SpeechPrompt: Prompting Speech Language Models for Speech Processing Tasks

Kai-Wei Chang, Haibin Wu, Yu-Kai Wang, Yuan-Kuei Wu, Hua Shen, Wei-Cheng Tseng, Iu-thing Kang, Shang-Wen Li, Hung-yi Lee

***Abstract*—Prompting has become a practical method for utilizing pre-trained language models (LMs). This approach offers several advantages. It allows an LM to adapt to new tasks with minimal training and parameter updates, thus achieving efficiency in both storage and computation. Additionally, prompting modifies only the LM's inputs and harnesses the generative capabilities of language models to address various downstream tasks in a unified manner. This significantly reduces the need for human labor in designing task-specific models. These advantages become even more evident as the number of tasks served by the LM scales up. Motivated by the strengths of prompting, we are the first to explore the potential of prompting speech LMs in the domain of speech processing. Recently, there has been a growing interest in converting speech into discrete units for language modeling. Our pioneer research demonstrates that these quantized speech units are highly versatile within our unified prompting framework. Not only can they serve as class labels, but they also contain rich phonetic information that can be re-synthesized back into speech signals for speech generation tasks. Specifically, we reformulate speech processing tasks into speech-to-unit generation tasks. As a result, we can seamlessly integrate tasks such as speech classification, sequence generation, and speech generation within a single, unified prompting framework. The experiment results show that the prompting method can achieve competitive performance compared to the strong fine-tuning method based on self-supervised learning models with a similar number of trainable parameters. The prompting method also shows promising results in the few-shot setting. Moreover, with the advanced speech LMs coming into the stage, the proposed prompting framework attains great potential.**

***Index Terms*—Prompting, speech language model, self-supervised learning, representation learning**

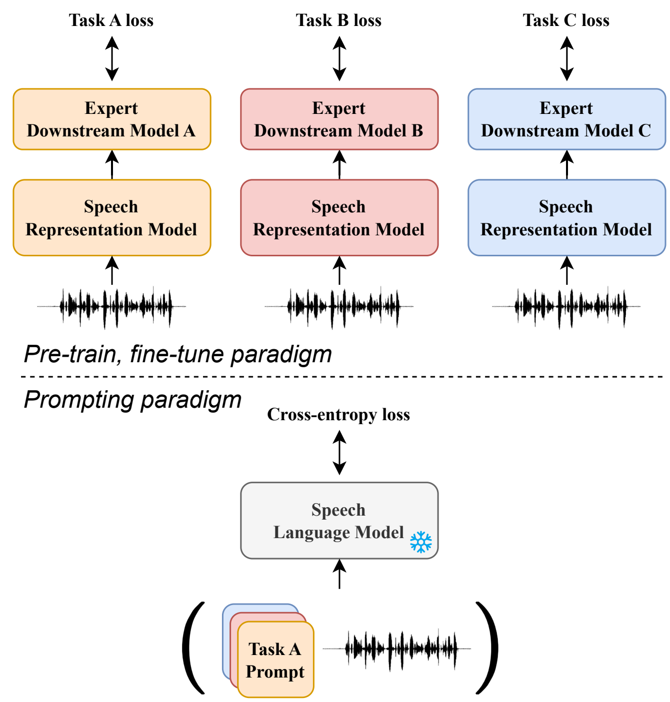


Fig. 1. Comparison of the pre-train, fine-tune paradigm with the prompting paradigm. The pre-train, fine-tune paradigm involves designing task-specific downstream models and loss functions by human experts, with distinct models trained for each task. In contrast, the prompting paradigm handles all downstream tasks in a unified manner, where only the prompt varies for each task, while the language model remains fixed.

I. Introduction

Recently, self-supervised representation learning has become an essential component in the speech processing field [1]. The speech *representation model* is trained on a large-scale unlabeled corpus in a self-supervised learning (SSL) manner. The learned representation has been demonstrated to be informative and can benefit a wide range of speech processing tasks [2]–[4].

When leveraging these speech representation models for a downstream task of interest, a typical approach is to follow the **"pre-train, fine-tune"** paradigm [1], [5]. Under this paradigm, the representation models serve as feature extractors. The models encode speech into informative representations, which are subsequently fed into a task-specific model. This model, referred to as the *expert downstream model*, specializes in solving a specific speech processing task. While fine-tuning often yields optimal performance, this paradigm, as depicted in Fig. 1, requires delicately designing a task-specific downstream model and loss function for each task. This complexity significantly causes an increasing burden of human labor. Furthermore, the requirement to train the expert downstream model alongside the optionally fine-tuned speech representation model leads to substantial computational and storage demands. This is especially challenging as the number of downstream tasks grows due to the necessity to store separate model parameters for each task.

On the other hand, researchers have explored the **"prompting paradigm"** [5] as an alternative method to leverage pre-trained language models (LMs) to solve downstream tasks in an efficient manner. Originating from the Natural Language Processing (NLP) field, prompting refers to the technique that finds a task-specific template or instruction, which is called **prompt**, to steer a pre-trained LM without modifying

TABLE I

COMPARATIVE ANALYSIS OF PROMPTING AND PRE-TRAIN, FINE-TUNE PARADIGMS ACROSS VARIOUS CRITERIA. SYMBOLS USED: ✓ INDICATES A RELATIVE ADVANTAGE. △ DENOTES COMPARABLE PERFORMANCE. × INDICATES NO SIGNIFICANT ADVANTAGE.

Criterion	Prompting	Pre-train Fine-tune
Objective Engineering	×	✓
Expert Model Engineering	×	✓
Task-Specific Performance	△	✓
Low-resource Performance	✓	△
Storage Efficiency	✓	×
Computation Efficiency	✓	×
Deployment Efficiency	✓	×

its architecture and parameters. For each specific task, these templates can be hand-crafted or identified through a search process and are composed of the model's vocabulary, known as *hard prompts* [6], [7]. For instance, in sentiment classification, an input sentence $\langle S \rangle$ can be fit into a template: "$\langle S \rangle$. It was __." and then fed into a pre-trained LM. The LM's output (e.g., "great", "terrible") is then transformed into sentiment classes (positive, negative) by a *verbalizer* [8], [9], which is often a hand-crafted or a searched mapping function [10], enabling us to determine the sentiment of $\langle S \rangle$. Alternatively, prompts are not necessarily to be human-readable. Researchers have proposed a prompting method known as **prompt tuning**, which involves learning continuous prompts [5], [11]–[14] within the model's embedding space. These prompt vectors, also called *soft prompts*, are trainable and have shown to be effective and efficient for leveraging pre-trained models, with applications extending beyond the NLP field. For example, prompt tuning has been applied to computer vision [15] and speech processing [16].

The prompting paradigm presents multiple advantages compared to the traditional pre-train, fine-tune paradigm:

(1) **Training Efficiency:** Only prompt vectors require updating, offering better computational efficiency than the full model and downstream head training in the typical fine-tuning paradigm. Moreover, reformulating downstream tasks into a unified sequence generation task eliminates the need for specialized downstream models and loss functions.

(2) **Inference Uniformity:** With prompting, the LM remains fixed, enabling a uniform forward process for diverse tasks. The task specificity is driven by the input prompts, facilitating *in-batch tasking* [11], [14], the concurrent handling of multiple tasks within a single batch.

(3) **Deployment Scalability:** Recently, language models are increasingly deployed as services. The low computational and storage demands of prompting offer significant advantages. This is because the LM does not require retraining when serving a user's own dataset and task; instead, only task-specific prompts containing a small set of parameters need to be identified. As the number of tasks or users grows, the scalability and efficiency of prompting become even more beneficial [17]. The advantages of both the prompting paradigm and the pre-train, fine-tune paradigm are illustrated in Table I.

This paper focuses on prompting the *textless speech language models* [18]–[20]. These models are a class of generative LMs that are trained on discrete speech units obtained by quantizing the SSL speech representations [21]. Discrete speech units have gained researchers' attention because they offer several advantages: (1) Discrete units require less storage space and transmission bitrate compared to raw waveforms [22]. (2) Discrete units contain essential acoustic and linguistic information while minimizing speaker-specific information [21], which is useful for scenarios where privacy is a major concern [22], [23]. Mirroring the text LMs in the NLP field, these textless speech LMs adopt discrete units as their vocabulary and undergo pre-training through tasks like next token prediction [24] and the denoising sequence-to-sequence [25] task. Thanks to these speech LMs, several works have demonstrated promising results in challenging speech processing tasks, including speech continuation [18] and speech-to-speech translation [20]— tasks that are hard to achieve with the traditional pre-train, fine-tune paradigm. The textless property is particularly compelling since many languages worldwide lack substantial text resources [26]. These languages may either have no written form or lack a standardized written format. By directly modeling the phonetic and acoustic patterns, we can not only bypass the constraints and potential biases of written languages but also reduce the need for paired speech-text data, which is often costly to get.

