

# TOWARDS UNIVERSAL SPEECH DISCRETE TOKENS: A CASE STUDY FOR ASR AND TTS

Yifan Yang<sup>\*,1</sup>, Feiyu Shen<sup>\*,1</sup>, Chenpeng Du<sup>1</sup>, Ziyang Ma<sup>1</sup>, Kai Yu<sup>1</sup>, Daniel Povey<sup>†,2</sup>, Xie Chen<sup>†,1</sup>

<sup>1</sup> MoE Key Lab of Artificial Intelligence, AI Institute  
X-LANCE Lab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China  
<sup>2</sup> Xiaomi Corporation, Beijing, China

## ABSTRACT

Self-supervised learning (SSL) proficiency in speech-related tasks has driven research into utilizing discrete tokens for speech tasks like recognition and translation, which offer lower storage requirements and great potential to employ natural language processing techniques. However, these studies, mainly single-task focused, faced challenges like overfitting and performance degradation in speech recognition tasks, often at the cost of sacrificing performance in multi-task scenarios. This study presents a comprehensive comparison and optimization of discrete tokens generated by various leading SSL models in speech recognition and synthesis tasks. We aim to explore the universality of speech discrete tokens across multiple speech tasks. Experimental results demonstrate that discrete tokens achieve comparable results against systems trained on FBank features in speech recognition tasks and outperform mel-spectrogram features in speech synthesis in subjective and objective metrics. These findings suggest that universal discrete tokens have enormous potential in various speech-related tasks. Our work is open-source and publicly available at <https://github.com/k2-fsa/icefall>.

**Index Terms**— self-supervised learning, discrete tokens, speech recognition, text-to-speech

## 1. INTRODUCTION

Self-supervised learning (SSL) [1, 2, 3, 4, 5, 6, 7] is a robust methodology capable of leveraging large amounts of unlabeled data to deliver substantial performance improvements. In speech-related downstream tasks, representations derived from SSL stand out for their generalizability and accessibility. While most existing research focuses on utilizing real-valued features from intermediate layers in SSL models for downstream tasks, there is scope for investigating the use of discretized token sequences. Speech discrete tokens can be broadly divided into semantic tokens and acoustic tokens [8]. Models including vq-wav2vec [3], wav2vec 2.0 [2], HuBERT [4], and WavLM [5] generate semantic tokens trained for discrimination or masking prediction, maintaining linguistic information of the speech. In contrast, acoustic tokens, produced by audio neural codec models like Soundstream [9] and Encodec [10], aim to reconstruct speech accurately, reflecting acoustic details.

Previous works have studied the application of discrete tokens in individual speech tasks. Baevski et al. [11] examined the utilization of discrete tokens in automatic speech recognition (ASR) and found underperformed compared to traditional FBank features unless combined with a BERT [12] as a preprocessing step. Chang et al. [13]

employed refined discretized token sequences via de-duplication and subword modeling techniques, approximating the performance of conventional acoustic features while reducing computational and storage costs. Du et al. [14] presented VQTTS, incorporating a txt2vec acoustic model and a vec2wav vocoder. This system utilizes self-supervised Vector Quantization (VQ) acoustic features to mitigate the adverse impact of the disparity between the ground truth and predicted mel-spectrograms, thereby enhancing the overall text-to-speech (TTS) system performance. Subsequently, Du et al. [15] introduced UniCATS, a TTS system based on semantic tokens and free of acoustic tokens and speaker embeddings. It integrates both semantic and acoustic contexts through the use of contextual VQ-diffusion in its CTX-txt2vec component and vocoding in its CTX-vec2wav component, achieving state-of-the-art (SOTA) TTS performance. Nguyen et al. [16] utilized discrete tokens from HuBERT and Encodec for speech resynthesis and found Encodec, designed for general audio compression, outperformed HuBERT units in this task.

The application of discrete tokens in multiple tasks remains preliminary. Recently, Wang et al. [17] presented VioLA, transforming speech signals into discrete tokens, enabling a unified conditional language modeling way for multiple spoken processing tasks. However, they only employ codec to build a multi-task model with limited performance compared to single tasks. Systematic experiments on different tasks using varied discrete tokens are still lacking.

