final ranking is determined by averaging the ranks across two primary metrics: UTMOS and bitrate. UTMOS is ranked in descending order, while bitrate is ranked in ascending order. To allow different focuses on sampling rates, we separate the bitrate into two groups (16kHz and 48kHz), depending on the sampling rate of the resynthesized waveform. The ranking of both UTMOS and bitrate would be considered separately in each group. If there's a tie in the overall ranking, UTMOS rankings will serve as a tiebreaker to establish the final positions.

2.4. TTS (Acoustic + Vocoder) task

In the challenge, the TTS (Acoustic + Vocoder) track focuses on the use of discrete units as an intermediate representation in a cascaded TTS system. Here the cascaded TTS highlights the TTS system that consists of both an acoustic model and a vocoder. This approach is supported by several research findings suggesting that discrete representations offer considerable benefits for speech synthesis systems. The potential benefits include easy predictability, stability during training, and versatility in interacting with different modalities [19, 37–40]. Participants are encouraged to explore the use of discrete units to enhance both the performance and efficiency of TTS systems.

Task Definition: The challenge's TTS task involves converting text into speech signals. Participants are required to use a cas-

caded TTS system where the acoustic model translates text into discrete units $\mathbf{U}$, and the vocoder converts $\mathbf{U}$ into the predicted waveform $\hat{\mathbf{x}}$. The model type or size for both the acoustic model and the vocoder is not subject to any limitations.

Data: The challenge focuses on a single-speaker TTS task using the LJSpeech dataset [41], with 250 utterances set aside for both development and test purposes. While there are no restrictions on the data used for learning or extracting discrete units, the provided training data must exclusively be used for training the TTS system components.

Evaluation metrics: The evaluation includes the same four metrics as in the TTS (Vocoder) track, with the addition of the word error rate (WER) from Whisper-large V2 [13].

Ranking: The ranking methodology mirrors that of the TTS (Vocoder) track, focusing on a combined assessment of speech quality and discrete unit efficiency to determine the overall performance standings. In case of a tie in the overall ranking, UTMOS ranking will be the tiebreaker for final positions.

2.5. SVS track

The SVS track distinguishes itself from TTS by focusing on the intersection of music and speech processing. Unlike previous works that often extend TTS frameworks to SVS [40, 42, 43], this challenge treats singing synthesis as an independent track to foster deeper exploration into singing-specific features.

Task Definition: Singing synthesis entails generating singing voices using musical score information. Mirroring the TTS (Acoustic + Vocoder) track, this challenge adopts a cascaded approach, incorporating an acoustic model and a vocoder. The acoustic model's role is to translate the music score into a sequence of discrete units $\mathbf{U}$, while the vocoder is tasked with synthesizing the waveform from $\mathbf{U}$. There are no additional constraints imposed on SVS modeling for this challenge.

Data: For the SVS track, the dataset employed is the 5.2-hour single-singer Opencpop dataset [44]. The challenge adheres to the original dataset's train, development, and test splits. Similar to the TTS tracks, training for the SVS track must only utilize the provided dataset, although any data source is permissible for extracting discrete representations.

Evaluation Metrics: The evaluation for the SVS track encompasses four metrics: MCD, F0 RMSE, MOS, and bitrate. The objective metrics (MCD, F0 RMSE, and bitrate) follow the same calculation methodology as in the TTS tracks. For MOS, 20 subjects rate the submissions (i.e., 206 utterances) on a 5-point scale, with 1 indicating "unreasonable singing" and 5 denoting "natural singing comparable to human performance."

Ranking: The overall ranking is determined by averaging the ranks across two key metrics: MOS and bitrate. MOS rankings are in descending order, while bitrate rankings are in ascending order. In the event of tied rankings, priority is given to the MOS results for final ranking decisions.

3. Baseline Systems

3.1. ASR baseline

The baseline system follows the model used in [15] and is implemented using ESPnet [45]. The ASR backbone uses the joint CTC/attention-based encoder-decoder architecture based on the E-Branchformer [9][1]. The baseline model undergoes training for 100 epochs, utilizing a single Nvidia V-100 32GB GPU, with a total training time of about 18 hours.

For discrete speech units, $1,024$-dimensional features are extracted from the 21-st layer of the WavLM-Large [12] model. A k-means model with $2,000$ clusters is trained using randomly chosen 15% of the data from the training set, described in Section. 2.2. Additionally, repeated tokens are removed, and the BPE model is applied with a vocabulary size of $6,500$, i.e. $|\mathcal{V}| = 6,500$ in Section. 2.2.