Furthermore, the ability of these discrete units to encapsulate both acoustic and linguistic [18], [21] information without text supervision has opened up new opportunities for prompting the speech LM for a variety of speech processing tasks. Leveraging the unique characteristic of discrete units, we reformulate (1) speech classification tasks (speech to class label), (2) sequence generation tasks (speech to label sequence), and (3) speech generation tasks (speech to speech) into a unified *speech-to-unit generation* tasks. In the meantime, we propose utilizing a learnable verbalizer specifically for addressing speech classification and sequence generation tasks. Despite its simplicity as a linear transformation, this verbalizer can effectively utilize the information encapsulated in the discrete units, bridging that rich information with the downstream labels. The experiment results show that with the proposed method, the speech LM can solve speech classification tasks and sequence generation tasks with competitive performance compared to the pre-train, fine-tune paradigm. Also, thanks to the generative capability of the speech LMs, the proposed method can also deliver promising results on speech generation tasks, which are challenging for the fine-tuning paradigm. All the tasks are solved in a unified pipeline and with promising trainable parameter efficiency.

The advantages of the proposed unified prompt framework are as follows: (1). We are pioneers in introducing prompt engineering to the speech domain. Our proposed method achieves results comparable to the fine-tuning approach based on self-supervised learning. (2). Compared with the "pre-train, fine-tune" paradigm, our unified framework is adaptable to a wide range of speech tasks and eliminates the need for designing task-specific downstream models and loss functions. This approach not only saves considerable effort but also paves the way for a universal speech model. (3). The learnable verbalizer boasts commendable explainability and adeptly utilizes the semantic information within the discrete units. This capacity allows for an effective linkage of that information with the labels associated with various downstream tasks. (4). The evolution from GSLM to Unit mBART has significantly enhanced the performance of our prompt framework. With more advanced speech language models (LMs) coming into the stage, we anticipate these developments will elevate our methods to unprecedented levels of success. (5). Imagine a near future where speech language models are offered in the cloud servers by major companies and widely adopted by numerous smaller businesses. In this scenario, our prompt framework utilizes discrete units, which saves storage space, speeds up data transmission, and importantly, enhances privacy. For example, discrete speech units are disentangled from the speaker's identity [21]. Therefore, compared to transmitting raw speech, the use of discrete speech units with speaker information removed mitigates privacy concerns.

II. Related Works

A. Self-supervised Speech Representation and Discretization

The exploration of speech representations through Self-Supervised Learning (SSL) objectives has evolved into a crucial research topic within the speech research area in recent years. By utilizing different SSL pre-training tasks, the representation models can mainly be grouped into three categories: predictive models [27], [28], contrastive models [29]–[31], and generative models [32]–[34]. To leverage SSL representations, a common way is to build specialized downstream models on top of SSL representations and fine-tune the entire model or only the downstream models for supervised downstream tasks. Based on this, SUPERB [2] benchmarks SSL speech models with a wide variety of downstream tasks.

Although using continuous SSL representations as features for downstream tasks can yield stronger performance [35], there's a growing trend of adopting discrete speech units derived by quantizing the SSL representations [22], [36]. A common approach involves applying the K-means algorithm to the SSL representations, quantizing them into clusters. Discrete units significantly reduce storage space and transmission bandwidth compared to raw waveforms and SSL features [22], [35]. For instance, as discussed in [22] and shown in Table II, a T-second 16kHz waveform in 16-bit format requires 16 $\times$ 16,000 $\times$ T bits for storage and transmission. In contrast, HuBERT representation with a dimension of 768 and a frame rate of 50 per second results in 6.4 times the data size using floating-point vectors (32-bit). Discrete units with 100 clusters (approximately 7 bits) and 1,000 clusters (approximately 10 bits) offer even more efficient speech data formats.

TABLE II
DATA SIZE FOR DIFFERENT FORMATS OF T-SECOND SPEECH.

Data format	Data size (bits)	Size ratio
Raw waveform	$16 \times 16000 \times T$	1
SSL representation	$32 \times 1024 \times 50 \times T$	6.4
HuBERT units (100 clusters)	$7 \times 50 \times T$	1×10^{-3}
HuBERT units (1,000 clusters)	$10 \times 50 \times T$	2×10^{-3}

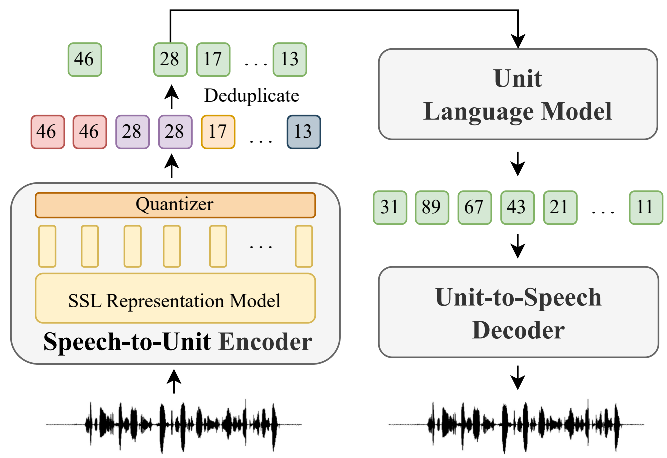


Fig. 2. The textless speech LM. It consists of three components, including (1) The speech-to-unit encoder, (2) the unit language model, and (3) the unit-to-speech decoder.

B. Textless Speech Language Models

Textless speech LMs regard discrete speech units as *pseudo-text* and adopt them as LM's vocabulary. Leveraging these discrete units, speech LMs are trained to perform language modeling tasks that mirror those in the NLP field.

As shown in Fig. 2, in the textless speech language model, there are three components: (1) speech-to-unit encoder, (2) unit language model, and (3) unit-to-speech decoder. **Speech-to-unit encoder** comprises an SSL representation model, such as HuBERT [27], paired with a quantizer, like K-means. The continuous representation extracted by the SSL model is clustered into discrete units. These discrete units have shown to encapsulate rich phonetic and linguistic information, thereby effectively representing speech [18], [21]. In conventional speech language models, these discrete units undergo a *deduplication* process, which removes consecutive repeated units to form a more compact sequence of tokens for language modeling. The **unit language model** is an LM that performs generative language modeling based on the discrete units. For instance, in GSLM [18], the unit language model conduct the next-token-prediction task akin to GPTs [24], [37]. Unit mBART performs the denoising sequence reconstruction task similar to the BART model [25]. The **unit-to-speech decoder** is responsible for transforming the generated discrete unit sequences back into continuous speech signals. The architecture is akin to the conventional speech synthesis models [38], [39] that train on the unit sequence and speech signal.

In addition to GSLM and Unit mBART, there are other notable speech language models such as AudioLM [40], TWIST [41], and SPECTRON [42]. These models bring additional complexity and advancements to the field; however, they are not currently fully open-sourced. As the development of speech LMs continues to evolve, there is significant potential for our framework to be expanded and utilized more extensively in future research and applications.

C. Prompting and Reprogramming in Speech Processing

This journal paper is an extension of our previous work [16], where we explored the concept of prompting on speech LM, particularly GSLM. Previous work [16] showed promising results in speech classification tasks such as spoken command recognition and intent classification and demonstrated better parameter efficiency compared to the pre-train, fine-tune paradigm. However, despite achieving notable results in sequence generation tasks like ASR and slot filling, its performance still lags behind the fine-tuning method. In this paper, we further explore an advanced encoder-decoder speech LM, Unit mBART, across a broader range of speech processing tasks. This includes a more diverse set of speech classification tasks, as well as speech generation tasks. The results are more promising: (1) Prompting Unit mBART achieves competitive performance in sequence generation tasks and (2) Prompting Unit mBART is well-suited for speech generation tasks, thereby establishing a unified prompting framework for various speech processing tasks. Additionally, compared to our previous work, we introduce a learnable verbalizer in this paper to bridge the gap between discrete units and downstream task labels, enhancing both explainability and performance.

WavPrompt [43] is also a pioneer in studying the prompting paradigm in speech processing. WavPrompt consists of a text LM, GPT-2 [37], and an audio encoder, wav2vec 2.0 [30]. The text LM is prompted with audio embeddings and text questions to perform few-shot speech understanding tasks. In contrast to SpeechPrompt, which uses textless speech LM for various speech processing tasks, WavPrompt employs a text LM and performs limited speech understanding tasks.

