Interspeech 2024
1-5 September 2024, Kos, Greece

How Should We Extract Discrete Audio Tokens from Self-Supervised Models?

Pooneh Mousavi[1,2], *Jarod Duret*[3], *Salah Zaiem*[4], *Luca Della Libera*[1,2], *Artem Ploujnikov*[5,2], *Cem Subakan*[6,2,1], *Mirco Ravanelli*[1,2,5]

[1]Concordia University, Canada [2]Mila - Quebec AI Institute, Canada [3]Avignon Université, France
[4]Telecom Paris, France [5]Université de Montréal, Canada [6]Université Laval, Canada

pooneh.mousavi@mail.concordia.ca

Abstract

Discrete audio tokens have recently gained attention for their potential to bridge the gap between audio and language processing. Ideal audio tokens must preserve content, paralinguistic elements, speaker identity, and many other audio details. Current audio tokenization methods fall into two categories: Semantic tokens, acquired through quantization of Self-Supervised Learning (SSL) models, and Neural compression-based tokens (codecs). Although previous studies have benchmarked codec models to identify optimal configurations, the ideal setup for quantizing pretrained SSL models remains unclear.

This paper explores the optimal configuration of semantic tokens across discriminative and generative tasks. We propose a scalable solution to train a universal vocoder across multiple SSL layers. Furthermore, an attention mechanism is employed to identify task-specific influential layers, enhancing the adaptability and performance of semantic tokens in diverse audio applications.

Index Terms: discrete audio token, semantic token, representation learning, speech processing.

1. Introduction

Learning effective, efficient, and robust representations is a core problem in modern audio and speech processing systems [1]. Over the past few years, continuous representations learned by large self-supervised models such as Wav2Vec2 [2], WavLM [3], and HuBERT [4] have achieved unprecedented performance. A recent research trend consists of learning discrete audio representations instead of continuous ones, resulting in what is known as *audio tokens*. These discrete tokens offer several potential advantages. Firstly, they facilitate the development of audio language models (LMs) [5–10] and the creation of multi-modal large language models [11], which can emit audio, text, and visual tokens. Additionally, their compression potential can contribute to efficient data transmission and storage. Discrete tokens also enable us to address audio generation tasks such as speech enhancement and synthesis using classification methods, instead of relying on complex high-dimensional regression models.

Following the terminology from [5, 12], audio tokenization techniques can be broadly categorized into Compression-based (codecs) tokens and Semantic tokens. Compression-based tokens [13–16] utilize encoder-decoder architectures coupled with Residual Vector Quantization (RVQ) [13]. They are explicitly trained to accurately reconstruct the original audio, making them particularly suitable for audio generation tasks. Semantic tokens [17–19], on the other hand, are generated through clustering or quantization of the layers of Self-Supervised Learning (SSL) models [2–4]. Often, this involves selecting a layer from the pretrained SSL model and clustering its representations, typically with the k-means algorithm. Semantic tokens primarily capture coarse information such as phonetic, semantics, and syntactic details. Since they are not explicitly trained to achieve accurate waveform reconstruction, it is more natural to use them in discriminative tasks like Automatic Speech Recognition (ASR). Recent research, however, has shown that semantic tokens can be effective for generative tasks as well [20, 21]. Additionally, semantic tokens have been used in a hybrid tokenizer [12, 22]. This hybrid approach combines semantic and compression-based tokens, separating content information in the initial layer while preserving paralinguistic details in subsequent layers. A similar strategy has been widely adopted in audio LLMs [5–7]. Nevertheless, the most effective setting for extracting semantic tokens remains largely unclear. Recent studies have primarily focused on ASR and Speech Translation [23–25], without considering a broader range of discriminative and generative tasks.

This paper addresses this gap by evaluating the effects of different heuristics required to derive semantic tokens for several discriminative and generative tasks, such as speech recognition, speaker recognition, emotion classification, speech enhancement, and text-to-speech. We investigate various crucial aspects, including the impact of the number of clusters and the selection of the intermediate layer of the SSL model to discretize. The latter factor turned out to be crucial and task-dependent, as early layers capture low-level information and higher layers encode content and semantic nuances. Common strategies include using the middle layer [17, 20] or leveraging the last layer [25]. Instead of relying on partial information only, we introduced a novel technique based on an informed layer selection mechanism. We propose to cluster all layers and inject their information into the acoustic models using learnable attention weights. This approach significantly boosts performance while also providing valuable insights into the importance of each layer.

