Speech Discrete Tokens or Continuous Features? A Comparative Analysis for Spoken Language Understanding in SpeechLLMs

Dingdong Wang, Junan Li, Mingyu Cui, Dongchao Yang, Xueyuan Chen, Helen Meng
The Chinese University of Hong Kong
dingdongwang@link.cuhk.edu.hk

Abstract

With the rise of Speech Large Language Models (SpeechLLMs), two dominant approaches have emerged for speech processing: discrete tokens and continuous features. Each approach has demonstrated strong capabilities in audio-related processing tasks. However, the performance gap between these two paradigms has not been thoroughly explored. To address this gap, we present a fair comparison of self-supervised learning (SSL)-based discrete and continuous features under the same experimental settings. We evaluate their performance across six spoken language understanding-related tasks using both small and large-scale LLMs (Qwen1.5-0.5B and Llama3.1-8B). We further conduct in-depth analyses, including efficient comparison, SSL layer analysis, LLM layer analysis, and robustness comparison. Our findings reveal that continuous features generally outperform discrete tokens in various tasks. Each speech processing method exhibits distinct characteristics and patterns in how it learns and processes speech information. We hope our findings will provide valuable insights to advance spoken language understanding in SpeechLLMs.

1 Introduction

Learning speech representations that are robust and effective is a key challenge in modern speech processing systems. Recent advances in Speech Large Language Models (SpeechLLMs), especially systems such as GPT-4o, have captured significant attention in the speech domain (Ji et al., 2024; Arora et al., 2025; Cui et al., 2025; Peng et al., 2025; Yin et al., 2024; Caffagni et al., 2024). With the advancement of SpeechLLMs, two main types of speech representations are employed as inputs: continuous features and discrete speech tokens.

Using speech continuous features is an intuitive approach (Chu et al., 2023; Chen et al., 2023; Gong et al., 2023a; Wang et al., 2023a; Tang et al., 2023; Gong et al., 2023b; Wang et al., 2025) for integrating speech signals with large language models (LLMs). This type of representation is typically obtained from certain layers of a self-supervised learning (SSL) model or a speech encoder module. For example, in SLAM-ASR (Ma et al., 2024), continuous features are obtained from the final layer of the HuBERT model. In models like the Qwen-Audio series (Chu et al., 2023, 2024) and SALMONN (Tang et al., 2023), raw speech is transformed into high-dimensional embeddings using the Whisper encoder (Radford et al., 2023) and adapted for LLMs through an adapter module. These continuous features preserve the audio signal's richness and have shown strong performance in speech understanding (Ji et al., 2024). However, since autoregressive LLMs like GPT (OpenAI, 2023) and LLaMA (Dubey et al., 2024) are naturally designed to work with discrete text tokens, this has inspired researchers to explore representing speech as sequences of discrete tokens.

Recent interest in speech discrete tokens has grown rapidly, driven by their compatibility with large language models that operate on text tokens. Unlike continuous features, which require a speech adapter for modality alignment, discrete tokens can be directly integrated into the LLM's vocabulary. Several works have used discrete tokens in SpeechLLMs (Zhang et al., 2023a; Rubenstein et al., 2023; Wang et al., 2023b, 2024; Mitsui et al., 2024; Nguyen et al., 2024; Défossez et al., 2024). For instance, both AudioPaLM (Rubenstein et al., 2023) and SpeechGPT (Zhang et al., 2023a) use K-means clustering to discretize speech embeddings. More recent works, such as Mini-Omni series (Xie and Wu, 2024a,b) and GLM-4-Voice (Zeng et al., 2024), they explore tokenizer modules based on compression methods, leveraging encoder-decoder architectures with residual vector quantization (RVQ) (Zeghidour et al., 2021) to extract discrete tokens.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 24925–24936
November 4-9, 2025

Continuous-feature-based and discrete-token-based SpeechLLMs have each demonstrated promising performance across various speech-related tasks (Ji et al., 2024; Arora et al., 2025). However, despite their success, there has been no systematic and fair comparison between these two types of representation within the spoken language understanding (SLU) area. Intuitively, discrete tokens compress information through quantization, sacrificing precision for compactness, while continuous representations retain fine-grained acoustic and temporal details, which may affect the performance of discrete tokens when compared to continuous features. However, the extent and nature of the potential performance differences remain unclear. Moreover, in multimodal settings, it is still largely unknown whether LLMs interpret these two forms of speech input differently, and how such differences affect various downstream SLU-related task outcomes. Clarifying these distinctions is crucial for understanding each paradigm's limitations and guiding future advancements in architecture, tokenizer design, and training methodologies.

To address this gap, we present a comprehensive benchmark comparing continuous features and discrete tokens under consistent experimental conditions. In this work, we focus on self-supervised learning (SSL) models for speech feature extraction, leveraging K-means clustering to obtain discrete semantic tokens and using original embeddings as continuous features. We choose this option because the SSL framework offers a unified paradigm for direct and controlled comparison between continuous and discrete representations. Specifically, we select HuBERT-Large (Hsu et al., 2021) and WavLM-Large (Chen et al., 2022) as SSL models, and Qwen1.5-0.5B (Bai et al., 2023) and Llama3.1-8B (Dubey et al., 2024) as LLM decoders. We aim to investigate the performance differences between discrete tokens and continuous features in spoken language understanding, evaluating both representations across a range of tasks, including automatic speech recognition, phoneme recognition, speech translation intent classification, keyword spotting, and emotion recognition.