In this context, this study aims to investigate the universality of speech discrete tokens across multiple speech-related tasks. We take two representative speech processing tasks, speech recognition and speech synthesis, for a case study. As illustrated in Fig. 1, we transform raw audio into speech discrete tokens using four leading SSL models, including vq-wav2vec, encodec, HuBERT, and WavLM. For ASR, we utilize these discrete tokens and their associated texts to train end-to-end (E2E) ASR models [18]. We introduce customized data augmentation techniques for discretized inputs to mitigate overfitting. In order to get a whole picture of the potential of discrete tokens, we train and evaluate this ASR model on three distinct data sets, including LibriSpeech, GigaSpeech, and AISHELL-1. Note that AISHELL-1 is a Mandarin speech corpus, while all SSL models are pretrained with English audio. For TTS, we conduct experiments on speech resynthesis tasks to explore its upper bound without interference from the acoustic model. We trained vocoders with various discrete tokens on LibriTTS and compared their performance through both subjective and objective evaluations. Notably, using discrete tokens yields superior performance compared to conventional real-valued mel-spectrogram features.

## 2. DISCRETE SPEECH TOKENS FROM SSL MODELS

In self-supervised learning models, continuous audio can be discretized using two primary methods: VQ and k-means clustering

\* stands for equal contribution. † stands for corresponding authors.

This work was supported by the National Natural Science Foundation of China (No. 62206171), and the International Cooperation Project of PCL, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

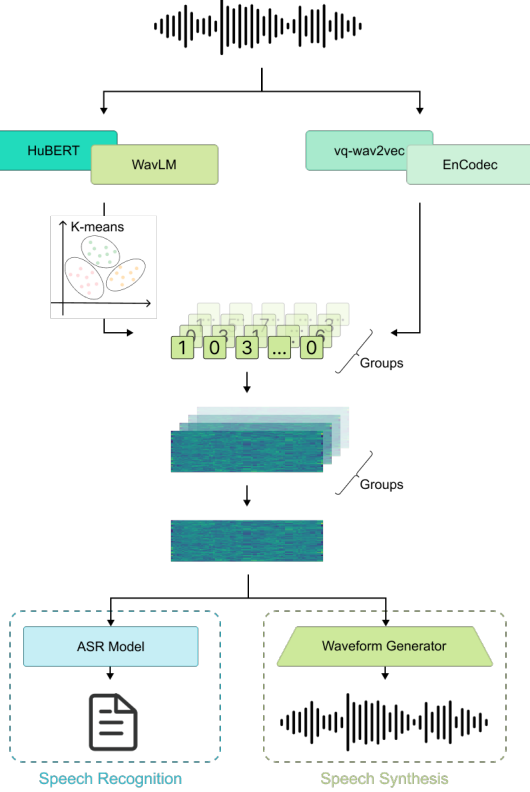


Fig. 1: Illustration of the pipeline for speech discrete tokens.

on embeddings. `vq-wav2vec` employs autoregressive Convolutional Neural Networks (CNNs) to extract feature representations from raw audio, which are then organized into multiple groups. Subsequently, vector quantization is applied in each group, facilitated by either the Gumbel-Softmax or online k-means clustering, yielding multiple sets of discrete token sequences. By contrast, `EnCodec` applies Residual Vector Quantization (RVQ) [9] on the output of the convolutional-based encoder. For `HuBERT` and `WavLM`, the discretization of speech can be realized by applying k-means clustering to the hidden embeddings from some specified Transformer-Encoder layer.

In practice, a pre-trained `vq-wav2vec` model is employed to process 16 kHz audio and outputs embeddings at a 100 Hz rate with 2 groups. A pre-trained `EnCodec` processes 24 kHz audio at varying bitrates, which generates 75 Hz embeddings from 24 kHz inputs. For RVQ, eight hierarchical quantizers with 1024 entries are employed. Hidden embeddings are extracted using pre-trained `WavLM Large` and `HuBERT Large` models. The final Transformer-Encoder layer is chosen for the ASR task according to the semantic correlation analysis in [19].

Discretization granularity can be adjusted by varying the number of clusters in the k-means algorithm. Specifically, 2000 clusters are chosen based on reports that more discrete tokens boost phone-normalized mutual information (PNMI) and enhance ASR performance [13].

### 3. SPEECH RECOGNITION WITH DISCRETE TOKENS

#### 3.1. ASR with Discretized Input

We individually employ discrete speech tokens of each type, along with their corresponding texts, to train distinct E2E ASR models,

leveraging RNN-T loss for optimization. These tokens are projected to 80 dimensions through a linear embedding layer. When several feature groups exist, each one is embedded, concatenated, and then projected to match the length of a single group. Subsequently, these features are interpolated to a uniform 100 Hz rate before feeding into the ASR model.