Since there is no built-in decoder in semantic tokens, a vocoder model for converting the semantic tokens into audio must be trained [26, 27]. Training such a vocoder is computationally demanding, making it highly impractical to train a separate vocoder for each layer or combination of layers. To address this challenge, we propose a novel scalable vocoder capable of operating with various layer combinations at no additional cost. This is achieved through a layer dropout training scheme, inspired by the bitrate scalability mechanism used in SoundStream [13]. Interestingly, our results show that the scalable vocoder outperforms all vocoders trained on every specific layer. Finally, for a comprehensive comparison, we provide experimental evidence using both in-domain and out-of-domain datasets for training k-means. For reproducibility and to encour-

10.21437/Interspeech.2024-2135

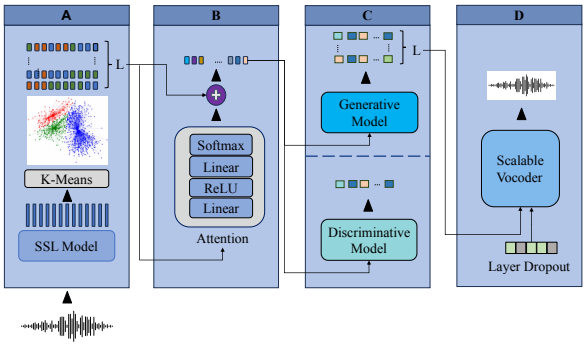

Figure 1: *The proposed method for audio token extraction from SSL models: (A) k-means discretizes the continuous representations of each layer, (B) an attention mechanism merges the discrete layer representations, (C) the mixed representations train acoustic models for discriminative and generative tasks, (D) our scalable vocoder generates waveforms (if needed).*

age further research, we release the code, built on the popular SpeechBrain [28] toolkit, and pretrained models publicly[1].

2. Model Design

The proposed architecture, illustrated in Fig. 1, consists of four components a) Tokenizer, b) Informed Layer Selector, c) Acoustic Model, and d) Scalable Vocoder. The following subsections will describe each module.

2.1. Tokenizer

For quantization, we cluster five layers taken from two pretrained SSL models using the k-means algorithm independently for each layer. We consider two widely-used models: WavLM-large[2] and HuBERT-large[3], both having 24 layers. We choose two layers from the lower part (3, 7) to capture fine-grained information, the middle layer (12), and two layers from the higher part (18, 23) for encoding content and meaning. This selection is based on observation from prior research [3, 29] which studied the contribution patterns of different layers across various tasks. As a result, this set of discrete hierarchical tokens captures rich information from the original audio signal. Each of the K clusters is assigned a unique index. Additionally, we store the continuous coordinates of each centroid for studying the effect of initializing input embeddings in downstream acoustic models (Sec. 4.4). The outcome of this tokenization process is a tensor $\mathbf{d}$ of shape $B \times T \times n_l$, where B represents the batch size, T is the sequence length, and n_l is the number of discretized layers.

2.2. Informed Layer Selector

As evident from the SSL literature [29–31], the choice of the layer within the SSL model significantly influences the performance of the downstream task of interest. This decision is equally critical for semantic tokens. Unlike prior methods that rely on heuristic layer selection [17,20,25], we integrate the information from our hierarchical multi-layer audio tokens with an attention mechanism. The attention mechanisms comprise a straightforward multi-layer perceptron (MLP) fed by the embeddings of the audio tokens from each layer. The MLP generates a score for each selected layer, that is normalized by a softmax function as shown in the following equations:

$$z_{l,t} = f\big(emb(d_{l,t})\big) \tag{1}$$

$$a_{l,t} = \frac{exp(z_{l,t})}{\sum_{k=1}^{nl} exp(z_{k,t})}, \quad h_t = \sum_l a_{l,t} z_{l,t}, \tag{2}$$

where, $z_{l,t}$ represents the score assigned to layer l at time t by the MLP function f. The variable emb refers to the lookup table that assigns embeddings to discrete tokens in d_l. The variable $a_{l,t}$ denotes the attention assigned to layer l at time t, and lastly h_t is the representation that is fed to the downstream MLP model. Note that we learn different layer combinations at each time-step, making this mechanism particularly effective.