Beyond benchmark comparison, we further conduct detailed analysis from multiple perspectives, including efficiency, SSL and LLM layer behaviour, and robustness. Across a range of tasks, our results show that continuous features generally outperform discrete tokens, with the magnitude of this gap varying by task. Interestingly, discrete tokens perform better on phonetic-level tasks, indicating their strength in capturing subword-level structure. Our SSL and LLM layer analyses also reveal distinct learning patterns for each representation across the model hierarchy. Each representation paradigm also demonstrates unique strengths: discrete tokens excel in data compactness and training efficiency, while continuous features offer superior robustness in diverse conditions. Our key contributions are as follows:

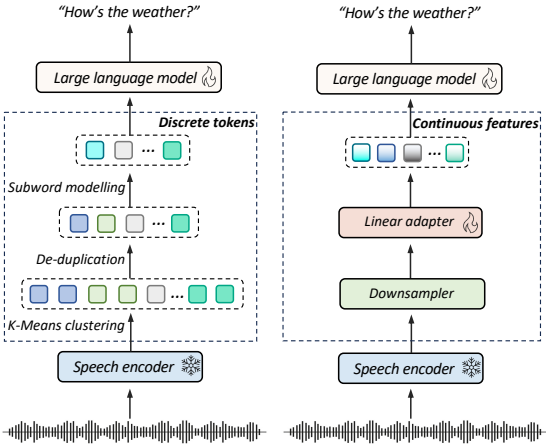


Figure 1: Architectures of two approaches for integrating speech into Large Language Models (LLMs). (Left) discrete token-based encoding. (Right) continuous feature processing.

- We present a comprehensive benchmark comparing speech continuous features and discrete tokens for spoken language understanding (SLU) under consistent experimental conditions.
- We conduct an in-depth analysis of both representations, examining efficiency, robustness, and learning dynamics across SSL and LLM layers.
- We identify the complementary strengths of each representation, offering practical guidance for future SpeechLLMs development.

2 Pipeline Design

To systematically compare the performance of continuous and discrete tokens in SLU related tasks, we adopt two widely used (Chu et al., 2024; Xu et al., 2025; Gong et al., 2023a; Wang et al., 2023a; Zhang et al., 2023a; Shon et al., 2024; Dekel and Fernandez, 2024) speech processing pipelines, as illustrated in Fig. 1: (a) a discrete token pipeline and (b) a continuous features pipeline.

2.1 Speech Discretization

In our approach, we utilize SSL models (HuBERT-Large (Hsu et al., 2021), WavLM-Large (Chen et al., 2022)) to generate discrete tokens. As shown in Fig. 1(a), we first generate a sequence of high-dimensional feature vectors $H = \{h_1, h_2, \ldots, h_T\}$ from the audio waveform using an SSL model. Second, we apply **K-Means Clustering** to obtain discrete tokens $Z = \{z_1, z_2, \ldots, z_T\}$. These tokens are represented as discrete tokens that can be processed like text tokens in NLP, making it possible to use traditional NLP techniques.

After clustering, the discrete token sequence contain redundant consecutive information. Thus we apply **De-duplication** to merge consecutive identical tokens to reduce redundancy, and finally use **Byte-Pair Encoding (BPE)** to enhance input tokens by combining frequent subsequences into shorter meta tokens $Z' = \{z'_1, z'_2, \ldots, z'_{T'}\}$, where $T' < T$. Practically, the length reduction ratio $\frac{T'}{T}$ typically ranges from 30% to 60%, depending on the K-means clustering granularity and the BPE vocabulary size.

2.2 Processing Continuous Features

For continuous features, we utilize the same speech encoders. As shown in Fig. 1(b), the processing pipeline involves two steps: (1) **Downsampling**: where the feature sequence $H = \{h_1, h_2, \ldots, h_T\}$ is reduced by concatenating every k consecutive frame, producing $Z^S = \{z_1^S, \ldots, z_N^S\}$ with $N = T/k$. This reduces computational complexity while preserving key information. (2) **Linear Adapter**: after downsampling, we use a single linear adapter layer to map the embeddings to the LLM's hidden size.

2.3 Instruction-Tuning with LLMs

We select six representative tasks related to spoken language understanding: Automatic Speech Recognition (ASR), Phoneme Recognition (PR), Keyword Spotting (KS), Emotion Recognition (ER), Spoken Intent Classification (IC), and Speech Translation (ST). For each task, we design specific prompts to guide the model in following instructions. Detailed prompt designs and task-related datasets information can be found in the appendix.

We perform instruction-tuning on the discrete and continuous inputs for these tasks using the Qwen1.5-0.5B (Bai et al., 2023) and Llama3.1-8B (Dubey et al., 2024) models as decoders. Our goal is to explore how different downstream tasks and speech representations (discrete tokens vs. continuous features) interact with LLMs of varying scales, revealing insights into their performance in spoken language understanding.

3 Experiments and Results

3.1 Experimental Setup

To investigate the performance of discrete tokens compared to continuous features across various spoken language understanding related tasks, we conduct experiments using a range of datasets: LibriSpeech (Panayotov et al., 2015), GigaSpeech M-size (Chen et al., 2021) and CHiME-4 (Vincent et al., 2016) for **ASR**, LibriSpeech 100-hour (Panayotov et al., 2015) for **PR**, Speech Commands-v2 (Warden, 2018) for **KS**, SLURP (Bastianelli et al., 2020) for **IC**, and GigaST (Ye et al., 2022) for **ST**. To evaluate paralinguistic emotion information, we also include **ER** using the IEMOCAP dataset (Busso et al., 2008).

For a fair comparison, we consistently extract features from the final layer of both HuBERT-Large (Hsu et al., 2021) and WavLM-Large (Chen et al., 2022) at a 16,000 Hz sampling rate for both feature types. For LLM training on Qwen1.5-0.5B, we perform full-parameter fine-tuning for both discrete and continuous methods. For LLaMA3.1-8B, we employ LoRA (Hu et al., 2021) for efficiency by injecting adapters (rank=8 and $\alpha = 16$) into the projection layers for keys and queries in all self-attention layers.

For the speech processing procedure of discrete token, we chose K-means with 2000 centroids and 6000 BPE vocabulary size for generating discrete tokens across all tasks. For continuous features, we apply a downsampling rate of $\alpha = 2$ and a single-layer linear adapter to project the embeddings to the LLM input space. All experiments were conducted under identical settings, with a learning rate of 1×10^{-5}, batch size of 32, using the AdamW optimizer with a weight decay of 10^{-2}. The parameter settings for each method are determined based on ablation studies in Sec. 3.3.