On the other hand, the work [44] studies hand-crafted prompts for a speech recognition model, Whisper [45], for various speech recognition tasks. The backbone model, Whisper, is trained using large-scale speech-text paired data. In contrast, our work prompts a textless speech LM, and we not only focus on speech recognition, a type of sequence generation task, but also explore speech generation tasks.

Another branch of utilizing a pre-trained model's capability for different tasks is *model reprogramming* [46], [47]. In [48], [49], the input data (target domain) are first transformed with a task-specific function to become the reprogrammed data. The pre-trained acoustic model is then capable of generating labels for this reprogrammed data. These labels (source domain) are then mapped to the classes of downstream tasks (target domain) by a mapping function. This mapping function serves the same role as the verbalizer in the prompting method and is usually a random mapping in the reprogramming literature. We also adopt the idea of reprogramming a foundation model for solving various tasks. For example, in speech classification tasks and sequence generation tasks, the speech LM is prompted/reprogrammed to adapt to the distribution of the target domain (the class label and the transcription).

III. Method

The overview of the proposed framework is depicted in Fig. 3. The input speech waveform is encoded into a sequence of discrete units using an SSL speech model and a quantizer. The unit LM (Section III-A) then takes this unit sequence and performs conditional generation based on the task-specific prompts. The design of task-specific prompts will be illustrated in Section III-B. The prompts steer the unit LM to solve the downstream speech processing task, which is reformulated into a speech-to-unit generation task as discussed in Section III-C. The resulting unit sequence is transformed into the downstream task's target through a verbalizer (for speech classification and sequence generation tasks) or through a pre-trained speech decoder (for speech generation tasks) as discussed in Section III-D. The notations used in the section are listed in Table III

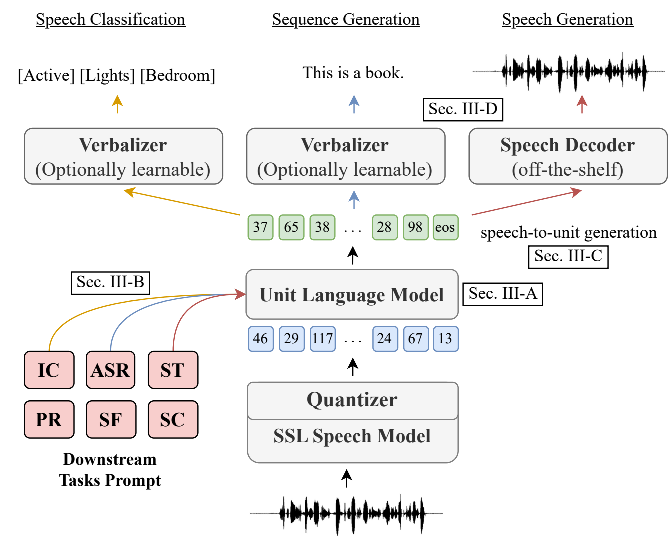


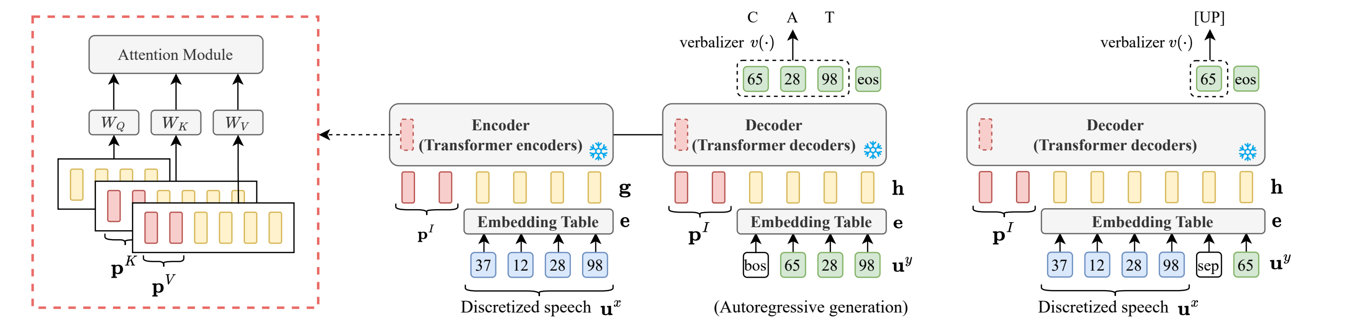
Fig. 3. An overview of the proposed framework, where all downstream tasks are treated as speech-to-unit generation processes. The generation of units is directed by the task-specific prompts that guide the unit language model. A verbalizer or speech decoder then bridges the gap between the generated units and the corresponding downstream labels.

TABLE III
NOTATION TABLE

Symbol	Description
u	Unit in the unit sequence
$\boldsymbol{u}^x$	Discretized speech, source unit sequence
$\boldsymbol{u}^y$	Generated target unit sequence
C	Context, including the input discretized speech, sequence of units before the current unit and the task prompts
$\boldsymbol{z}_{tj}$	Logit for j-th unit at timestep t
$P(u_j \mid C_t)$	Probability of unit u_j at timestep t given context C_t.
$\mathcal{E}$	Encoder in encoder-decoder unit LM
$\mathcal{D}$	Decoder in decoder-only or encoder-decoder unit LM
$\boldsymbol{e}(u)$	Unit LM's vocabulary embedding vector for a unit u
V	Vocabulary set $\{u_1, u_2, \ldots, u_{\lvert V\rvert}\}$
$\boldsymbol{g}^{(i)}$	Hidden representation input to the i-th layer of encoder
$\boldsymbol{h}^{(i)}$	Hidden representation input to the i-th layer of decoder
T	Sequence length of encoder's hidden representation
T'	Sequence length of decoder's hidden representation
$\boldsymbol{p}$	Trainable prompt sequence $[p_1, \ldots p_l]$ with prompt length l
$\boldsymbol{y}$	Downstream label sequence $[y_1, \ldots y_{T'}]$
$\lvert Y\rvert$	Number of classes in the downstream task

A. Unit Language Models

This subsection explains the backbone unit language model in our prompt framework. As shown in Fig. 4, these unit LMs



(a) Prompts at the input of the Transformer layer (b) Prompting Encoder-Decoder unit language model (c) Prompting Decoder-only unit language model

Fig. 4. An overview of the proposed framework, where all downstream tasks are treated as speech-to-unit generation processes. The generation of units is directed by the task-specific prompts that guide the unit language model. A verbalizer or speech decoder then bridges the gap between the generated units and the corresponding downstream labels.

receive discretized speech units sequence $\boldsymbol{u}^x$ and trainable prompts $\boldsymbol{p}$ as inputs, subsequently using them to generate target unit sequence $\boldsymbol{u}^y$ for downstream speech processing tasks.

Without loss of generality, in this paper, we investigate two variants of widely-adopted unit LMs based on Transformers [50]: (1) The decoder-only unit LM that mimics the GPT architecture [24], and (2) The encoder-decoder unit LM that mirrors the BART language model [25]. Both model types employ a causal decoder and are characterized as autoregressive LMs, enabling the capability to generate outputs of varying lengths. Specifically, the probability of each unit $u_t^y \in \boldsymbol{u}^y$ generated by the model at the timestep t is conditioned on the preceding context, denoted by C_t. The context C_t includes the input discretized source speech $\boldsymbol{u}^x$, the task prompts $\boldsymbol{p}$, and the units $\boldsymbol{u}^y_{<t}$ generated preceding the timestep t in the autoregressive process. Formally, the autoregressive model generates the probability of a unit u_j within a vocabulary $V = \{u_1, u_2, \dots, u_{|V|}\}$ at timestep t given the context C_t as:

$$P(u_j|C_t) = \frac{e^{\boldsymbol{z}_{tj}}}{\sum_{k=1}^{|V|} e^{\boldsymbol{z}_{tk}}}, \tag{1}$$

where $\boldsymbol{z}_{tj} \in \mathbb{R}^{|V|\times 1}$ is the logit for the j-th unit at timestep t, and the denominator is the sum of exponentiated logits for all units at that timestep.

1) Encoder-Decoder Unit LM: The encoder-decoder unit LM includes the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ based on Transformer. The discretized speech is first processed by the encoder $\mathcal{E}$ to form part of the enriched context that the decoder $\mathcal{D}$ performs cross-attention on to guide the generation of the discrete units. The encoder $\mathcal{E}$ is composed of multiple layers that process the input unit sequence:

$$\boldsymbol{g}^{(1)} = [\boldsymbol{e}(u_1^x), \boldsymbol{e}(u_2^x), \dots, \boldsymbol{e}(u_T^x)], \tag{2}$$

where T is the sequence length and $\boldsymbol{e}(\cdot) : \mathbb{Z} \mapsto \mathbb{R}^d$ denotes the vocabulary embedding table, which transforms a discrete unit $u \in \mathbb{Z}$ into its corresponding embedding vector $\boldsymbol{e}(u) \in \mathbb{R}^d$, and d is the embedding dimension. In the encoder, the i-th layer receives hidden representation $\boldsymbol{g}^{(i)} = [g_1^{(i)}, g_2^{(i)}, \dots, g_T^{(i)}]$ as input and outputs $\boldsymbol{g}^{(i+1)}$. The decoder layers operate similarly, with each taking input $\boldsymbol{h}^{(i)} = [h_1^{(i)}, h_2^{(i)}, \dots, h_{T'}^{(i)}]$, and outputs $\boldsymbol{h}^{(i+1)}$, where T' represents the decoder sequence length, which increases incrementally during the autoregressive process.