This simple yet effective approach offers several advantages. Firstly, it enhances flexibility by reducing reliance on heuristic layer selection. The model can now dynamically capture information from different layers for each task. Additionally, as shown in Sec.4, this mechanism yields performance improvements when compared with models utilizing information from a single SSL layer. Lastly, the informed layer selections enhance interpretability, enabling us to analyze the learned weights and understand the relative importance of each layer for each downstream task.

2.3. Acoustic Model

The mixed representations are fed to a neural model trained to address various downstream tasks[4]. While previous studies [23–25] have primarily focused on a few discriminative tasks, we aim to provide evidence across a diverse range of speech applications, considering both discriminative and generative tasks. We consider ASR, speaker identification, and emotion recognition as discriminative tasks. For generative tasks, we focus on text-to-speech and speech enhancement. The details for each task are reported in Sec. 3.

2.4. Scalable Vocoder

Although SSL models such as Wav2vec2, HuBERT, and WavLM are not designed for accurate waveform reconstruction, we can potentially adapt them for generative tasks by training a vocoder on top of their representations. The dominant approach involves training a separate vocoder for each possible layer combination. However, this approach is impractical and computationally demanding since each downstream task may require a different set of layers. In this work, we propose a universal and scalable vocoder capable of accommodating various layer combinations. To train such a model, we modify HiFi-GAN [16] to accept a variable number of multi-layer discrete tokens as input. We introduce a layer dropout mechanism, similar to structured dropout [32]. For each input example, we randomly sample k layers from the range $[1, n_l]$, as shown in the following equations:

$$\mathbf{d}_S \sim \text{Sample}(\mathbf{d}, k), \quad \mathbf{o} = V(\mathbf{d}_S), \tag{3}$$

where 'Sample($\cdot$)' randomly selects k layers from the discrete representations $\mathbf{d}$, and V represents the vocoder function that outputs the waveform $\mathbf{o}$. Layers are combined with an attention mechanism that assigns weights to different layers and

[1] github.com/speechbrain/benchmarks/tree/DASB
[2] huggingface.co/microsoft/wavlm-large
[3] huggingface.co/facebook/hubert-large-ll60k

[4] We train the attention mechanism, embeddings, and the acoustic models jointly.

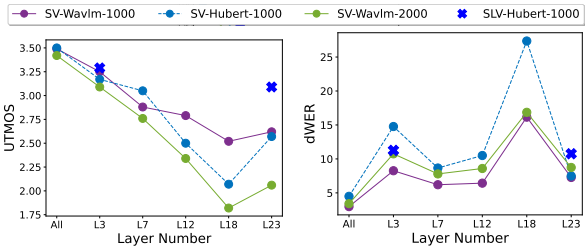

Figure 2: *Performance of the Scalable Vocoder (SV) at different layers compared to a Single-Layer Vocoder (SLV). Vocoders and tokenizers are trained using the LJSpeech dataset with 1000 and 2000 centroids.*

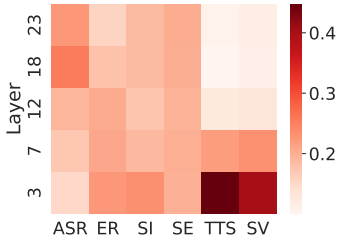

Figure 3: *Attention analysis across various tasks and layers of the discrete WavLM model with in-domain tokenizers.*

ensures that the dimensionality of the embeddings remains consistent regardless of the number of layers. The model is trained to decode audio by considering all possible combinations of layers. During inference, the desired combination of layers can be selected. In addition to its flexibility, this vocoder has demonstrated superior performance compared to vocoders trained on single layers, as we will show in Sec. 4.

3. Experiments

The tasks in our experiments are divided into two groups: Discriminative tasks involving transcription and classification, and generative tasks producing audio. For the downstream architecture choices and training procedures, we follow the best-performing approaches for classic continuous self-supervised representations [30]. We employ 1000 centroids across all tasks, except for ASR and emotion recognition, where we adopt 2000 centroids based on insights from prior research on ASR with discrete representations [25]. The effect of this selection is probed in Sec. 4.3.

3.1. Discriminative Tasks

Automatic Speech Recognition (ASR): We consider two CTC-based speech recognition tasks. The first one is English ASR using Librispeech *train-clean-100* for training and *test-clean*, *test-other* for testing. The second one uses French data coming from the CommonVoice (CV) 16.1 Corpus [33]. We select 100 hours for training, keeping the original validation and test sets. We use two layers of BiLSTM as a downstream head. The evaluation metric is the Word Error Rate (WER).