3.2. TTS baseline

Vocoder track: For the TTS (Vocoder) track, the baseline involves k-means ($k = |\mathcal{V}| = 500$) clustering over the whole training set on the 9$^{\text{th}}$ layer outputs of a pre-trained HuBERT-base model [11]. The setting is aligning with previous SSL-based unit extraction methodologies [19,32,37,38,46–48]. The derived token sequence is processed using a discrete-token-based HiFi-GAN within the ESPnet framework [35–37].[2]

Acoustic + Vocoder track: For the TTS (Acoustic + Vocoder) track, we separately train an acoustic model and a vocoder. For the vocoder, we use the same vocoder setting as the TTS (Vocoder) track with LJSpeech training data. For the acoustic model, we adopt a Fastspeech2 architecture [49], adapted to output discrete units instead of spectrograms. The acoustic model configuration follows the LJSpeech Fastspeech2 recipe in ESPnet-TTS [35].[3]

3.3. SVS baseline

The SVS baseline consists of an acoustic model and a vocoder. The acoustic model is adapted from XiaoiceSing [42]. We replace the output spectrogram in original XiaoiceSing into two streams, including one stream of quantized fundamental frequency (with a resolution of 10Hz) and another stream with semantic discrete tokens. The discrete tokens are extracted from the 6$^{\text{th}}$ layer of WavLM-large with a k-means ($k = |\mathcal{V}| = 1024$) over the whole training set. The acoustic model consists of an encoder, a length regulator and a decoder. The implementation is based on ESPnet-Muskits [50]. The network architecture and the training configuration follow the XiaoiceSing model configuration in corresponding Opencpop recipe.[4] The vocoder utilizes the same architecture as the TTS baselines.

4. Preliminary Results

This section presents the initial results that we collected before the paper deadline. Due to time constraints, a more in-depth analysis and detailed results will be presented following the conclusion of the challenge.

4.1. ASR results

Prior to the submission deadline, nine systems were submitted for the ASR track. We list the performance of the top-3 submitted systems in Table 2. In comparison to the provided baseline system (B1), the submitted system S2 outperformed on all metrics. S2 employed a similar discrete token process as the baseline, utilizing the XLSR2-300M for feature extraction. A 2000-

[1]We follow the model configurations in https://github.com/espnet/espnet/blob/master/egs2/interspeech2024_dsu_challenge/asr2/conf/tuning/train_discrete_asr_e_branchformer1_1gpu_lr5e-4_warmup30k.yaml

[2]We follow the model configurations in https://github.com/kan-bayashi/ParallelWaveGAN/blob/master/egs/cvss_c/hubert_voc1/conf/hifigan_hubert_duration.v1.yaml

[3]https://github.com/espnet/espnet/blob/master/egs2/ljspeech/tts1/conf/tuning/train_fastspeech2.yaml

[4]https://github.com/espnet/espnet/blob/master/egs2/opencpop/svs1/conf/tuning/train_xiaoice.yaml

Table 1: *The performance of the baseline and submitted systems on the ASR task. We use CERs on the English test sets and the multi-lingual counterpart, as well as the bitrate. Brief discrete token information collected from the participants is added.*

Team ID	Discrete token info	CER_{EN}	CER_{ML}	Bitrate
B1	WavLM_Large_21st, k-means	2.37	22.40	356.19
S1	(XLSR2_300M, WavLM_Large), k-means	1.91	16.03	946.77
S2	XLSR2_300M, k-means	2.21	17.32	262.64
S3	(WavLM_Large, XLS-R), HMM-GMM	1.98	20.23	599.20

Table 2: *The performance of the baseline and submitted systems on the TTS (Vocoder) task. S is the sampling rate of targeted audio from the system.*

Team ID	S	MCD	F0 RMSE	UTMOS	Bitrate
B1	16k	7.19	0.42	2.27	**448.3**
S1	16k	6.24	0.24	**3.59**	547.0
S2	24k	4.81	0.21	3.58	670.3
S3	16k	3.57	**0.18**	3.57	1479.5
S4	48k	**3.54**	**0.18**	3.56	1479.5
S5	48k	4.47	**0.18**	3.48	834.0
S6	48k	4.47	**0.18**	3.48	834.0

cluster k-means model was applied to the features, followed by BPE with a vocabulary size of 6000. This approach resulted in a notable 7%, 23%, and 26% reduction in CER_{EN}, CER_{ML}, and bitrate, respectively. S1 and S3 utilize the fusion techniques to combine the discrete tokens from multiple streams. In contrast to other approaches, S3 employs the Hidden Markov Model (HMM) for computing the discrete tokens.

4.2. TTS results

For this track, we received 13 systems for the TTS (Vocoder) task and 8 systems for the TTS (Acoustic + Vocoder) task. In this paper, we present the preliminary results by selecting the top three systems in terms of the overall ranking.