2) Decoder-only Unit LM: In the decoder-only LM, the model lacks the encoder and relies solely on the decoder $\mathcal{D}$, which functions in an analogous fashion to the encoder-decoder setup but without the encoder's guidance. Without the encoder, the discretized source speech $\boldsymbol{u}^x$ is integrated at the beginning of the sequence, serving as the initial context for the decoder to predict the subsequent units. A separation token $\langle sep \rangle$ is inserted in between the source unit sequence $\boldsymbol{u}^x$ and the generated units $\boldsymbol{u}^y$. Therefore, for each timestep t in the autoregressive process, the input to the decoder $\mathcal{D}$ is:

$$\boldsymbol{h}^{(1)} = [\boldsymbol{e}(\boldsymbol{u}^x), \boldsymbol{e}(\langle sep \rangle), \boldsymbol{e}(u_1^y), \dots, \boldsymbol{e}(u_{<t}^y)]. \tag{3}$$

B. Prompt Tuning

As depicted in Fig. 3, the speech LM is capable of performing predefined speech tasks when provided with various types of prompts. In this subsection, we will elaborate on the process of prompt design.

Prompting employs task-specific templates, known as prompts, to steer the generation process of the LM. This technique involves freezing the LM's parameters while integrating prompts as part of the input. Our method, inspired by the prompt tuning approaches [12], [14], is implemented in two positions: (1) at the input of the unit LM, termed *input prompt tuning*, and (2) at the input of each Transformer layer, termed *deep prompt tuning*.

1) Input Prompt Tuning: Inspired by the method in [14], input prompt tuning prepends continuous prompt vectors at the LM's input. Specifically, the prompts are prepended at the embedding sequence of the first layer's input $\boldsymbol{h}^{(1)}$ (and $\boldsymbol{g}^{(1)}$ for Encodoer-Decoder model):

$$\boldsymbol{h}^{(1)} \leftarrow Concat(\boldsymbol{p}^I, \boldsymbol{h}^{(1)}), \tag{4}$$

$$\boldsymbol{g}^{(1)} \leftarrow Concat(\boldsymbol{p}^I, \boldsymbol{g}^{(1)}), \tag{5}$$

where $\boldsymbol{p}^I = [p_1^I, p_2^I, \dots, p_l^I]$ represents a series of prompt vectors $p \in \mathbb{R}^d$ at the input of the unit LM, with l indicating the prompt length.

2) Deep Prompt Tuning: Inspired by prefix-tuning [12], deep prompt tuning involves concatenating prompt vectors at the input of the Transformer layer. Specifically, it modifies the input of the attention modules to guide the forward process of the LM. The self-attention module at the beginning of each transformer layer takes the Query (Q), Key (K), and Value (V) as input:

$$Attn(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{6}$$

where $\sqrt{d_k}$, the square root of the dimensionality of the key vectors, scales the dot product to ensure normalization of the attention weights by the softmax function. For self-attention, the matrices Q, K, and V are projections of the same input $\boldsymbol{g}$ or $\boldsymbol{h}$ transformed by the weight matrices W_Q, W_K, and W_V, respectively. Trainable prompt vectors are prepended to the input of each transformer layer, affecting both Key (K) and Value (V) matrices in the attention mechanism:

$$K \leftarrow Concat(\boldsymbol{p}^K, \boldsymbol{h})W_K, \tag{7}$$

$$V \leftarrow Concat(\boldsymbol{p}^V, \boldsymbol{h})W_V, \tag{8}$$

where $\boldsymbol{p}^K = [p_1^K, p_2^K, \ldots, p_l^K]$ and $\boldsymbol{p}^V = [p_1^V, p_2^V, \ldots, p_l^V]$ are series of trainable prompt vectors for key and value, respectively, and has the same prompt length l as $\boldsymbol{p}^I$.

Similar adjustments are applied to the encoder's representation $\boldsymbol{g}$ for encoder-decoder unit LM. It is crucial to note that throughout the prompt tuning process, only the prompt vectors are trainable. The embedding table and the unit LM remain fixed.

C. Speech-to-Unit Generation

In this paper, we focus on leveraging the generative capabilities of autoregressive speech LMs to handle various downstream tasks. Specifically, we recast speech processing tasks, including speech classification, sequence generation, and speech generation, into a unified *speech-to-unit generation task*. In this approach, speech LM takes discretized speech as input and generates a sequence of discrete units corresponding to the intended output for the task at hand.

In *sequence generation* tasks, like automatic speech recognition (ASR), the model generates a unit sequence $\boldsymbol{u}^y = [u_1^y, ..., u_{T'}^y, \langle eos \rangle]$. Each unit u_t^y represents a discrete token corresponding to the character y_t in the target character sequence $\boldsymbol{y} = [y_1, ..., y_{T'}]$. The mapping from units to characters is facilitated by the verbalizer, detailed in Section III-D. For *speech classification* tasks like spoken command recognition (SCR), which involve single-label classification, the model's goal is to classify an utterance into a predefined category. Instead of directly predicting a label y_1, it generates a unit sequence $\boldsymbol{u}^y = [u_1^y, \langle eos \rangle]$, where u_1^y will be transformed into the label y_1. In *speech generation* tasks, the generated unit sequence can be synthesized back into the target speech signal using an off-the-shelf unit-to-speech decoder. Notably, the autoregressive nature of speech LMs allows them to handle varying label lengths across different tasks, thus enabling a unified framework.

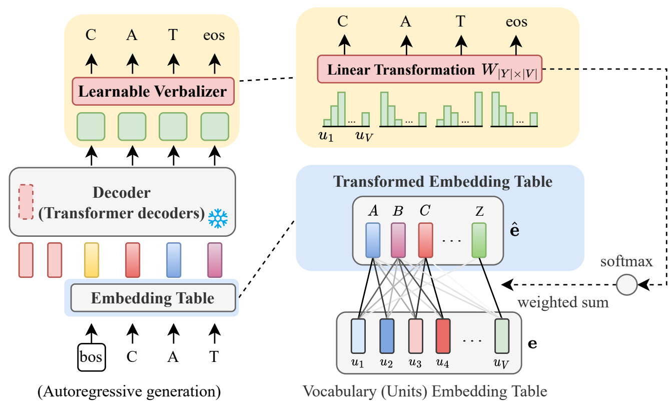


Fig. 5. Illustration of the learnable verbalizer. The logits are transformed into labels for the downstream task through a linear transformation. Furthermore, the original vocabulary embeddings are converted into class-specific embeddings using weighted transformations, aligning them more closely with the downstream task.

D. Verbalizer and Speech Decoder

Within the prompting paradigm, the *verbalizer* [8], [9] $v(\cdot)$ is a label-mapping module, which establishes the connection between the downstream task labels and the LM's vocabulary. For speech LM, the vocabulary is the discrete units. The verbalizer can adopt various forms, including random mapping [49], [51], [52] and heuristic methods [16], we refer to this as "fixed verbalizer" since the mapping is pre-defined and does not inlcude updates. On the other hand, to generate speech signal, a speech decoder is employed to synthesize waveform from discrete unit sequence.

1) Fixed Verbalizer: The fixed verbalizer establishes a static mapping between the downstream task label and a unique unit. For example, in the ASR task, it might map the character "a" to "unit 28" and "b" to "unit 72." In spoken command recognition (SCR), it could map the command "[UP]" to "unit 65," following either a random mapping or a frequency-based approach [16]. Once established, this mapping remains static without further learning or adaptation. In practice, with a fixed verbalizer, the most probable unit at each timestep t is selected and directly converted to the downstream task's label y_t.

2) Speech Decoder: For speech generation tasks, where the target output is a speech signal rather than a sequence of labels, the discrete units can be synthesized back into speech signals using a pre-trained, off-the-shelf unit-to-speech decoder. This speech decoder is self-supervised and trained with pairs of discrete units and their corresponding speech. In this work, we employ a speech decoder that corresponds to the given unit LM, as illustrated in Fig. 2.