#### 3.2. Discretized Input Augmentation Policy

We engineer data augmentation techniques tailored for discretized inputs to mitigate overfitting and enhance robustness for discrete token representations. Inspired by the `SpecAugment` [20] strategy for continuous features like `FBanks`, our policy incorporates the following deformations on the discrete tokens sequence  $\mathbf{x} = (x_1, \dots, x_T)$  of length  $T$ , and its corresponding embeddings sequence  $\mathbf{f} = (f_1, \dots, f_T)$ :

**Time Warping** is implemented using the `interpolate` function of `PyTorch` in nearest mode. A time warp factor  $W = 80$  is employed. A random centre  $C$  is selected from  $[W + 1, T - W]$ , and resultant warped size  $S$  is determined within  $[C - W, C + W + 1]$ . Consequently, the sequence warps at the segments  $x_{1:C-1}$  and  $x_{C:T}$  to size  $S$  and  $T - S$ , respectively.

**Time Masking** is executed on consecutive temporal steps of  $\mathbf{x}$ .  $N$  denotes the number of mask regions, limited to  $\min(10, \lfloor 0.0015 \times T \rfloor)$ . The maximum mask width  $M$  is set as  $\min(100, \lfloor \frac{0.15 \times T}{N} \rfloor)$ . Each mask  $i$  has a random width  $m_i$  up to  $M$  and starts at the time step  $\lambda(T - m_i)$ , where  $\lambda \sim \mathcal{U}(0, 1)$ .

**Embedding Masking** is performed successively along the vertical embedding dimension of  $\mathbf{f}$ . The masking stride is determined by a random value  $m_j$  ranging from 0 to 27, and the starting position of the embedding masking is  $\lambda(F - m_i)$ , where  $\lambda \sim \mathcal{U}(0, 1)$  and  $F$  is the embedding dimension. The embedding masking is applied twice independently to improve the robustness.

**Gaussian Noise** followed by standard normal distribution is added to  $\mathbf{f}$  with a probability of 0.25.

In the same batch, each audio sample is independently augmented with a probability of 0.9. All masking values are set to zero.

### 4. SPEECH SYNTHESIS WITH DISCRETE TOKENS

To investigate TTS performance among various discrete tokens, we conduct experiments on speech resynthesis tasks as it eliminates possible interference from the acoustic model and serves as an upper bound of TTS systems. In this study, we utilize `CTX-vec2wav` proposed in `UniCATS` [15], which has demonstrated SOTA performance among vocoders operating on discrete tokens. Since discrete tokens primarily contain speech content, additional information, such as speaker identity or prosody, must be incorporated for resynthesis. In contrast to other approaches that utilize explicit speaker embedding or acoustic token prefixes, `CTX-vec2wav` employs contextual vocoding by utilizing mel-spectrogram as the prompt.

The `CTX-vec2wav` consists of a frontend encoding discrete tokens and a `HifiGAN` [21] generator. The semantic tokens are first projected and encoded by two `Conformer` [22] encoders with an auxiliary feature adapter in between. The auxiliary feature adapter, similar to the variance predictor in `FastSpeech2` [23], incorporates a three-dimensional prosody feature (pitch, energy, probability-of-voiced) to enhance audio quality. To condition on the mel-spectrogram prompt, the two `Conformer` encoders are equipped with an extra cross-attention layer following self-attention, which takes convolutional encoded mel-spectrogram as key and value. Then, the output from the second `Conformer` encoder is fed to `HifiGAN` to produce waveform.

## 5. EXPERIMENTS

### 5.1. ASR Experimental Setups

The efficacy of the proposed protocol in ASR is assessed on LibriSpeech [24], GigaSpeech [25], and AISHELL-1 [26] corpora. Performance metrics include Word Error Rate (WER) for English and Character Error Rate (CER) for Chinese, assessed on LibriSpeech’s test-clean and test-other sets, GigaSpeech’s development and test sets, and AISHELL-1’s development and test sets. Separate k-means models are trained for each data set on a 100-hour, randomly selected subset of speech.