For the TTS (Vocoder) track, all six top systems from both 16kHz and 48kHz settings surpass the baseline by a large margin in UTMOS score and other objective evaluation metrics, suggesting better resynthesis quality. Model S1 refines the SSL pre-trained representation with audio resynthesis tasks, and shows the best UTMOS score in the challenge. Different from B1 and S1 originated from SSL pre-trained models, the other two methods S2 and S3 adapt neural codec-based models, including Descript Audio Codec (DAC) [51] and APCodec [52]. The discrete unit extractor is optimized on the audio resynthesis task with adversarial training. The codebooks from the pre-trained codecs are then used as the discrete representation for the task. Notably, their bitrates are generally higher than B1 and S1 due to the use of multi-stream information.

In the TTS (Acoustic + Vocoder) track, we compare the top three models: S1 and S2 employ discrete representations from a DAC-based neural codec, while S3 utilizes explicit vector quantization within an end-to-end TTS training framework. S3 stands out by delivering the highest UTMOS scores and the lowest WER, illustrating its superior performance. However, this comes at the expense of a higher bitrate. Conversely, S1 achieves a commendable equilibrium between bitrate efficiency and UTMOS performance, presenting a viable option for scenarios where a balance between audio quality and resource usage is essential.

Table 3: *The performance of the baseline and submitted systems on the TTS (Acoustic + Vocoder) task.*

Team ID	MCD	F0 RMSE	WER	UTMOS	Bitrate
B1	7.19	**0.26**	8.1	3.73	448.3
S1	**6.96**	0.29	7.7	4.33	**277.6**
S2	7.15	0.29	7.4	4.33	353.9
S3	7.70	0.29	**6.8**	**4.42**	727.5

Table 4: *The performance of the baseline and submitted systems on the SVS task. 95% confidence interval is in parentheses.*

Team ID	MCD	F0 RMSE	MOS	Bitrate
B1	8.47	0.18	3.43 ($\pm$ 0.05)	2094.7
S1	**7.56**	**0.17**	**3.70** ($\pm$ 0.05)	1899.9
S2	7.72	0.19	3.09 ($\pm$ 0.06)	874.8
S3	11.44	0.24	2.73 ($\pm$ 0.06)	725.9

4.3. SVS results

For the SVS challenge, six systems were submitted. We focus on the top-3 systems based on their performance metrics. S1 and S2 employ SSL-based discrete tokens within a non-autoregressive framework, contrasting with S3, which is built on neural codecs and operates in an autoregressive manner. Despite the varied configurations among the models, S1 and S2, along with baseline B1, outperform S3. This superiority could be attributed to the limited training data provided in the challenge, introducing additional challenges to autoregressive modeling. Though the data scarciy is a common constraint in SVS tasks, the SSL-based discrete units utilized in S1 and S2 appear to offer robust representations for discrete SVS systems.

5. Conclusion

This paper serves as a comprehensive overview of the Interspeech 2024 challenge on speech processing with discrete units. The challenge garnered a notable 40 submissions across ASR, TTS, TTS-vocoder, and SVS tasks, underscoring the significant interest in this domain. We provide detailed insights into the motivation, challenge rules, baseline systems, and initial submission results.

Upon reviewing the initial submissions, several initial observations can be made. Notably, in the ASR task, the utilization of semantic tokens from SSL models demonstrates promising outcomes. While for TTS tasks, neural codec-based model usually exhibit high-quality acoustics, which significantly enhance the synthesized audio quality. In the SVS track, on the other hand, SSL-based units demonstrate strong performance over the dataset, suggesting the rich acoustic information can be also obtained from SSL-based pre-trained models in the singing domain. However, to derive more nuanced and conclusive findings, a thorough and in-depth analysis requires additional time and efforts. Detailed analyses and findings will be unveiled as we invest the necessary resources in their examination.

6. Acknowledgements

Experiments of this work used the Bridges2 system at PSC and Delta system at NCSA through allocations CIS210014 and IRI120008P from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, supported by NSF grants #2138259,#tel:2138286, #tel:2138307, #tel:2137603, and #tel:2138296.

7. References

[1] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[2] Y. Qian *et al.*, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Trans. ASLP.*, vol. 24, no. 12, pp. 2263–2276, 2016.

[3] A. Graves *et al.*, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[4] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[5] J. Chorowski *et al.*, "Attention-based models for speech recognition," *Proc. NeurIPS*, vol. 2015, pp. 577–585, 2015.

[6] L. Dong *et al.*, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE ICASSP*, 2018.

[7] A. Gulati *et al.*, "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. Interspeech*. ISCA, 2020, pp. 5036–5040.

[8] P. Guo *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *Proc. IEEE ICASSP*, 2021, pp. 5874–5878.

[9] K. Kim *et al.*, "E-branchformer: Branchformer with enhanced merging for speech recognition," in *Proc. IEEE SLT*, 2023, pp. 84–91.

[10] A. Baevski *et al.*, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Proc. NeurIPS*, vol. 33, pp. 12 449–12 460, 2020.