Speaker Identification (SID): We employ an ECAPA-TDNN model [34] to determine the speaker identity of each utterance. The widely used VoxCeleb1 [35] is adopted, and the evaluation metric is accuracy (ACC).

Emotion Recognition (ER): We use ECAPA-TDNN for emotion recognition [36] on the IEMOCAP dataset. The task consists of predicting one of the four considered classes: *happy*, *sad*, *angry*, and *neutral*. The evaluation metric is accuracy (ACC).

3.2. Generative Tasks

Speech Enhancement (SE): We utilize a non-autoregressive transformer encoder [37], which consists of 6 layers, 4 attention heads, a model dimension of 256, and a feed-forward layer dimension of 2048. Input tokens are extracted from the noisy signal, and target tokens from the clean one. Training is conducted end-to-end using cross-entropy loss. Noisy samples are generated by mixing clean samples from LJSpeech [38] with noise from WHAM! [39]. The signal-to-noise ratios (SNRs) are uniformly distributed between 0 and 5 dB. Due to the misalignment of the vocoder's output with the target at the sample level, metrics like Si-SNR can be degraded. Therefore, we use the deep noise suppression mean opinion score (DNSMOS) [40] for the speech quality metric, following a previous study [20]. Intelligibility is evaluated through the differential word error rate (dWER) [41], which measures the WER between the transcribed enhanced signal and the transcribed target signal. Transcriptions are obtained using the small version of Whisper [42].

Text-to-Speech (TTS): We train an end-to-end autoregressive Transformer [37] with 6 layers in the encoder, 12 layers in the decoder, 4 attention heads, a model dimension of 512, and a feed-forward layer in 2048. To facilitate convergence, we employ guided attention [43]. The model takes text embeddings as its input and generates the audio tokens for each considered layer. We utilize a shared transformer decoder, where each tokenizer head has its own learned embedding, and there is a distinct final linear layer for each token. We train all models on the LJSpeech dataset [38]. For assessing speech quality, we use UTMOS [44] to estimate human quality ratings. To evaluate fidelity to the text, we assess generated samples using the WER computed with the small version of Whisper [42].

4. Results

4.1. Scalable Vocoder

Our results cover findings from two distinct setups: 1) a scalable vocoder trained across five layers, and 2) a vocoder trained on a single layer. In both setups, the tokenizers and the vocoders are trained with LJSpeech (in-domain condition). In the first scenario, models are trained with HuBERT discrete tokens and WavLM discrete tokens, each with the number of clusters set to 1000. To further explore the influence of k-means cluster size on speech quality, we introduce an additional model with the number of clusters set to 2000. In the second setup, we focus on models trained specifically on a single layer (3 or 23) using HuBERT discrete tokens and in-domain tokenizer. This experiment aims to compare the performance of the scalable vocoder against the vocoder trained on a single layer.

The results are summarized in Fig. 2. WavLM combined with an in-domain tokenizer achieves higher UTMOS and lower dWER scores across all setups. About the impact of the number of clusters, our experiment shows that setting k to 2000 degrades the quality of synthesized speech. Finally, both models trained on a single layer are outperformed on both evaluation metrics by the one trained on five layers, confirming the benefits of the scalable approach. Lastly, we explore an out-of-domain

Table 1: *Assessing the impact of the number of clusters and embedding initialization on discrete WavLM-Large across different tasks.*

Setting	ASR (EN)	ASR (FR)	SID	ER	SE		TTS	
	WER ↓	*WER* ↓	*ACC* ↑	*ACC* ↑	*DNSMOS*↑	*dWER*↓	*UTMOS*↑	*WER* ↓
Effect of Number Of Clusters								
1000	7.15	34.61	79.0	61.8	3.93	6.75	3.65	5.76
2000	6.96	32.94	79.5	67.2	3.93	6.58	3.55	5.62
Effect of Embedding Initialization								
Random	6.96	32.94	81.0	67.2	3.93	6.75	3.65	5.76
PreTrained & finetune	8.93	35.81	77.5	63.9	3.93	6.82	3.64	6.62
PreTrained & freeze	9.26	35.12	73.1	67.0	3.93	6.98	3.66	6.42