E. Learnable Verbalizer

Fixed verbalizers can lead to subpar performance in speech processing tasks because, unlike the distinct semantic meaning present in NLP vocabulary, the vocabulary of discrete speech units lacks clear semantic meanings. To address this, we introduce a learnable verbalizer coupled with a novel input transformation (Fig. 5) that aligns the discrete units with the downstream task labels more meaningfully. In a learnable verbalizer, the mappings are determined by a learnable linear transformation matrix $W \in \mathbb{R}^{|Y| \times |V|}$, where $|V|$ is the size of the original LM's vocabulary, and $|Y|$ is the number of classes in the downstream task. This matrix is applied to the logits vector $\boldsymbol{z}_t \in \mathbb{R}^{|V| \times 1}$ to produce a transformed logits vector $\hat{\boldsymbol{z}}_t \in \mathbb{R}^{|Y| \times 1}$ over the downstream task labels:

$$\hat{\boldsymbol{z}}_t = W \cdot \boldsymbol{z}_t \tag{9}$$

Following this transformation, the label y_t is sampled from the transformed logits:

$$y_t = \underset{y}{\operatorname{argmax}}\ P(y|\hat{\boldsymbol{z}}_t), \tag{10}$$

where $P(y|\hat{\boldsymbol{z}}_t)$ is the probability of class y given the transformed logits $\hat{\boldsymbol{z}}_t$ at timestep t.

To facilitate autoregressive processing and incorporate the predicted downstream tasks' labels as input to the unit LM, we propose an input transformation mechanism, which is coupled with the learnable verbalizer matrix W. This mechanism transforms the original vocabulary embeddings into new embeddings suitable for the downstream task's labels.

Mathematically, the transformed embedding for a given class y, denoted as $\hat{\boldsymbol{e}}(y)$, is computed as a weighted sum of the original vocabulary embeddings $\boldsymbol{e}$. Let $W_{y:}$ denote the y-th row of the matrix W, where the i-th element in $W_{y:}$ represents the learned weight of the i-th unit contributing to the class y. $W_{y:}$ is then input into the softmax function with the temperature parameter τ, transforming the weights into probabilities. The formula is expressed as:

$$\hat{\boldsymbol{e}}(y) = \sum_{i=1}^{|V|} \left(softmax \left(\frac{W_{y:}}{\tau} \right) \right)_i \cdot \boldsymbol{e}(u_i) \tag{11}$$

In this formulation, $\hat{e}(y)$ signifies the newly generated embedding for class y, tailored to the demands of the downstream task. The process effectively creates class-specific embeddings by aggregating the original embeddings, each weighted according to the transformed softmax outputs. This method not only preserves the intrinsic properties of the original vocabulary embeddings but also aligns them more closely with the target classes of the downstream task, thereby enhancing the model's adaptability and effectiveness in handling varied speech processing applications.

The learnable verbalizer offers improved explainability, effectively utilizing the information in the discrete units, which will be discussed in Sec. V-C . In the meanwhile, it preserves parameter efficiency in the prompting paradigm. For instance, in the ASR task featuring 28 classes and a Unit mBART model with 1,000 units, the verbalizer necessitates fewer than 30,000 learnable parameters.

IV. Experimental Setup

In this work, we compare the pre-train, fine-tune paradigm with the prompting paradigm for speech processing across three types of tasks: (1) speech classification tasks, (2) sequence generation tasks, and (3) speech generation tasks. The used dataset and the basic statistics are presented in Table IV.

TABLE IV
THE DOWNSTREAM TASKS PERFORMED IN THIS PAPER, INCLUDING SPEECH CLASSIFICATION, SEQUENCE GENERATION, AND SPEECH GENERATION TASKS. LANGUAGE ABBREVIATIONS ARE IN ISO 639-1 FORMAT. $\boldsymbol{N}_{\text{CLASS}}$: NUMBER OF CLASSES FOR EACH DOWNSTREAM TASK. $\overline{|\boldsymbol{u}^x|}$: AVERAGE DISCRETE UNIT LENGTH OF THE UTTERANCE. $\overline{|\boldsymbol{u_{de}}|}$: AVERAGE DEDUPLICATED DISCRETE UNIT LENGTH OF THE UTTERANCE. $\overline{|\boldsymbol{y}|}$: AVERAGE LABEL LENGTH.

Task	Dataset	Language	$\boldsymbol{N}_{\text{class}}$	$\overline{\lvert\boldsymbol{u}^x\rvert}$	$\overline{\lvert\boldsymbol{u_{de}}\rvert}$	$\overline{\lvert\boldsymbol{y}\rvert}$
Speech Classification						
SCR	Google SC v1	en	12	48	25	1
	Arabic SC	ar	16	41	25	1
	Lithuanian SC	lt	15	51	28	1
	DM-SC	zh	19	169	68	1
	Grabo SC	nl	36	132	71	1
IC	Fluent SC	en	24	115	61	3
SD	Mustard++	en	2	229	128	1
AcC	AccentDB	en	9	205	91	1
LID	Voxforge	en, es, fr, de, ru, it	6	392	231	1
VAD	Google SC v2 / FreeSound	en / audio	2	31	16	1
Sequence Generation						
ASR	LibriSpeech	en	28	591	355	172
PR	LibriSpeech	en	71	591	355	116
SF	AudioSNIPS	en	107	142	96	53
Speech Generation						
ST	CoVoST2	en, es	$\lvert V\rvert$	305	167	120
SC	LJSpeech	en, es	$\lvert V\rvert$	328	199	199

A. Tasks and Datasets

1) Speech Classification Tasks:

Speech Command Recognition (SCR): The task is to recognize which keyword is presented in a given utterance. We adopted the Google Speech Commands dataset [53] and low-resource datasets in different languages. These include Grabo Speech Commands (Grabo-SC) [54], Lithuanian Speech Commands (LT-SC) [55], Dysarthric Mandarin Speech Commands (DM-SC) [56], and Arabic Speech Commands (AR-SC) [57].
Intent Classification (IC): This task classifies utterances into predefined classes to determine the intent of speakers. We used the Fluent Speech Commands dataset [58], where each utterance has three labels: action, object, and location.
Language Identification (LID): The objective of this task is to recognize the language present in a given utterance. We utilized the Voxforge Dataset [59], which comprises utterances in six different languages.
Sarcasm Detection (SD): This task aims to determine if an utterance is sarcastic. We employed the Mustard++ dataset [60].
Voice Activity Detection (VAD): This task is to determine whether a segment of an utterance contains human speech or is just background noise or silence. Following MarbleNet [61], we used Google Speech Commands v2 [62] as speech data and FreeSound dataset [63] as background noise data. We refer to this mixed dataset as GFSound.

2) Sequence Generation Tasks:

Automatic Speech Recognition (ASR): The task is to transcribe an utterance into text (character sequence). We utilized the LibriSpeech [64] train-clean-100 dataset for training and the test-clean dataset for testing. The evaluation metrics are word error rate (WER) and character error rate (CER).

Phoneme Recognition (PR): The task involves transcribing an utterance into a phoneme sequence. The evaluation metric is phoneme error rate (PER).

Slot Filling (SF): In the slot filling task, models are expected not only to recognize the spoken content but also to decode the associated slot type. Specifically, the slot type is decoded in conjunction with the transcription in a sequence generation approach. We adopted AudioSNIPS dataset [65] and the evaluation metrics are character error rate (CER) and F1 score.

3) Speech Generation Tasks:

Speech Translation (ST): ST is the process of converting speech signals from the source language into speech in the target language, enabling communication between individuals who speak different languages. We utilize the CoVoST2 [66] Es-En dataset. This dataset comprises parallel text data for Spanish (Es) and English (En) translations. Following [20], we utilize a single-speaker TTS system [1] to synthesize the speech of the target language. We utilize an off-the-shelf ASR system [2] to transcribe the generated speech and calculate BLEU scores with sacrebleu [3].

Speech Continuation (SC): SC aims to generate coherent continuation of a given speech input while preserving the semantic context. In the experiment, we adopt LJSpeech [67], which contains approximately 24 hours of English speech from a single speaker. We divided the LJSpeech dataset into training, validation, and testing subsets. Within these subsets, we designated each utterance's initial r fraction as the seed segment for the speech continuation tasks. We refer to the value of r as the *conditional ratio*. Given this seed segment, our model aims to generate a coherent continuation of the speech. Following ST, the generated speech is first transcribed into text, after which Perplexity (PPX) [4] and Auto-BLEU [18] are evaluated.

B. Model and Training Setup

We compare the pre-train, fine-tune paradigm with the prompting paradigm to assess whether prompting can achieve competitive performance while also providing parameter efficiency and other associated benefits as discussed in Table. I.