In FBank-based experiments, SpecAugment [20] is applied during training for robustness. The input is 80-channel FBank features extracted over windows of 25ms strided by 10ms. The classification units are 500-class Byte Pair Encoding (BPE) [27] word pieces. In discrete tokens experiments, training incorporates discretized input augmentations as described in Sec. 3.2.

The neural Transducer architecture is adopted for ASR. Pruned RNN-T loss [28] is used as the training objective function, implemented within the k2 [29] framework<sup>1</sup>. The encoder employs a 6-stack Zipformer [30] with downsampling factors of (1,2,4,8,4,2). The label decoder employs a stateless decoder [31], which consists of an embedding layer followed by a 512-dim Conv1D layer. A convolution subsampling module with a stride of 2 is placed to reduce the frame rate to 50 Hz before being fed into the encoder. Overall, the model has 65.5M parameters. All models are trained with 4 NVIDIA V100 32GB GPUs.

### 5.2. Results on LibriSpeech

Table 1 shows the ASR performance of models trained on LibriSpeech 100h using different types of discrete tokens. It indicates that systems trained on tokens from WavLM and HuBERT yield improvements of 17% and 28% over those trained on FBank features for test-clean and test-other, respectively. The discrete tokens derived from EnCodec and vq-wav2vec perform worse than FBank features. This reveals the importance of SSL models for discrete token generation and the superiority of discrete tokens from WavLM and HuBERT models on low-resource data. We also report the results from [13] to compare with the existing method. Our approach exhibits lower WERs and more substantial performance gains from self-supervised models.

**Table 1:** LibriSpeech-100h WERs(%) of FBank features and various discrete tokens.

Method	Feature	# Units	Bandwidth (kbps)	test	
				clean	other
Chang et al. (2023) [13]	FBank	-	256.00	8.30	22.20
	WavLM-Large	2000	0.55	5.90	12.80
Ours	FBank	-	256.00	6.27	16.67
	WavLM-Large	2000	0.55	5.22	11.85
	HuBERT-Large	2000	0.55	5.19	11.94
	EnCodec	1024 <sup>8</sup>	6.00	7.16	22.04
	vq-wav2vec	320 <sup>2</sup>	1.66	11.76	33.08

Table 2 presents the ASR performance of models trained on LibriSpeech 960h using different types of discrete tokens. Systems trained on tokens from WavLM and HuBERT show competitive performance against FBank-based systems. The results from [13] are also given for comparison. Our approach not only achieves lower

WERs but also narrows the performance gap with FBank-based systems. Similar to the observation on LibriSpeech 100h as shown in Table 1, discrete tokens from EnCodec and vq-wav2vec are consistently worse than those from HuBERT and WavLM models. This could be explained by the fact that the features of HuBERT and WavLM are extracted from the top Transformer layer, which contains more semantic information. It is also worth noting that the discrete tokens yield comparable WER results to the continuous FBank feature on the test-clean set. But in contrast, a relatively larger WER gap is observed on test-other. One possible reason is that the discrete tokens are good at extracting semantic information under clean conditions but still incompetent for complicated acoustic conditions.

**Table 2:** LibriSpeech-960h WERs (%) of FBank features and various discrete tokens.

Method	Feature	# Units	test	
			clean	other
Chang et al. (2023) [13]	FBank	-	2.60	6.20
	WavLM-Large	2000	3.00	7.00
Ours	FBank	-	2.23	5.15
	WavLM-Large	2000	2.29	5.50
	HuBERT-Large	2000	2.29	5.73
	EnCodec	1024 <sup>8</sup>	2.57	6.81
	vq-wav2vec	320 <sup>2</sup>	3.19	9.44

### 5.3. Results on GigaSpeech 1000h and AISHELL-1

Previous experiments were conducted on LibriSpeech, which matches well with the SSL training data. To get a whole picture of the ASR performance with discrete tokens, two additional datasets are selected for ASR training and evaluation. The first data set is GigaSpeech 1000h (GigaSpeech-M), which aims to evaluate the ASR performance on a more diverse, noisy, and challenging dataset. The other test set is AISHELL-1, a Mandarin speech corpus recorded in clean condition, to assess the adaptability of discrete tokens across languages. Table 3 shows the ASR performance of models trained on GigaSpeech 1000h and AISHELL-1. In GigaSpeech, discrete tokens extracted from WavLM are consistently superior to those of HuBERT. Discrete tokens of WavLM lag behind the continuous FBank feature by a relative WER degradation of 7%. Despite that, the WER results achieved by the discrete tokens are quite impressive and competitive compared to previous studies [25, 32] on the GigaSpeech-M. In contrast, in the Mandarin AISHELL-1 dataset, the discrete tokens present much worse WER results than the FBank feature. Recalling that the SSL models used for extracting discrete tokens are trained in English corpus, the generalization of current discrete tokens across languages is yet to be improved.