3.2 Main Results

In our experiments, we explore the performance of discrete tokens and continuous features in a range of tasks, comparing them across different LLM scales (Table 1). Overall, for both the Qwen1.5-

SSL model	*Token type*	ASR (WER↓)			PR (PER↓)	ST (BLEU↑)	KS (ACC↑)	IC (ACC↑)	ER (ACC↑)
		LS (clean|other)	GigaSpeech (test)	CHiME4 (test)	LS-100 (clean)	GigaST (En-Zh|En-De)	SC (test|val)	SLURP (test)	IEMOCAP (test)
Qwen 1.5-0.5B									
HuBERT	Discrete	4.56 / 9.79	19.40	13.35	9.69	22.75 / 20.14	93.70 / 93.85	57.04	38.65
WavLM	Discrete	4.72 / 10.45	16.34	12.94	9.64	24.62 / 21.22	92.87 / 92.45	59.96	37.98
HuBERT	Continuous	4.91 / 6.43	17.45	8.62	12.84	26.63 / 25.42	95.38 / 95.70	76.84	56.72
WavLM	Continuous	2.92 / 4.61	13.96	8.68	12.62	29.44 / 28.12	97.76 / 97.36	81.35	59.45
Llama 3.1-8B									
HuBERT	Discrete	2.56 / 6.49	12.86	10.56	7.85	26.64 / 25.13	96.75 / 96.69	63.44	39.84
WavLM	Discrete	2.96 / 7.48	13.35	9.13	**7.02**	28.62 / 26.87	97.92 / 98.17	66.96	36.12
HuBERT	Continuous	1.76 / 4.58	9.04	5.72	9.83	32.02 / 34.42	**99.74 / 98.59**	**86.84**	64.54
WavLM	Continuous	**1.65 / 4.22**	**8.86**	**5.43**	10.44	**35.17 / 37.20**	97.34 / 98.26	85.57	**65.87**

Table 1: Comparison benchmark of discrete tokens and continuous features on various tasks. Discrete tokens use K-means (2000 clusters) with BPE size 6000 for all tasks. For LibriSpeech (LS) datasets, we evaluate test-clean (clean) and test-other (other) set. SC stands for Speech Commands-v2 dataset.

0.5B (Bai et al., 2023) and LLaMA3.1-8B (Dubey et al., 2024), continuous features consistently outperform discrete tokens in most tasks. Notably, WavLM-Large's continuous features perform the best across Automatic Speech Recognition (ASR), Speech Translation (ST), and Emotion Recognition (ER) tasks on the LLaMA3.1-8B model, while HuBERT-Large's continuous features show the best performance in Keyword Spotting (KS) and Intent Classification (IC) tasks on the same LLM.

The performance gap between discrete tokens and continuous features tends to increase as the LLM model size grows. For example, in Speech Translation, the BLEU score of continuous features on LLaMA3.1-8B is notably higher than that of discrete tokens, showing the increasing advantage of continuous features with larger models. In certain tasks and datasets, such as CHiME-4 (a noisy background ASR dataset), discrete tokens show a decline in ASR performance, whereas continuous features maintain more stable performance. Additionally, in Emotion Recognition, as the LLM decoder size increases, discrete token performance remains consistently poor, with accuracy significantly lower than that of continuous features. For example, WavLM-Large's discrete tokens achieve an accuracy of 37.98% on Qwen1.5-0.5B and 36.12% on LLaMA3.1-8B, while the corresponding continuous features achieve 59.45% on Qwen1.5-0.5B and 65.87% on LLaMA3.1-8B model.

Interestingly, in the Phoneme Recognition (PR) task, discrete tokens outperform continuous features across all model scales. Specifically, WavLM-Large's discrete tokens achieve the best performance on LLaMA3.1-8B, with 7.02% phoneme error rate (PER) score. This improvement likely stems from the discrete token representation, which aligns more closely with the phoneme-level structure of speech, facilitating easier learning than continuous features.

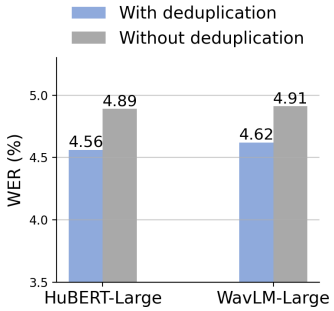
3.3 Ablation Study

Our ablation study is conducted using the LibriSpeech 960-hour dataset with the Qwen1.5-0.5B model, and evaluations are performed according to the test-clean subset.

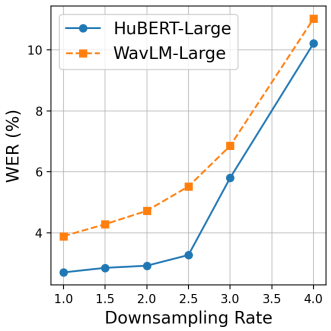
K-Meaning Clustering and BPE Size Settings. We investigate the effect of varying K-means clustering sizes and BPE (Byte Pair Encoding) vocabulary sizes on ASR performance. As shown in Fig. 2(a), increasing the number of centroids generally improves model performance. BPE-based subword modelling can further improve Word Error Rate (WER) results by reducing token sequence length while preserving key semantic information. Notably, the combination of $k = 2000$ centroids and 6000 BPE vocabulary size achieves a balanced trade-off between performance gains and computational efficiency. In our comparison study, we adopt this configuration across all datasets rather than tuning task-specific optimal settings, in order to ensure consistency in discrete token representa-

SSL model	*Manipulation*		WER ↓	
	K-Means	BPE size	test-clean	test-other
HuBERT-Large	k=1000	-	7.48	12.82
	k=2000	-	5.02	10.55
	k=3000	-	5.01	10.51
	k=2000	4000	4.99	10.78
	k=2000	6000	**4.56**	**9.79**
	k=2000	8000	5.04	11.20
WavLM-Large	k=1000	-	5.33	10.54
	k=2000	-	5.04	10.11
	k=3000	-	5.03	10.34
	k=2000	4000	4.88	10.62
	k=2000	6000	4.72	**10.45**
	k=2000	8000	**4.62**	10.82

(a)



(b)



(c)

Figure 2: Ablation studies: (a) Effect of K-means centroids and BPE size on WER; (b) Impact of deduplication method; (c) WER variation with different downsampling rates.

tions and enable fair cross-task comparison.