1) Prompting Paradigm: We explore two types of speech LMs within the prompting paradigm: the decoder-only *Generative Spoken Language Model (GSLM)* and the encoder-decoder model *Unit mBART*. GSLM is pre-trained using a next token prediction task on discrete units obtained by quantizing the 6-th layer of HuBERT representations into 100 clusters. The GSLM paper [18] considered different settings, including various SSL models and cluster numbers. This setting is selected for its superior performance. On the other hand, Unit mBART is pre-trained on a multilingual denoising task using discrete units derived from quantizing the 11-th layer of mHuBERT representations into 1,000 clusters.

In our experiments, we set the prompt length $l = 5$ for GSLM on speech classification tasks and $l = 3$ for Unit mBART. In sequence generation tasks, the prompt lengths are $l = 180$ for GSLM and $l = 50$ for Unit mBART. For speech generation tasks, we adopt a prompt length of $l = 200$ for Unit mBART and a prompt length of $l = 180$ for GSLM. The prompt length is a hyperparameter that can be adjusted for different numbers of trainable parameters. Since the architecture of both models is different, the positions in which the prompts can be inserted differ; notably, Unit mBART has an extra encoder for this purpose. Consequently, we have employed different prompt lengths for the two models with the aim of maintaining a comparable number of trainable parameters for a fair comparison with the pre-train, fine-tune paradigm. We used random mapping for the fixed verbalizer. We adopt random mapping instead of a heuristic frequency-based approach [16] for two reasons: (1) In the preliminary study, we do not observe significant performance improvement when utilizing the heuristic method. (2) In the few-shot learning scenario, the statistics of the discrete units are inadequate. For the learnable verbalizer, we set the softmax's temperature $\tau = 0.01$ in the input transformation.

2) Pre-train, Fine-tune Paradigm: In the pre-train, fine-tune paradigm, we train an expert downstream model for each task, utilizing the SSL speech representation as inputs. We adopt the same layer of the intermediate representation that derives the discrete units for the expert downstream model. [35] indicates that using the SSL speech representation as input is a strong baseline compared to using discrete units as input.

For the expert downstream model's design, we follow SUPERB [2] and adjust the hidden dimension of the downstream model to achieve a lightweight expert model to compare with the prompting paradigm. For speech classification tasks, we employ a linear model with a cross-entropy loss function. We utilize a 2-layer LSTM for sequence generation tasks [5], and a 2-layer Transformer for speech generation tasks.

V. Results

A. Main Results

1) Speech Classification Tasks: The comparison of the prompting paradigm (PT) and the pre-train, fine-tune paradigm (FT) for the speech classification tasks are shown in Table V. Our results indicate that the prompting method generally delivers competitive performance and often outperforms the fine-tuning approach. Specifically, for HuBERT and GSLM models, prompting outperforms fine-tuning in 6 out of 10 datasets (AR-SC, LT-SC, DM-SC, Grabo-SC, Fluent-SC, and Mustard++). For mHuBERT and mBART, prompting excels in

[1] https://huggingface.co/espnet/kan-bayashi_ljspeech_vits

[2] "Wav2vec2_large_lv60k" with CTC decoder available on PyTorch

[3] https://github.com/mjpost/sacrebleu

[4] Evaluated with pre-trained LM "transformer_lm.wmt19.en" available on Fairseq

[5] In the SUPERB setting, phoneme recognition utilizes CTC loss and a linear downstream model for frame-wise prediction. Following SUPERB, we also employ a linear model for the pre-train, fine-tune paradigm.

TABLE V

PERFORMANCE COMPARISON ON SEQUENCE GENERATION TASKS FOR THE "PRE-TRAIN, FINE-TUNE PARADIGM" (**FT**) AND THE "PROMPTING PARADIGM" (**PT**). **HuBERT + Expert** AND **mHuBERT + Expert**: BUILDING AN EXPERT DOWNSTREAM MODEL ON TOP OF THE SSL SPEECH MODEL AND FINE-TUNING THE EXPERT MODEL. **GSLM$_{\text{fixed}}$** AND **Unit mBART$_{\text{fixed}}$**: PROMPTING THE SPEECH LANGUAGE MODEL WITH A FIXED VERBALIZER. **GSLM$_{\text{learn}}$** AND **Unit mBART$_{\text{learn}}$**: PROMPTING THE SPEECH LM WITH A LEARNABLE VERBALIZER. IN THE PT SCENARIOS, AROUND 0.15M TRAINABLE PARAMETERS ARE INCLUDED; WHILE FT ADOPTS AROUND 0.2M PARAMETERS.

Speech Classification Tasks (full dataset setting)											
Paradigm	Scenario	SCR Google-SC	SCR AR-SC	SCR LT-SC	SCR DM-SC	SCR Grabo-SC	IC Fluent SC	SD Mustard++	AcC AccentDB	LID Voxforge	VAD GFSound
FT	HuBERT + Expert	**94.88**	98.38	92.86	52.14	91.44	93.18	61.72	**99.53**	**97.73**	**98.55**
PT	GSLM$_{\text{fixed}}$	94.5	**99.7**	**93.2**	74.30	92.40	**98.76**	63.33	78.9	90.9	96.6
	GSLM$_{\text{learn}}$	94.71	99.19	92.86	**74.36**	**95.76**	98.58	**65.83**	80.02	87.69	97.01
FT	mHuBERT + Expert	93.59	99.46	**91.84**	64.96	89.92	93.57	60.42	**93.78**	98.23	**98.40**
PT	Unit mBART$_{\text{fixed}}$	93.99	96.22	**91.84**	64.96	**97.11**	**97.81**	**63.33**	88.71	**98.81**	97.26
	Unit mBART$_{\text{learn}}$	**94.45**	**99.73**	**91.84**	**77.78**	95.07	**97.81**	**63.33**	87.38	98.13	97.43

TABLE VI

PERFORMANCE COMPARISON ON SEQUENCE GENERATION TASKS FOR THE "PRE-TRAIN, FINE-TUNE PARADIGM" (**FT**) AND THE "PROMPTING PARADIGM" (**PT**).

Sequence Generation Tasks										
Paradigm	Scenario	ASR-LibriSpeech # Params.	ASR-LibriSpeech WER ↓	ASR-LibriSpeech CER ↓	PR-LibriSpeech # Params.	PR-LibriSpeech PER ↓	SF-AudioSnips # Params.	SF-AudioSnips CER ↓	SF-AudioSnips F1 ↑
FT	HuBERT + Expert	2.89M	**15.67**	**4.55**	2.64M	**5.34**	2.89M	**40.08**	**78.53**
PT	GSLM$_{\text{fixed}}$	4.5M	34.17	26.14	4.5M	21.10	4.5M	66.90	59.47
FT	mHuBERT + Expert	2.89M	14.44	**4.43**	2.64M	12.42	2.89M	32.24	85.26
PT	Unit mBART$_{\text{fixed}}$	2.6M	13.85	5.91	2.6M	5.16	2.6M	33.09	**87.20**
	Unit mBART$_{\text{learn}}$	2.6M*	**11.56**	5.13	2.6M	**4.95**	2.6M	**30.69**	87.08

8 out of 10 datasets, including all datasets under SCR (with LT-SC achieving identical performance), IC, SD, and LID.

For the few tasks where prompting is slightly outperformed by fine-tuning in HuBERT and GSLM settings (Google-SC, and GFSound), the performance gap is minimal, within a 2% difference. However, fine-tuning demonstrates a noticeable advantage in tasks like AcC and LID. This could be attributed to the loss of certain information, such as prosody, in the quantized HuBERT discrete units, leading to inferior GSLM performance compared to HuBERT with the downstream expert model. In the mHuBERT and Unit mBART settings, while fine-tuning outperformed in 2 tasks, the performance difference in VAD is marginal (about 1.2%).

When assessing the effectiveness of utilizing a learnable verbalizer compared to a fixed one for prompting, it's observed that for GSLM, performance is enhanced in 6 out of 10 datasets (Google-SC, DM-SC, Grabo-SC, Mustard++, AccentDB, and GFSound). For unit mBART, the performances have improved or are on par in 7 datasets (Google-SC, AR-SC, LT-SC, DM-SC, Fluent-SC, Mustard++, and GFSound)

In summary, the prompting methods greatly match or exceed the performance of the fine-tuning approach across most speech classification tasks (6 outperform and 3 comparable for HuBERT and GSLM; 8 outperform and 1 comparable for mHuBERT and unit BART), except in accent classification.

2) Sequence Generation Tasks: The experiment results of sequence generation tasks are shown in Table VI. In sequence generation tasks, we observe that although prompting the decoder-only model GSLM can yield non-trivial results, it still underperforms compared to the fine-tuning paradigm by a substantial margin. The reasons are discussed in previous work [16], including that quantizing speech into discrete units results in longer sequences, which might be difficult for a decoder model to handle. In our preliminary study, even utilizing a learnable verbalizer for prompting GSLM does not show any performance improvement.