Table 2: *Out-of-domain and in-domain performance of discrete HuBERT and WavLM models across the downstream tasks.*

SSL Model	Tokenizer	ASR (EN)	ASR (FR)	SID	ER	SE		TTS		Vocoder	
		WER ↓	*WER* ↓	*ACC* ↑	*ACC* ↑	*DNSMOS*↑	*dWER*↓	*UTMOS*↑	*WER* ↓	*UTMOS*↑	*dWER*↓
HuBERT Large [4]	In-Domain	7.89	38.29	67.2	64.5	3.98	17.64	3.61	6.46	3.50	4.49
	Out-Of-Domain	N/A	39.50	67.8	61.7	3.95	15.92	3.54	5.45	3.48	2.92
WavLM Large [3]	In-Domain	6.96	32.94	81.0	67.2	3.93	6.75	3.65	5.76	3.49	2.98
	Out-Of-Domain	N/A	36.25	79.0	61.9	3.96	6.49	3.61	5.73	3.68	2.95

scenario where the tokenizers are trained on LibriSpeech and the vocoders are on LJSpeech. As shown in Table 2 (last column), we do not observe any significant performance degradation when using the scalable vocoder in an out-of-domain condition.

4.2. Layer Analysis

Fig. 3 depicts the average weights assigned to different layers in the WavLM model across various downstream tasks on the test dataset. In both TTS and the scalable vocoder, lower levels get greater importance as they prioritize effective reconstruction. Conversely, for ASR, the upper layers become more crucial in capturing the semantic aspects of spoken utterances. In the case of ER and SID, the third layer receives the highest weight. Our findings align with the observed pattern in continuous representations [45]. For SE, all layers are equally weighted, indicating the necessity of all hierarchical levels to achieve optimal audio quality while preserving the semantic content of the input.

4.3. Effect of Number of Clusters

We train k-means models with both 1000 and 2000 centroids and examine the impact of the number of clusters across different tasks, as illustrated in Table 1. In Generative tasks, TTS and SE, no significant differences are observed between models trained with 1000 and 2000 clusters. However, for ASR in both English and French, as well as ER, models with a higher number of clusters outperform those with fewer clusters. In the case of SID, the model trained with 1000 clusters exhibits comparable accuracy to the model with 2000 centroids. As expected, the ideal number of clusters is task-dependent. For multi-modal LLMs where a single set of tokens is desired to solve multiple tasks, we recommend a cluster count between 1000 and 2000.

4.4. Effect of Embedding Initialization

We study various configurations for initializing the embedding layers of audio tokens (Table 1). Three options are considered: 1) Random initialization of the embedding layers, 2) Initialization of the embedding layer with the corresponding centroid's embedding, while freezing the layer, and 3) Initialization of the embedding layer with the corresponding centroid's embedding, without freezing the layer. Across all tasks, there is no advantage observed in initializing the embedding with pretrained centroid embeddings, and random initialization consistently outperforms it in all scenarios. However, discriminative tasks show greater benefits from random initialization, while generative tasks exhibit comparable performance across all three settings. This observation eliminates the need for having the same embedding size as the SSL models, allowing the choice of a smaller and more efficient embeddings.

4.5. Out-of-Distribution Generalization

To evaluate the robustness of discrete representations under distribution shifts, we train tokenizers on both in-domain and out-of-domain datasets (Table 2). In discriminative tasks, k-means models are trained using the same dataset employed for training acoustic models. In the out-of-domain scenarios, k-means models are trained on *train-clean-100*, *train-clean-360*, and *train-other-500*. For generative tasks, k-means models are trained on LJSpeech for in-domain evaluation and *LibriSpeech-960h* for OOD evaluation, while both the acoustic model and vocoder are trained on LJSpeech. For all discriminative tasks, the in-domain tokenizer outperforms its OOD counterpart. Interestingly, in all generative tasks, training the model using the OOD tokenizer does not adversely affect performance and, in some instances, even improves the results. We speculate that this trend may arise because generative tasks primarily depend on tokens capturing low-level information, which tends to be more "universal" and transferable across different domains.

5. Conclusions

Discrete semantic tokens, derived from the quantization of SSL models, play an important role, providing *"pseudo-text"* valuable for training text-free speech language models and multimodal LLMs. We explore the optimal configuration of semantic tokens across discriminative and generative tasks. We introduce a novel technique involving an informed layer selection mechanism, utilizing learnable attention weights to integrate information from different SSL layers. This approach significantly enhances the performance and interpretability of the model. Furthermore, we propose a scalable solution for training a universal vocoder across multiple SSL layers, demonstrating its superiority over vocoders trained on specific layers. As future work, we plan to explore more diverse tasks and quantization methods, and the development of a multi-speaker vocoder.