De-duplication Processing. As shown in Fig. 2(b), we examine the impact of de-duplication method applied during discretization. Our experiments show that using deduplication consistently improves WER performance, with WavLM-Large showing a reduction from 4.91% to 4.62%. The de-duplication process condenses consecutive identical tokens into a single token, reducing redundancy and improving information transmission efficiency.

Downsampling Settings. As shown in Fig. 2(c), we investigate the effect of downsampling rates on continuous features. As the rate increases, we observe a notable increase in WER for both HuBERT-Large and WavLM-Large. When the downsampling rate reaches 3, the WER starts to increase more sharply, suggesting a diminishing return in performance due to excessive downsampling. We select a downsampling rate of 2 as the optimal balance between efficiency and performance for all continuous feature settings.

4 Discussion

To provide a deeper understanding of each speech processing paradigm (continuous v.s. discrete), we conduct an in-depth analysis across several key areas, including efficiency comparisons, SSL layer contributions, LLM layer behaviours, and robustness in noisy conditions.

4.1 Efficiency Comparison

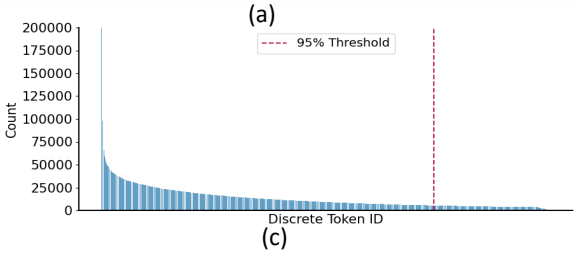
Data Efficiency For SpeechLLMs, the practical cost of a representation is its *bit-rate* $R = \log_2 V \cdot C \cdot R_s$, where V is vocabulary size, C the number of codebooks, and R_s the emission rate (codes s^{-1}). Fig. 3(a) compares the data size required to represent a T-second speech utterance using continuous SSL features and compressed discrete tokens. Continuous features from HuBERT-Large (same as WavLM-Large) require $32 \times 1024 \times 25 \times T$ bits, where 32 is the bit depth, 1024 is the feature dimensionality, and 25 is the frame rate. In contrast, after K-means compression, this is reduced to $13 \times 50 \times T$ bits. Further reductions can be achieved through de-duplication and BPE subword modelling, as calculated based on the LibriSpeech 100-hour dataset.

This analysis shows that continuous SSL features require orders-of-magnitude more bits than their discrete counterparts. However, bit-rate is only a proxy for Shannon information; 32-bit embeddings exhibit substantial numerical redundancy. The 99.9% reduction in bit-rate does not strictly correspond to the same level of information loss, but instead indicates an efficient compression of information for discrete tokens. As shown across all tasks, the discrete pipeline incurs only a modest accuracy drop, far smaller than the three-order-of-magnitude gain in bandwidth efficiency (see Table 1). Hence, discrete tokens offer an exceptionally bandwidth-efficient representation, making them attractive for on-device inference, low-bit-rate transmission, and large-scale pre-training where storage or I/O bandwidth is the principal bottleneck.

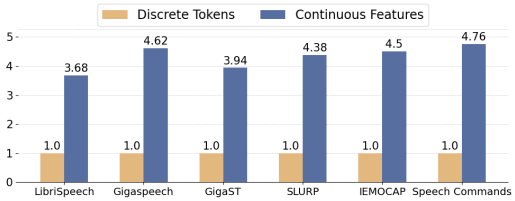
Training Efficiency We maintain consistent training settings (e.g. learning rate schedule) to ensure a fair comparison. Fig. 3(b) shows the total training time for convergence with discrete tokens and continuous features across datasets, and the training time for discrete tokens is normalized to 1 for comparison. Our experiments reveal that

Method	Data Size (bits)	Reduction Ratio
SSL features	$32 \times 1024 \times 25 \times T$	–
K-Means	$13 \times 50 \times T$	99.90%
+ De-duplication	$13 \times 36.47 \times T$	27.06%
+ BPE	$13 \times 20.24 \times T$	44.56%

(a)



(c)



(b)

SSL Model	WER (W/ U.T.)	WER (W/o U.T.)
HuBERT-Large	4.56	4.57
WavLM-Large	4.72	4.68

(d)

Figure 3: Efficiency analysis. (a) Data size (bit) comparison of a T-second utterance among: 1) SSL-based features using HuBERT-Large; 2) discrete tokens in 13-bit with 25 frames/sec. Reduction Ratio is calculated compared to the previous row based on LibriSpeech-100h. (b) Total training time until convergence for discrete tokens and continuous features, with discrete token training time normalized to 1 for all datasets. (c) Frequency distribution of discrete tokens with a codebook size of 6000, based on the GigaSpeech M-size corpus. The red line indicates the 95% cumulative frequency threshold. (d) WER comparison with and without under-trained tokens (U.T.)

discrete tokens converge in 4 to 5 epochs, while continuous features take 10 to 15 epochs. In each epoch, discrete tokens also train faster, as their shorter input sequences reduce the computational load, resulting in faster forward/backward passes and lighter token-embedding lookups. This results in a marked reduction in total training time, requiring only 21% to 27% of the time needed for continuous features. Discrete tokens' compact representation not only accelerates the training process but also reduces computational resource demands, making them a more training-efficient choice.