On the other hand, surprisingly, prompting an encoder-decoder model like Unit mBART can achieve competitive performance, outperforming the fine-tuning paradigm in most scenarios, except for the metric CER in ASR, which falls behind by 0.7. Furthermore, we can observe the effectiveness of introducing a learnable verbalizer in Unit mBART. For every metric other than the F1 score in Slot Filling (SF), there is a substantial improvement when comparing Unit mBART$_{\text{learn}}$ with Unit mBART$_{\text{fixed}}$. The analysis of the learnable verbalizer will be discussed in Section V-C. From GSLM to Unit mBART, the speech LM becomes better, and tasks that previously yielded poor results with GSLM can now yield favorable outcomes with Unit mBART. We anticipate that in the future, with more advanced speech LMs emerging, further performance improvement can be seen with the proposed prompting framework.

3) Speech Generation Tasks: In speech generation tasks, we focus on two tasks: Speech Translation (ST) and Speech Continuation (SC). Our experiments show the effectiveness of prompting Unit mBART for speech translation, as detailed in Table VIII. Speech-to-speech translation poses significant challenges, often requiring incorporating auxiliary tasks [68],

TABLE VII
EVALUATION FOR THE SPEECH CONTINUATION TASK. BOTH GSLM AND Unit mBART ARE IN THE PROMPTING PARADIGM. ORIGINAL: THE GROUND TRUTH CORPUS IN THE DATASET. r: CONDITIONAL RATE.

Speech Generation Task (Speech Continuation)									
Senario	$r = 0.25$			$r = 0.5$			$r = 0.75$		
	PPX (↓)	Auto-BLEU-1	Auto-BLEU-2	PPX (↓)	Auto-BLEU-1	Auto-BLEU-2	PPX (↓)	Auto-BLEU-1	Auto-BLEU-2
GSLM	**422.62**	21.58	8.80	**341.30**	20.43	7.76	**282.26**	18.57	6.98
Unit mBART	543.80	14.20	1.51	420.49	13.84	1.60	283.03	14.09	2.41
Original	202.92	13.9	2.42	202.92	13.9	2.42	202.92	13.9	2.42

TABLE VIII
EVALUATION ON SPANISH TO ENGLISH SPEECH-TO-SPEECH TRANSLATION.

Speech Generation Task (Speech Translation)				
Scenario	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
mHuBERT + Expert	×			
GSLM	×			
Unit mBART	43.8	30.4	21.8	15.9

[69] or adopting an advanced speech LM [20]. Our results also support this observation: neither the prompting GSLM baseline nor the fine-tuning baseline with an expertly built mHuBERT model yielded reasonable results. We experimented with various learning rates; however, the fine-tuning baseline still yielded unsatisfactory outcomes. However, Unit mBART demonstrates proficiency in producing reasonable translations, as evidenced by its BLEU-1 through BLEU-4 scores, which indicate competitive translation quality. Examples of the speech generation tasks can be found on the demo page [6]

Similarly, in speech continuation tasks, the SSL speech models (HuBERT and mHuBERT) paired with expert models did not produce reasonable results. As shown in Table VII, for various conditional ratios r, we observed that prompting GSLM outperformed Unit mBART in terms of Perplexity (PPX), aligning with GSLM's pre-training for such tasks. Regarding the Auto-BLEU metric [18], Unit mBART achieved scores comparable to the original utterances in the LJ Speech dataset. This suggests that the utterances generated by Unit mBART are as diverse as the oracle utterances, a challenge where GSLM falls behind. Future research will explore varying the sampling temperature to enhance utterance generation quality, as discussed in [18].

B. Few-shot Learning

The prompting method has demonstrated its few-shot learning capabilities in the NLP field [5], [9] because of the inherent rich prior knowledge within language models. Similarly, speech LMs have already learned to comprehend discretized speech, that is, the discrete speech units. This study extends the investigation into the few-shot learning abilities of the prompting method for speech LMs. Table IX illustrates the performance of the prompting method in comparison to the pre-train, fine-tune paradigm in a 10-shot learning scenario.

The experiment shows that the prompting paradigm (PT) possesses robust few-shot learning capabilities and generally outperforms the fine-tuning paradigm (FT) in most speech classification tasks. For HuBERT and GSLM, the FT method can only outperform the PT in 3 out of 10 tasks (LT-SC, AccentDB, GFSound). Meanwhile, for mHuBERT and Unit mBART, the FT method can only outperform the PT in 1 out of 10 tasks (GFSound).

Generally, in speech classification tasks under few-shot scenarios, prompting with Unit mBART achieves the best overall performance, showing top or near-top results in most tasks. Interestingly, we do not observe consistent performance improvement when utilizing a learnable verbalizer in these few-shot scenarios. We hypothesize that this might be due to the limited data, which causes challenges for the learnable verbalizer to extract the hidden information encapsulated in the discrete units effectively. The investigation of the underlying reason remains a future work.

C. Verbalizer Analysis

In this study, we introduce an optional learnable verbalizer that bridges the gap between discrete units and the downstream tasks' labels. Prior research has shown that discrete units encapsulate acoustic and phonetic information [18], [70]. Thus, rather than employing a random mapping of the heuristic method in ASR and PR, it is more reasonable to employ a learnable verbalizer that discerns which discrete units correlate with specific labels, such as characters or phonemes. The efficacy of the learnable verbalizer is presented in Fig. 6. This figure demonstrates the capability of the learnable verbalizer in linking discrete units with characters for the ASR task, as displayed in the figure's first row, and with phonemes for the PR task, as illustrated in the second row. The heatmaps display the weights W from the learnable verbalizer in Euqation 9, with each map's right side indicating the connected discrete unit. Besides the units, the top three phonemes with the highest correlation to the discrete units are listed, as determined by forced alignment. We observe that for a particular character, such as "B," the verbalizer prefers discrete units with a strong association with the phoneme "B." This pattern is consistent in the second row, which pertains to phoneme recognition tasks. Here, labels are connected to units with a high correspondence to the relevant phonemes.

VI. DISCUSSION

We list observations, limitations, and future directions:

Decoder-only versus Encoder-Decoder Speech Language Models: In the field of NLP, there is a growing trend towards employing decoder-only language models, particularly GPT variants, for a broad range of text generation tasks. However,

[6]Demo page: https://ga642381.github.io/SpeechPrompt/speechgen

TABLE IX
PERFORMANCE COMPARISON ON SPEECH CLASSIFICATION TASKS WITH LOW RESOURCE DATASET SETTING (10-SHOT) FOR THE "PRE-TRAIN, FINE-TUNE PARADIGM" (**FT**) AND THE "PROMPTING PARADIGM" (**PT**).

Speech Classification Tasks (10-shot setting)											
Paradigm	Scenario	Google-SC	AR-SC	SCR LT-SC	DM-SC	Grabo-SC	IC Fluent-SC	SD Mustard++	AcC AccentDB	LID Voxforge	VAD GFSound
FT	HuBERT + Expert	75.33	68.65	**80.61**	50.42	43.55	47.04	54.17	**78.71**	56.69	**92.85**
PT	$\text{GSLM}_{\text{fixed}}$	**79.55**	42.16	79.59	**62.39**	49.23	**73.5**	55.83	22.35	32.16	87.1
	$\text{GSLM}_{\text{learn}}$	77.15	**90.00**	68.36	61.50	**86.50**	72.87	**65.83**	26.71	**85.41**	89.30
FT	mHuBERT + Expert	76.88	94.59	72.45	47.01	76.24	47.01	56.77	50.96	90.65	**95.97**
PT	Unit $\text{mBART}_{\text{fixed}}$	**81.47**	**95.14**	78.57	**70.85**	**87.10**	55.81	**63.33**	49.23	**94.46**	95.21
	Unit $\text{mBART}_{\text{learn}}$	80.46	93.51	**86.74**	64.96	85.85	**64.90**	53.33	**57.32**	93.83	92.84