[11] W.-N. Hsu *et al.*, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. ASLP.*, vol. 29, pp. 3451–3460, 2021.

[12] S. Chen *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE JSTSP*, vol. 16, no. 6, pp. 1505–1518, 2022.

[13] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023, pp. 28 492–28 518.

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

[15] X. Chang *et al.*, "Exploration of efficient end-to-end asr using discretized input from self-supervised learning," *arXiv preprint arXiv:2305.18108*, 2023.

[16] K. Lakhotia *et al.*, "On generative spoken language modeling from raw audio," *Trans. ACL.*, vol. 9, pp. 1336–1354, 2021.

[17] E. Kharitonov *et al.*, "Text-free prosody-aware generative spoken language modeling," *arXiv preprint arXiv:2109.03264*, 2021.

[18] P. K. Rubenstein *et al.*, "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv:2306.12925*, 2023.

[19] 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.

[20] M. Hassid *et al.*, "Textually pretrained speech language models," *arXiv preprint arXiv:2305.13009*, 2023.

[21] F. Kreuk *et al.*, "Textless speech emotion conversion using discrete and decomposed representations," *arXiv preprint arXiv:2111.07402*, 2021.

[22] T. A. Nguyen *et al.*, "Generative spoken dialogue language modeling," *Trans. ACL.*, vol. 11, pp. 250–266, 2023.

[23] G. Maimon and Y. Adi, "Speaking style conversion with discrete self-supervised units," *arXiv preprint arXiv:2212.09730*, 2022.

[24] T. Hayashi and S. Watanabe, "Discretalk: Text-to-speech as a machine translation problem," *arXiv preprint arXiv:2005.05525*, 2020.

[25] M. Kim *et al.*, "Many-to-many spoken language translation via unified speech and text representation learning with unit-to-unit translation," *arXiv preprint arXiv:2308.01831*, 2023.

[26] F. Kreuk *et al.*, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.

[27] J. Copet *et al.*, "Simple and controllable music generation," *arXiv preprint arXiv:2306.05284*, 2023.

[28] W.-N. Hsu *et al.*, "Revise: Self-supervised speech resynthesis with visual input for universal and generalized speech regeneration," in *Proc. IEEE/CVF CVPR*, 2023, pp. 18 795–18 805.

[29] J. Shi *et al.*, "ML-SUPERB: Multilingual Speech Universal PERformance Benchmark," in *Proc. Interspeech*, 2023, pp. 884–888.

[30] V. Panayotov *et al.*, "Librispeech: an asr corpus based on public domain audio books," in *Proc. IEEE ICASSP*, 2015, pp. 5206–5210.

[31] X. Chang *et al.*, "Exploring speech recognition, translation, and understanding with discrete speech units: A comparative study," *arXiv preprint arXiv:2309.15800*, 2023.

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

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

[34] W. C. Huang *et al.*, "The VoiceMOS Challenge 2022," in *Proc. Interspeech*, 2022, pp. 4536–4540.

[35] T. Hayashi *et al.*, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *Proc. IEEE ICASSP*. IEEE, 2020, pp. 7654–7658.

[36] ——, "ESPnet2-TTS: Extending the edge of tts research," *arXiv preprint arXiv:2110.07840*, 2021.

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

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

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

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

[41] K. Ito and L. Johnson, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[42] P. Lu *et al.*, "XiaoiceSing: A high-quality and integrated singing voice synthesis system," *Proc. Interspeech*, 2020.

[43] Y. Zhang *et al.*, "VISinger: Variational inference with adversarial learning for end-to-end singing voice synthesis," in *Proc. IEEE ICASSP*, 2022.

[44] Y. Wang *et al.*, "Opencpop: A High-Quality Open Source Chinese Popular Song Corpus for Singing Voice Synthesis," in *Proc. Interspeech*, 2022, pp. 4242–4246.

[45] S. Watanabe *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[46] A. Polyak *et al.*, "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations," in *Proc. Interspeech*, 2021, pp. 3615–3619.

[47] A. Lee *et al.*, "Direct speech-to-speech translation with discrete units," in *Proc. ACL*, 2022, pp. 3327–3339.

[48] J. Shi *et al.*, "Enhancing speech-to-speech translation with multiple TTS targets," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[49] Y. Ren *et al.*, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *Proc. ICLR*, 2020.

[50] J. Shi *et al.*, "Muskits: an end-to-end music processing toolkit for singing voice synthesis," in *Proc. Interspeech*, 2022.

[51] R. Kumar *et al.*, "High-fidelity audio compression with improved rvqgan," *Proc. NeurIPS*, vol. 36, 2024.

[52] Y. Ai *et al.*, "APCodec: A neural audio codec with parallel amplitude and phase spectrum encoding and decoding," *arXiv preprint arXiv:2402.10533*, 2024.