Utility Efficiency We conduct utility efficiency analysis for discrete tokens as shown in Fig. 3(c), which demonstrates the frequency distribution of 6 000-size discrete codebook on the GigaSpeech (M size) corpus. Roughly the least-frequent 20% of tokens together account for only 5% of all occurrences (this long-tailed pattern we also observe on other datasets). In this context, we follow prior work (Land and Bartolo, 2024) in labelling these infrequent tokens as under-trained tokens. These tokens exist in the LLM's vocabulary and are included during the training process, but are not sufficiently seen.

This imbalance exposes a key challenge of discrete tokenization: a large portion of the codebook is under-utilized, leading to under-trained tokens that the model rarely, if ever, encounters. Because the LLM codebook reserves fixed vocabulary slots for these under-utilized tokens, capacity that could support higher-utility tokens is wasted. The skewed distribution further injects noise into learning: the model is optimised mainly on the head of the distribution and receives little gradient signal for the tail, which may degrades generalisation—especially for speech segments whose acoustics are mapped to poorly trained codes.

As shown in Fig. 3(d), when we remove 10% of under-trained tokens during the LibriSpeech 960-hour training process, the WER on the LibriSpeech test-clean set remains roughly unchanged. This suggests that under-trained tokens have a limited effect on performance. These tokens occupy space in the codebook with little gain in performance. By contrast, continuous embeddings provide dense, frame-level representations that capture fine-grained acoustic and linguistic cues without suffering from sparse token utilization.

4.2 SSL Layer Analysis

We analyze the performance of discrete tokens and continuous speech embeddings from HuBERT-Large across various tasks. From the results in Fig. 4, we observe notable trends across different layers. For both discrete tokens and continuous features, we observe similar patterns in phonetic tasks such as ASR and phoneme recognition, where performance is strongest in the deeper layers. A similar trend is observed in semantic tasks like intent classification, suggesting that the model relies increasingly on deeper layers to capture semantic and content-related information. In contrast, emotion recognition exhibits a distinct pattern: continuous features achieve stronger performance in

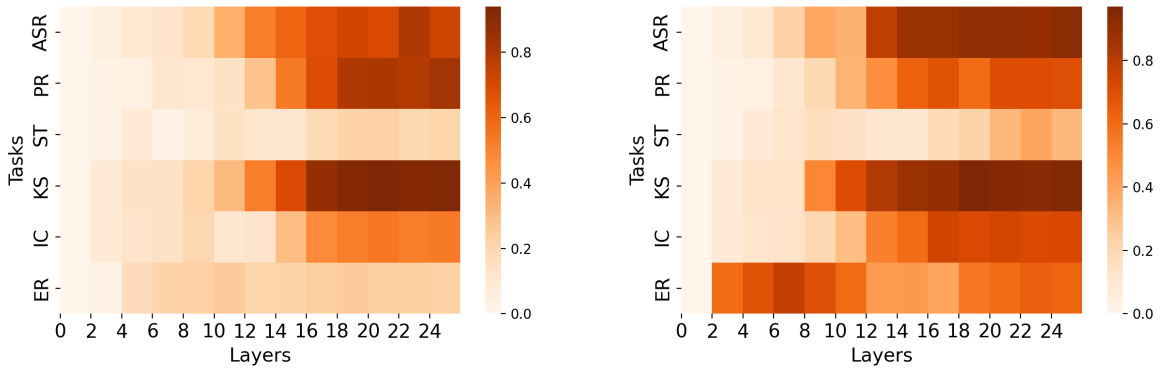


Figure 4: Layer analysis for each task on HuBERT-Large. (Left) Distribution of discrete token performance across each layer. (Right) Distribution of continuous features' performance across each layer.

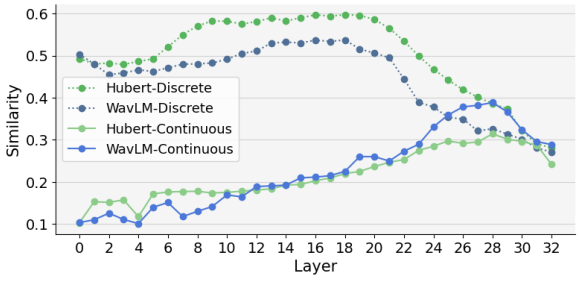


Figure 5: Alignments of features obtained from text-form and speech-form inputs.

Datasets	*Training strategy*	Performance
		test
Discrete Tokens		
SLURP	Baseline	57.04
	+SLURP ASR	66.36
	+Libri-100	67.18
CHiME4	Zero Shot	20.35
	Baseline	16.76
	+Libri-960	11.37
Continuous Features		
SLURP	Baseline	76.84
	+SLURP ASR	81.15
	+Libri-100	80.86
CHiME4	Zero Shot	10.81
	Baseline	8.62
	+Libri-960	6.59

Table 2: Robustness analysis with different training strategies: The baseline for SLURP utilizes SLURP data for Intent Classification (IC), while the baseline for CHiME4 employs CHiME4 data for ASR training. "+SLURP ASR" incorporates ASR pretraining on SLURP transcripts, and "+Libri-100" adds 100 hours of LibriSpeech data. For IC, performance is measured by accuracy, and for ASR, by word error rate (WER).

the shallower layers, particularly around layers 4 to 6. It indicates that the model can learn paralinguistic emotion information with the bottom layers. Notably, discrete tokens from K-means quantization fail to perform well in this task, indicating that this method is limited in capturing fine-grained emotional information. These findings offer preliminary insights into the functional roles of SSL layers, guiding future work on interpretability and task-specific SpeechLLMs.

4.3 LLM Layer Analysis

Fig. 5 illustrates the alignment of features extracted from speech and text inputs over the same words across different decoder layers of the Llama 3.1-8B model. The figure compares the maximum pairwise cosine similarity between speech input (discrete token or continuous feature representations) and text-form sequences across different LLM layers. Interestingly, the trends for discrete tokens and continuous features differ notably.