**Table 3:** GigaSpeech-1000h WERs(%) and AISHELL-1 CERs(%) of FBank features and various discrete tokens.

Dataset	Feature	# Units	WER	
			dev	test
GigaSpeech	FBank	-	12.24	12.19
	WavLM-Large	2000	13.20	13.05
	HuBERT-Large	2000	14.45	14.79
	FBank	-	4.27	4.49
AISHELL-1	WavLM-Large	2000	8.41	8.83

### 5.4. Ablation Study

Table 4 demonstrates the effectiveness of customized data augmentation techniques for discretized inputs. Although Gaussian noise doesn’t offer immediate benefits in LibriSpeech 100h, it helps enhance the stability of the convergence curve.

<sup>1</sup><https://github.com/k2-fsa/k2>

**Table 4:** Ablation study of augmentation policy on LibriSpeech 100h using discrete tokens from WavLM.

Augmentation Policy	test	
	clean	other
Without Augmentation	6.74	14.71
+ Time Warping	6.09	13.48
+ Time Masking	5.57	12.36
+ Embedding Masking	5.58	12.31
+ Frame Duplication	5.31	11.74
+ Gaussian Noise	5.24	11.83

Table 5 compares the impact of different initialization methods for the embedding layer on ASR performance, indicating the incorporation of a codebook along with an additional linear layer increases parameters without notably improving performance.

**Table 5:** Ablation study of embedding initialization for discrete tokens strategies on LibriSpeech 100h using discrete tokens from WavLM.

Embedding Method	test		# Parameters
	clean	other	
Random Initialized	5.24	11.83	65.7 M
Codebook Initialized + Linear Projection	5.28	11.71	67.7 M

## 5.5. TTS Experimental Setups

Speech resynthesis experiments are conducted to compare the speech synthesis ability between the mel-spectrogram feature and various discrete tokens. LibriTTS [33] containing 580 hours of speech from 2306 speakers is adopted. We first filter out utterances less than 5 seconds from the train-960 subset and use the remaining for training. For CTX-vec2wav [15], both Conformer encoders consist of 2 blocks, each with 2 attention heads and 184 attention dimensions. The mel-spectrogram encoder employs a 1D convolution with a kernel size of 5 and an output channel of 184. The HifiGAN generator has upsampled kernel size (16,10,8,4) for HuBERT, WavLM, and (16,10,4,4) for vq-wav2vec.

In training, we randomly divide each utterance into two segments. The first segment has a duration of 2-3 seconds, which is used to extract the mel-spectrogram and acts as the acoustic context. The remaining segment is utilized to extract discrete tokens and perform vocoding. The CTX-vec2wav are optimized using the same objective as HifiGAN, supplemented by an L1 loss from the auxiliary feature adaptor. To stabilize training, we also incorporate the multi-task warmup technique proposed in [14]. All models are trained with an Adam optimizer for 500k steps with the initial learning rate set to  $2 \times 10^{-4}$  and halved every 200k steps. A variant of Encodec, DAC [34], is also adopted for speech resynthesis. For the tokens from Encodec and DAC, we utilize the publicly released checkpoint to decode tokens back to the waveform without training on LibriTTS.

Also, we trained a HifiGAN vocoder with mel-spectrogram input, which is identical to the HifiGAN generator in the vq-wav2vec variant. The mel-spectrogram feature is extracted with a shift of 10ms and a frame length of 29.0625ms. Other configurations remain consistent with those mentioned earlier.

## 5.6. Resynthesis Results on LibriTTS 960h

To evaluate the performance of these models, we conduct resynthesis on a test set comprised of 500 randomly selected utterances<sup>2</sup> from

<sup>2</sup>[https://francisfy.github.io/universal\\_discrete\\_token.html](https://francisfy.github.io/universal_discrete_token.html)

the LibriTTS test-clean subset, which includes recordings from 37 different speakers. For the CTX-vec2wav models trained with HuBERT, WavLM, and vq-wav2vec tokens, we perform resynthesis on each utterance using a 3-second prompt from the same speaker. In the case of the HifiGAN vocoder with mel-spectrogram input, we directly perform vocoding. Finally, for the Encodec and DAC variants, we convert the discrete tokens to waveforms using the provided decoder.