6. References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Proc. of NeurIPS*, 2020.

[3] S. Chen *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

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

[5] Z. Borsos *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2523–2533, 2023.

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

[7] Q. Chen *et al.*, "Lauragpt: Listen, attend, understand, and regenerate audio with gpt," *arXiv preprint arXiv:2310.04673*, 2023.

[8] T. Wang *et al.*, "Viola: Unified codec language models for speech recognition, synthesis, and translation," *arXiv preprint arXiv:2305.16107*, 2023.

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

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

[11] Gemini-Team, "Gemini: A family of highly capable multimodal models," 2023.

[12] X. Zhang, D. Zhang, S. Li, Y. Zhou, and X. Qiu, "Speechtokenizer: Unified speech tokenizer for speech large language models," *arXiv preprint arXiv:2308.16692*, 2023.

[13] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[14] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

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

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

[17] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations," in *Proc. Interspeech*, 2021.

[18] D. Wells, H. Tang, and K. Richmond, "Phonetic analysis of self-supervised representations of english speech," in *Proc. of Interspeech*, 2022.

[19] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *Proc. of ASRU*, 2021.

[20] Z. Wang, X. Zhu, Z. Zhang, Y. Lv, N. Jiang, G. Zhao, and L. Xie, "SELM: Speech Enhancement Using Discrete Tokens and Language Model," *arXiv preprint arXiv:2312.09747*, 2023.

[21] Y. Yang, F. Shen, C. Du, Z. Ma, K. Yu, D. Povey, and X. Chen, "Towards universal speech discrete tokens: A case study for asr and tts," in *Proc. of ICASSP*, 2024.

[22] Z. Du, S. Zhang, K. Hu, and S. Zheng, "Funcodec: A fundamental, reproducible and integrable open-source toolkit for neural speech codec," *arXiv preprint arXiv:2309.07405*, 2023.

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

[24] D. Zhang, R. Ye, T. Ko, M. Wang, and Y. Zhou, "DUB: Discrete unit back-translation for speech translation," in *Proc. of ACL*, 2023.

[25] X. Chang, B. Yan, Y. Fujita, T. Maekaku, and S. Watanabe, "Exploration of efficient end-to-end asr using discretized input from self-supervised learning," *arXiv preprint arXiv:2305.18108*, 2023.

[26] J. Kong, J. Kim, and J. Bae, "Hifi-gan: generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. of NeurIPS*, 2020.

[27] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," *arXiv preprint arXiv:1811.0000*, 2018.

[28] T. P. Mirco Ravanelli *et al.*, "SpeechBrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[29] S.-w. Yang *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[30] S. Zaiem, Y. Kemiche, T. Parcollet, S. Essid, and M. Ravanelli, "Speech self-supervised representation benchmarking: Are we doing it right?" in *Proc. of Interspeech*, 2023.

[31] S. Zaiem, R. Algayres, T. Parcollet, E. Slim, and M. Ravanelli, "Fine-tuning strategies for faster inference using speech self-supervised models: A comparative study," in *Proc. of ICASSP*, 2023.

[32] N. Srivastava *et al.*, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[33] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proc. of LREC*, 2020.

[34] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Proc. of Interspeech*, 2020.

[35] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," 2017.

[36] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of NIPS*, 2017.

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

[39] G. Wichern *et al.*, "WHAM!: Extending speech separation to noisy environments," in *Proc. of Interspeech*, 2019.

[40] C. K. Reddy, V. Gopal, and R. Cutler, "DNSMOS P.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *Proc. of ICASSP*, 2022.

[41] Z.-Q. Wang *et al.*, "Sequential multi-frame neural beamforming for speech separation and enhancement," in *Proc. of SLT*, 2021.

[42] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[43] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. of ICASSP*, 2018.

[44] S. Takaaki *et al.*, "UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022," *Proc. of Interspeech*, 2022.

[45] S. Zaiem, Y. Kemiche, and T. Parcollet, "Speech Self-Supervised Representations Benchmarking: a Case for Larger Probing Heads," 7 2023.