For discrete tokens, the similarity between speech and text sequences increases up to layer 22 before decreasing. This suggests that the model initially learns to map discrete speech with text in a manner more similar to text-based representations, but as the layers progress, this alignment starts to diminish. In contrast, for continuous features, the similarity continually rises until around layer 28, where it reaches its peak. This pattern may point to a difference in how the LLM model handles discrete versus continuous representations. With discrete tokens, speech and text seem to be processed as relatively similar modalities: token-level cues highlighting textual similarity at earlier layers. In contrast, continuous features appear to support a more gradual layer-by-layer alignment, suggesting a smoother transition between spoken and written forms throughout the network.

4.4 Robustness Comparison

We evaluate the robustness of discrete tokens and continuous features based on HuBERT-Large, within the Qwen-1.5-0.5B model on two noisy datasets. SLURP combines varied accents (e.g.,

Indian English) with a mix of close and distant microphone recordings, while CHiME-4 introduces diverse background environments. The complex acoustic contamination in these datasets challenges the model's noise robustness, especially for discrete tokens. According to our main results, discrete features consistently perform significantly worse than continuous features on both datasets (Table 1).

We hypothesize that the suboptimal performance of discrete tokens in intent classification on the SLURP dataset is due to their limited robustness in basic phonetic perception. To address this, as shown in Table 2, we first incorporate SLURP transcriptions into ASR training, followed by fine-tuning on intent classification. This approach leads to a substantial improvement in intent classification performance for discrete tokens (accuracy increases from 57.04% to 66.36%), with additional gains observed when incorporating the LibriSpeech 100-hour dataset. Similarly, in the CHiME-4 dataset, the zero-shot performance for discrete tokens is suboptimal when evaluated on a model trained solely with LibriSpeech 960-hour ASR data. The performance improved as we progressively incorporated CHiME-4 data along with additional ASR training data. In contrast, for continuous features, the performance improvements from additional training data augmentation are minimal, which indicates less sensitivity to training conditions and a more stable performance across varying data inputs.

5 Related Work

Following the terminology from prior works (Zhang et al., 2023b; Borsos et al., 2023), discrete tokens are categorized into two main types: Acoustic tokens and Semantic tokens. Acoustic tokens (e.g., WavTokenizer (Ji et al., 2025), Soundstream (Zeghidour et al., 2021), Encodec (Défossez et al., 2022) and ALMTokenizer (Yang et al., 2025)) are obtained using compression-based methods, which rely on encoder-decoder architectures with residual vector quantization (RVQ) (Zeghidour et al., 2021). Semantic tokens (Hsu et al., 2021; Chen et al., 2022) use clustering algorithms like K-means on SSL model features, with cluster indices serving as discrete representations. In the literature, both semantic and acoustic tokenizers are widely employed in SpeechLLMs. We adopt semantic tokens in our experiments because they enable a more direct and fair comparison with continuous features. Unlike acoustic tokens, which rely on complex compression methods, semantic tokens are derived directly from SSL models—the same source used for continuous features—allowing for a controlled comparison without additional confounding factors.

Most of the prior comparisons between discrete tokens and continuous features have been conducted within traditional speech processing settings. Works like (Nguyen et al., 2022; van Niekerk et al., 2022; Mousavi et al., 2024; Chang et al., 2024; Sharma et al., 2022) are grounded in conventional architectures rather than the multimodal framework of SpeechLLMs, making their findings less directly applicable to this context. Existing comparison analysis research on SpeechLLMs typically focuses on task-specific evaluations (Li et al., 2025; Xu et al., 2024), especially in ASR (Xu et al., 2024), without benchmarking across a broader range of spoken language understanding tasks, which is an essential step toward enabling effective human-machine interaction. Moreover, most related studies only focus on surface-level comparisons, lacking deeper analysis of internal mechanisms or cross-task implications. To address this gap, our study evaluates six SLU-related tasks across seven datasets using two SSL models and LLMs of different scales, providing a comprehensive analysis of the characteristics of discrete tokens and continuous features within the SpeechLLMs framework.

6 Conclusion

This work provides a comprehensive comparison benchmark of SSL-based discrete and continuous speech features across six spoken language understanding tasks: ASR, ST, KS, IC, ER, and PR. Our findings show that, regardless of whether using HuBERT-Large or WavLM-Large, continuous features generally outperform discrete tokens across various LLM backbone scales, except in phoneme recognition. Additionally, our experiments reveal distinct processing patterns for discrete tokens and continuous features in both SSL and LLM layers. Discrete tokens offer advantages in data and training efficiency but show potential for improvement in robustness and utility efficiency. We hope our findings provide valuable insights for the future development of SpeechLLMs.

7 Limitations

Due to limited computational resources, we were unable to incorporate a broader range of LLM backbones in our study. Expanding the number of models would have provided a more extensive comparison, potentially uncovering additional insights into the strengths and weaknesses of discrete tokens and continuous features. Furthermore, due to the extensive number of tasks and domains covered in our evaluation, we focused primarily on the LibriSpeech 960-hour dataset for our ablation study. While this allows for fair comparisons between models on a consistent dataset, it does not take into account the large-scale fine-tuning typically employed in state-of-the-art (SOTA) settings for each specific downstream task. This omission may limit the applicability of our findings to real-world settings, where task-specific fine-tuning is often crucial for achieving optimal performance.

References

Siddhant Arora, Kai-Wei Chang, Chung-Ming Chien, Yifan Peng, Haibin Wu, Yossi Adi, Emmanuel Dupoux, Hung-Yi Lee, Karen Livescu, and Shinji Watanabe. 2025. On the landscape of spoken language models: A comprehensive survey. *Preprint*, arXiv:2504.08528.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. Slurp: A spoken language understanding resource package. *arXiv preprint arXiv:2011.13205*.