The resynthesis results are assessed using MOS listening tests, where 10 participants rate naturalness and speaker similarity on a 1-5 scale. Speaker Encoder Cosine Similarity (SECS) is calculated using Resemblyzer<sup>3</sup> as an auxiliary metric for speaker similarity. Table 6 shows the results of subjective and objective evaluations for the speech resynthesis task with various discrete tokens and mel-spectrogram features. It can be seen that all discrete tokens, except Encodec, are able to generate high-quality audio and present similar MOS and SECS performance compared to mel-spectrogram features. It is worth noting that both Encodec and DAC are not fine-tuned on the LibriTTS data as other models. Interestingly, the DAC gives the best resynthesis performance, while the Encodec presents the worst performance. Given the intrinsic similarity of Encodec and DAC, there is a large potential for the neural audio codec model. Discrete tokens from vq-wav2vec outperform those from HuBERT and WavLM. Recalling its inferiority in the ASR task, one possible explanation is that vq-wav2vec contains more acoustic information. Discrete tokens from HuBERT and WavLM generate high-fidelity audio with comparable quality.

**Table 6:** Comparison of TTS performance on LibriTTS-960h among mel spectrogram and various discrete tokens.

Feature	# Units	Bandwidth (kbps)	MOS		SECS
			Naturalness	Similarity	
Ground-truth	-	-	4.48	4.18	0.843
Mel spectrogram	-	256.00	4.36	4.17	0.834
Encodec	1024 <sup>8</sup>	6.00	3.83	3.85	0.834
DAC	1024 <sup>8</sup>	4.00	4.41	4.30	0.841
vq-wav2vec	320 <sup>2</sup>	1.66	4.36	4.21	0.842
HuBERT Large	2000	0.55	4.26	4.18	0.833
WavLM Large	2000	0.55	4.18	4.18	0.836

## 6. CONCLUSIONS

With the goal of exploring the universality of speech discrete tokens across multiple speech tasks, this paper conducted a comprehensive study employing two representative speech processing tasks: speech recognition and speech synthesis. We investigate speech discrete tokens from four leading SSL models, including vq-wav2vec, encodec, HuBERT, and WavLM. Experimental results indicate that discrete tokens are competitive with FBank features in speech recognition and surpassed mel-spectrogram features in speech synthesis. These empirical findings suggest that universal discrete tokens have considerable potential across diverse speech-related tasks. It’s crucial to note that this study is preliminary. While it has contributed valuable insights through its comparative analysis of various speech pre-training models and tasks, there is still ample room for refinement. We aspire that this work may serve as a foundational stepping stone, bridging the representation of speech and text, and promoting research in the domain of cross-modal exploration.