Zalán Borsos, Raphaël Marinier, Vincent, and 1 others. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Sungbok Kazemzadeh, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42.

Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. 2024. The revolution of multimodal large language models: A survey. *Preprint*, arXiv:2402.12451.

Xuankai Chang, Brian Yan, Kwanghee Choi, Jee-Weon Jung, Yichen Lu, Soumi Maiti, Roshan Sharma, Jiatong Shi, Jinchuan Tian, Shinji Watanabe, and 1 others. 2024. Exploring speech recognition, translation, and understanding with discrete speech units: A comparative study. In *ICASSP*. IEEE.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Junbo Zhang, and 1 others. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*.

Qian Chen, Yunfei Chu, Zhifu Gao, Zerui Li, Kai Hu, Xiaohuan Zhou, Jin Xu, Ziyang Ma, Wen Wang, Siqi Zheng, and 1 others. 2023. Lauragpt: Listen, attend, understand, and regenerate audio with gpt. *arXiv preprint arXiv:2310.04673*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Jinyu Liu, and Kanda. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Yiwen Guo, and Irwin King. 2025. Recent advances in speech language models: A survey. *Preprint*, arXiv:2410.03751.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.

Avihu Dekel and Raul Fernandez. 2024. Exploring the benefits of tokenization of discrete acoustic units. *arXiv preprint arXiv:2406.05547*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue.

Yuan Gong, Alexander H Liu, Hongyin Luo, Leonid Karlinsky, and James Glass. 2023a. Joint audio and speech understanding. In *ASRU*, pages 1–8. IEEE.

Yuan Gong, Hongyin Luo, Alexander H Liu, Leonid Karlinsky, and James Glass. 2023b. Listen, think, and understand. *arXiv preprint arXiv:2305.10790*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Ruslan Lakhotia, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and 1 others. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, and 1 others. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, Ziang Zhang, Xiaoda Yang, Rongjie Huang, Yidi Jiang, Qian Chen, Siqi Zheng, and Zhou Zhao. 2025. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. *Preprint*, arXiv:2408.16532.

Sander Land and Max Bartolo. 2024. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11631–11646, Miami, Florida, USA. Association for Computational Linguistics.

Yixing Li, Ruobing Xie, Xingwu Sun, Yu Cheng, and Zhanhui Kang. 2025. Continuous speech tokenizer in text to speech. *Preprint*, arXiv:2410.17081.

Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Du, and 1 others. 2024. An embarrassingly simple approach for llm with strong asr capacity. *arXiv preprint arXiv:2402.08846*.

Kentaro Mitsui, Koh Mitsuda, Toshiaki Wakatsuki, Yukiya Hono, and Kei Sawada. 2024. Pslm: Parallel generation of text and speech with llms for low-latency spoken dialogue systems. *Preprint*, arXiv:2406.12428.

Pooneh Mousavi, Jarod Duret, Salah Zaiem, Luca Della Libera, Artem Ploujnikov, Cem Subakan, and Mirco Ravanelli. 2024. How should we extract discrete audio tokens from self-supervised models? *arXiv preprint arXiv:2406.10735*.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Mary Williamson, Gabriel Synnaeve, Juan Pino, Benoit Sagot, and Emmanuel Dupoux. 2024. Spirit lm: Interleaved spoken and written language model.

Tu Anh Nguyen, Benoit Sagot, and Emmanuel Dupoux. 2022. Are discrete units necessary for spoken language modeling? *IEEE Journal of Selected Topics in Signal Processing*, 16(6).

OpenAI. 2023. Gpt-4 technical report.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*. IEEE.

Jing Peng, Yucheng Wang, Yu Xi, Xu Li, Xizhuo Zhang, and Kai Yu. 2025. A survey on speech large language models. *Preprint*, arXiv:2410.18908.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *ICML*. PMLR.

Paul K Rubenstein, Chulayuth Asawaroengchai, Nguyen, and 1 others. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.

Roshan Sharma, Hira Dhamyal, Bhiksha Raj, and Rita Singh. 2022. Unifying the discrete and continuous emotion labels for speech emotion recognition.

Suwon Shon, Kwangyoun Kim, Yi-Te Hsu, Prashant Sridhar, Shinji Watanabe, and Karen Livescu. 2024. Discreteslu: A large language model with self-supervised discrete speech units for spoken language understanding. *arXiv preprint arXiv:2406.09345*.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*.

Benjamin van Niekerk, Marc-Andre Carbonneau, Julian Zaidi, Matthew Baas, Hugo Seute, and Herman Kamper. 2022. A comparison of discrete and soft speech units for improved voice conversion. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Emmanuel Vincent and 1 others. 2016. The 4th chime speech separation and recognition challenge. `http://spandh.dcs.shef.ac.uk/chime_challenge`. Last accessed on 1 August, 2018.

Chen Wang, Minpeng Liao, Zhongqiang Huang, Jinliang Lu, Junhong Wu, Yuchen Liu, Chengqing Zong, and Jiajun Zhang. 2023a. Blsp: Bootstrapping language-speech pre-training via behavior alignment of continuation writing. *arXiv preprint arXiv:2309.00916*.

Dingdong Wang, Jin Xu, Ruihang Chu, Zhifang Guo, Xiong Wang, Jincenzi Wu, Dongchao Yang, Shengpeng Ji, and Junyang Lin. 2025. Inserter: Speech instruction following with unsupervised interleaved pre-training. *Preprint*, arXiv:2503.02769.

Tianrui Wang, Long Zhou, and 1 others. 2023b. Viola: Unified codec language models for speech recognition, synthesis, and translation. *arXiv preprint arXiv:2305.16107*.

Xiaofei Wang, Manthan Thakker, and 1 others. 2024. Speechx: Neural codec language model as a versatile speech transformer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*.

Zhifei Xie and Changqiao Wu. 2024a. Mini-omni: Language models can hear, talk while thinking in streaming.

Zhifei Xie and Changqiao Wu. 2024b. Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities. *ArXiv*, abs/2410.11190.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025. Qwen2.5-omni technical report. *Preprint*, arXiv:2503.20215.

Yaoxun Xu, Shi-Xiong Zhang, Jianwei Yu, Zhiyong Wu, and Dong Yu. 2024. Comparing discrete and continuous space llms for speech recognition.

Dongchao Yang, Songxiang Liu, Haohan Guo, Jiankun Zhao, Yuanyuan Wang, Helin Wang, Zeqian Ju, Xubo Liu, Xueyuan Chen, Xu Tan, Xixin Wu, and Helen Meng. 2025. Almtokenizer: A low-bitrate and semantic-rich audio codec tokenizer for audio language modeling. *Preprint*, arXiv:2504.10344.

Rong Ye, Chengqi Zhao, Tom Ko, Chutong Meng, Tao Wang, Mingxuan Wang, and Jun Cao. 2022. Gigast: A 10,000-hour pseudo speech translation corpus. *arXiv preprint arXiv:2204.03939*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12).

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *Preprint*, arXiv:2412.02612.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023a. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2023b. Speechtokenizer: Unified speech tokenizer for speech large language models. *arXiv preprint arXiv:2308.16692*.

A Appendix

A.1 Task Definitions and Prompt Design

In our instruction-tuning experiments, we select six tasks that span a wide range of spoken language understanding capabilities. Each task is associated with a specific prompt that guides the model to produce the desired output. The details of each task and its corresponding prompt are as follows:

- **Automatic Speech Recognition (ASR)**: The task of converting spoken language into written text. ASR systems are essential for transforming speech into usable data for various applications. *Prompt*: *"Identify the text corresponding to the speech."*
- **Phoneme Recognition (PR)**: Involves identifying and classifying the smallest sound units, or phonemes, in speech. This task is fundamental for speech understanding, especially for speech-to-text systems. *Prompt*: *"Identify the phonemes corresponding to the speech."*
- **Keyword Spotting (KS)**: The goal is to detect specific predefined words or phrases within an audio stream. This task is important for applications like voice assistants, where the system must recognize specific commands. *Prompt*: *"Identify the keyword corresponding to the speech."*
- **Emotion Recognition (ER)**: Focuses on identifying and classifying the emotional state of a speaker based on their speech patterns. This task is crucial for understanding the speaker's emotional tone, often applied in customer service or mental health assessments. *Prompt*: *"Identify the emotion corresponding to the speech."*
- **Spoken Intent Classification (IC)**: The objective is to determine the speaker's intended action or meaning from their speech. This task is commonly used in conversational agents to understand user commands or queries. *Prompt*: *"Identify the intention corresponding to the speech."*
- **Speech Translation (ST)**: Converts spoken language from one language into another. This task is essential for real-time translation services, such as simultaneous interpretation or voice translation applications. *Prompt*:

"Translate the English speech into Chinese (or German) text."

Each task in our study is approached with tailored prompts, designed to help the model understand the task's specific requirements and produce accurate results. The prompt design ensures that each task is appropriately framed for instruction tuning, which plays a key role in adapting the model to each task's unique needs.

A.2 Datasets Used for Instruction Tuning

In our experiments, we utilize the following datasets for training and evaluating each task:

LibriSpeech (Panayotov et al., 2015): A large-scale corpus of approximately 1000 hours of 16kHz sampled English read speech, sourced from the LibriVox audiobook project. The dataset is carefully segmented and aligned, widely used for Automatic Speech Recognition (ASR) tasks.

GigaSpeech (Chen et al., 2021): A large-scale, multi-domain English speech recognition dataset containing approximately 10,000 hours of labelled audio. It covers a variety of accents and environments, making it suitable for training diverse speech recognition models.

CHiME-4 (Vincent et al., 2016): The CHiME-4 dataset is designed for evaluating automatic speech recognition (ASR) in noisy environments. It includes both real and simulated noisy speech data, recorded in four challenging acoustic settings: bus, café, pedestrian area, and street junction. The dataset features speech from the Wall Street Journal (WSJ0) corpus, captured using a 6-channel microphone array in real-world noise conditions.

IEMOCAP (Busso et al., 2008): A dataset containing 151 dialogue sessions between actors, annotated with 9 emotion labels (e.g., anger, excitement, fear, sadness). This dataset is used for emotion recognition tasks, providing rich samples of emotional speech.

SLURP (Bastianelli et al., 2020): A task-oriented spoken language understanding dataset containing dialogue data from 18 different domains, primarily used for intent classification tasks. Each speech sample is labelled with an intent category, which helps the model classify user intent from spoken commands.

GigaST (Chen et al., 2021): A large-scale speech translation corpus created by translating the transcribed text of GigaSpeech into German and Chinese. The training data is machine-translated, while the test set is manually translated, making it suitable for speech translation tasks.

Speech Commands (Warden, 2018): provided by Google, consists of 35 short spoken keywords with over 100,000 audio samples collected from 2,600 speakers. It is primarily used for keyword spotting tasks and includes labeled 1-second .wav files recorded in English.

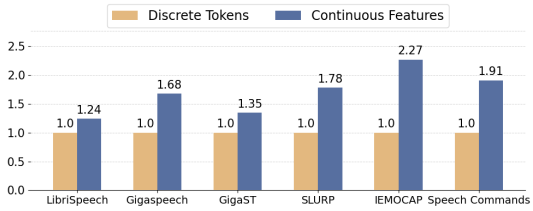


Figure 6: Training time per epoch using SSL continuous / discrete tokens. We normalize the training time of discrete tokens to unit 1 for all datasets

A.3 One Epoch Training Efficiency Comparison

Fig. 6 highlights the training time for one epoch across the datasets. After normalizing the training time of discrete tokens to 1, we observe that the time required for training with discrete tokens is between 33% and 67% less than that of continuous features. This further underscores the efficiency of discrete tokens, especially in terms of both time and resource usage.