<sup>3</sup><https://github.com/resemble-ai/Resemblyzer>

## 7. REFERENCES

- [1] Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al., “Self-supervised speech representation learning: A review,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, 2022.
- [2] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NeurIPS*, 2020.
- [3] Alexei Baevski, Steffen Schneider, and Michael Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in *Proc. ICLR*, 2020.
- [4] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhota, Ruslan Salakhutdinov, and Abdelrahman Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, 2021.
- [5] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al., “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, 2022.
- [6] Ziyang Ma, Zhisheng Zheng, Changli Tang, Yujin Wang, and Xie Chen, “MT4SSL: Boosting Self-Supervised Speech Representation Learning by Integrating Multiple Targets,” in *Proc. Interspeech*, Dublin, 2023.
- [7] Ziyang Ma, Zhisheng Zheng, Guanrou Yang, Yu Wang, Chao Zhang, and Xie Chen, “Pushing the limits of unsupervised unit discovery for SSL speech representation,” in *Proc. Interspeech*, Dublin, 2023.
- [8] Zalan Borsos, Raphael Marinier, Damien Vincent, Eugene Kharitonov, et al., “AudioLM: a language modeling approach to audio generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, 2023.
- [9] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi, “SoundStream: An end-to-end neural audio codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, 2021.
- [10] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi, “High fidelity neural audio compression,” in *arXiv preprint arXiv:2210.13438*, 2022.
- [11] Alexei Baevski, Michael Auli, and Abdelrahman Mohamed, “Effectiveness of self-supervised pre-training for speech recognition,” in *arXiv preprint arXiv:1911.03912*, 2019.
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, Minneapolis, 2019.
- [13] Xuankai Chang, Brian Yan, Yuya Fujita, Takashi Maekaku, and Shinji Watanabe, “Exploration of efficient end-to-end asr using discretized input from self-supervised learning,” in *Proc. Interspeech*, Dublin, 2023.
- [14] Chenpeng Du, Yiwei Guo, Xie Chen, and Kai Yu, “VQTTs: High-fidelity text-to-speech synthesis with self-supervised vq acoustic feature,” in *Proc. Interspeech*, Incheon, 2022.
- [15] Chenpeng Du, Yiwei Guo, Feiyu Shen, Zhijun Liu, Zheng Liang, Xie Chen, Shuai Wang, Hui Zhang, and Kai Yu, “UniCATS: A unified context-aware text-to-speech framework with contextual vq-diffusion and vocoding,” *CoRR*, vol. abs/2306.07547, 2023.
- [16] Tu Anh Nguyen, Wei-Ning Hsu, Antony D’Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, et al., “Expresso: A benchmark and analysis of discrete expressive speech resynthesis,” in *Proc. Interspeech*, Dublin, 2023.
- [17] Tianrui Wang, Long Zhou, Ziqiang Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei, “ViOLA: Unified codec language models for speech recognition, synthesis, and translation,” in *arXiv preprint arXiv:2305.16107*, 2023.
- [18] Jinyu Li et al., “Recent advances in end-to-end automatic speech recognition,” *APSIPA Transactions on Signal and Information Processing*, vol. 11, 2022.
- [19] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhota, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al., “SUPERB: Speech processing universal performance benchmark,” in *Proc. Interspeech*, Brno, 2021.
- [20] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proc. Interspeech*, Graz, 2019.
- [21] Jiaqi Su, Zeyu Jin, and Adam Finkelstein, “HiFi-GAN: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks,” in *Proc. NeurIPS*, 2020.
- [22] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, “Conformer: Convolution-augmented transformer for speech recognition,” in *Proc. Interspeech*, Shanghai, 2020.
- [23] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, “FastSpeech 2: Fast and high-quality end-to-end text to speech,” in *Proc. ICLR*, Vienna, 2021.
- [24] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “LibriSpeech: an asr corpus based on public domain audio books,” in *Proc. ICASSP*, South Brisbane, 2015.
- [25] Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al., “GigaSpeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio,” in *Proc. Interspeech*, Brno, 2021.
- [26] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, “AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline,” in *Proc. Oriental COCODSA*, Seoul, 2017.
- [27] Rico Sennrich, Barry Haddow, and Alexandra Birch, “Neural machine translation of rare words with subword units,” in *Proc. ACL*, Berlin, 2016.
- [28] Fangjun Kuang, Liyong Guo, Wei Kang, Long Lin, Ming-shuang Luo, Zengwei Yao, and Daniel Povey, “Pruned RNN-T for fast, memory-efficient asr training,” in *Proc. Interspeech*, Incheon, 2022.
- [29] Daniel Povey, Piotr Zelasko, and Sanjeev Khudanpur, “Speech recognition with next-generation kaldi (k2, lhotse, icefall),” in *Interspeech: tutorials*, 2021.
- [30] Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey, “Zipformer: A faster and better encoder for automatic speech recognition,” in *arXiv preprint arXiv:2310.11230*, 2023.
- [31] Mohammadreza Ghodsi, Xiaofeng Liu, James Apfel, Rodrigo Cabrera, and Eugene Weinstein, “Rnn-Transducer with stateless prediction network,” in *Proc. ICASSP*, Barcelona, 2020.
- [32] Xun Gong, Yu Wu, Jinyu Li, Shujie Liu, et al., “LongFNT: Long-form speech recognition with factorized neural Transducer,” in *Proc. ICASSP*, Rhodes Island, 2023.
- [33] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, et al., “LibriTTS: A corpus derived from LibriSpeech for text-to-speech,” in *arXiv preprint arXiv:1904.02882*, 2019.
- [34] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar, “High-fidelity audio compression with improved RVQGAN,” in *arXiv preprint arXiv:2306.06546*, 2023.