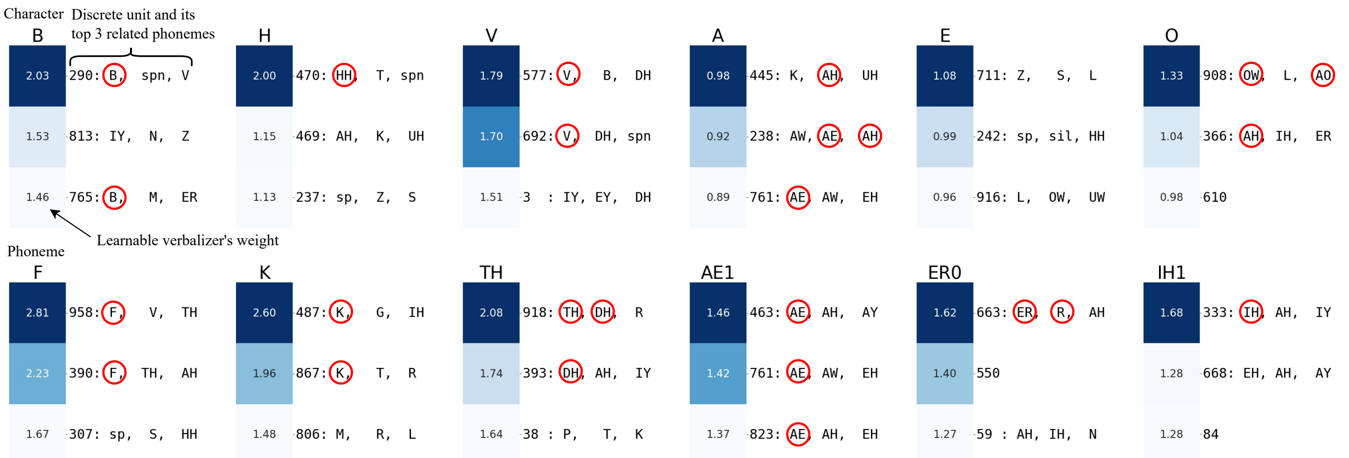


Fig. 6. Analysis of the Learnable Verbalizer. The top row presents heatmaps for the ASR task, with each subplot dedicated to the analysis of an individual character. The bottom row relates to the Phoneme Recognition (PR) task, with each subplot focused on a particular phoneme. The heatmaps show the weights that the learnable verbalizer assigns to the discrete units; for example, the phoneme "AE1" is most strongly linked to "Unit 463". Besides the units, the top three phonemes with the highest correlation to the discrete units are listed, which is determined by forced alignment. This visualization illustrates the learnable verbalizer's ability to effectively utilize the information encoded in the discrete units to map to suitable labels. Related phonemes to the downstream tasks labels are circled.

based on the experimental results, we suggest that encoder-decoder models may offer distinct advantages for speech processing. This is primarily because many speech processing tasks require handling different modalities, especially the speech signal and text. The unique continuous characteristics of speech signals may be more effectively processed by an encoder. Therefore, encoder-decoder models are likely better suited for the first encoding the speech signal into a compact representation, after which the decoder generates the desired output, be it a class label, text, or another speech signal. This observation aligns with recent work [71] comparing GSLM and Wav2Seq [72] models of similar sizes and datasets. The encoder-decoder model Wav2Seq demonstrates an advantage.

Performance of Prompting Speech Language Models: We have observed the competitive performance of the prompting Unit mBART model in both speech classification and generation tasks. Notably, in speech generation tasks, relying solely on an SSL speech model does not yield satisfactory performance. However, a discernible performance gap still exists between the prompting and fine-tuning paradigms, especially the sequence generation task. Taking SUPERB as an example, the setting involves performing a weighted sum over the representations of each layer of SSL speech models and building an expert model on top of this, along with adopting a customized loss for the downstream task. Although such a setting requires considerable human labor and computational resources, its performance is competitive. In Table. X, we list the ranking of the prompted Unit mBART for the ASR task and compare it with the SSL speech models on SUPERB.

TABLE X
SEQUENCE GENERATION TASK PERFORMANCE. THE MODELS ARE ORDERED BASED ON THE ASR PERFORMANCE.

Model	**ASR (WER ↓)**	**SF (CER ↓)**	**SF (F1 ↑)**	# params
FBANK	23.18	52.94	69.64	43M
modified CPC	20.18	49.91	71.19	43M
TERA	18.17	54.17	67.5	43M
vq-wav2vec	17.71	41.54	77.68	43M
wav2vec	15.86	43.71	76.37	43M
DeCoAR 2.0	13.02	34.73	83.28	43M
Unit $\text{mBART}_{\text{learn}}$	11.56	30.69	87.08	**2.89M**
wav2vec 2.0 Base	6.43	24.77	88.3	43M
HuBERT Base	6.42	25.2	88.53	43M
WavLM Base	6.21	22.86	89.38	43M
data2vec Large	**3.36**	**22.16**	**90.98**	43M

Develop Advanced Speech Language models: Speech language models are currently in their nascent stage of development compared to text-based language models. The proposed prompt framework, although effective in motivating speech LMs, may not achieve exceptional performance. However, with advancements in speech LMs, such as the transition from GSLM to Unit mBART, there has been a significant improvement in prompt performance. Particularly, tasks that were previously challenging for GSLM now exhibit improved performance with Unit mBART. We anticipate the emergence of even more promising speech LMs in the future.

Beyond Content Information: Current speech LMs do not fully capture speaker and emotion information, posing a challenge for tasks beyond content-related aspects. In scenarios where preserving speaker and emotion information is possible, we plan to explore the integration of plug-and-play modules specifically designed to incorporate speaker and emotion details into the framework. Looking ahead, we anticipate that future speech LMs will incorporate and leverage these additional factors and better handle speaker and emotion-related aspects in speech generation tasks. Google's latest speech LM [42] tries to include such information.

VII. Conclusion

In this paper, we investigate how prompting can leverage the generative capabilities of speech language models (speech LMs) for solving a wide range of speech processing tasks. Our approach includes minimal trainable parameters to guide the speech LMs within a unified framework, achieving competitive performance compared to the fine-tuning paradigm while keeping the benefits of the prompting paradigm. The proposed framework exhibits several desirable characteristics, including its textless nature, versatility, efficiency, transferability, and affordability. To demonstrate our framework's capabilities, we study the decoder-only GSLM and encoder-decoder Unit mBART as case studies. We conduct experiments on three distinct types of speech processing tasks: speech classification, sequence generation, and speech generation. Also, the proposed framework shows promising results in the few-shot scenario. We observe a trend that as more advanced speech LMs are developed, the performance of prompting will significantly improve. We also discuss the limitations and future directions of prompting speech LMs. With the imminent arrival of advanced speech LMs, our unified framework holds immense potential in terms of efficiency and effectiveness, standing on the shoulders of giants.

References

[1] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe *et al.*, "Self-supervised speech representation learning: A review," *arXiv preprint arXiv:2205.10643*, 2022.

[2] S. Yang, P. Chi, Y. Chuang, C. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G. Lin, T. Huang, W. Tseng, K. Lee, D. Liu, Z. Huang, S. Dong, S. Li, S. Watanabe, A. Mohamed, and H. Lee, "SUPERB: speech processing universal performance benchmark," in *Interspeech*, 2021, pp. 1194–1198.

[3] S. Evain, H. Nguyen, H. Le, M. Z. Boito, S. Mdhaffar, S. Alisamir, Z. Tong, N. A. Tomashenko, M. Dinarelli, T. Parcollet, A. Allauzen, Y. Estève, B. Lecouteux, F. Portet, S. Rossato, F. Ringeval, D. Schwab, and L. Besacier, "*LeBenchmark*: A reproducible framework for assessing self-supervised representation learning from speech," in *Interspeech*, 2021, pp. 1439–1443.

[4] H. Tsai, H. Chang, W. Huang, Z. Huang, K. Lakhotia, S. Yang, S. Dong, A. T. Liu, C. Lai, J. Shi, X. Chang, P. Hall, H. Chen, S. Li, S. Watanabe, A. Mohamed, and H. Lee, "SUPERB-SG: enhanced speech processing universal performance benchmark for semantic and generative capabilities," in *ACL (1)*. Association for Computational Linguistics, 2022, pp. 8479–8492.

[5] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[6] T. Shin, Y. Razeghi, R. L. L. IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," in *EMNLP (1)*. Association for Computational Linguistics, 2020, pp. 4222–4235.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020.

[8] T. Schick and H. Schütze, "It's not just size that matters: Small language models are also few-shot learners," in *NAACL-HLT*. Association for Computational Linguistics, 2021, pp. 2339–2352.

[9] ——, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *EACL*. Association for Computational Linguistics, 2021, pp. 255–269.

[10] S. Hu, N. Ding, H. Wang, Z. Liu, J. Wang, J. Li, W. Wu, and M. Sun, "Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 2225–2240.

[11] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models," *arXiv preprint arXiv:2203.06904*, 2022.

[12] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4582–4597.

[13] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, "Gpt understands, too," *AI Open*, 2023.

[14] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 3045–3059.

[15] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision*. Springer, 2022, pp. 709–727.

[16] K.-W. Chang, W.-C. Tseng, S.-W. Li, and H. yi Lee, "An Exploration of Prompt Tuning on Generative Spoken Language Model for Speech Processing Tasks," in *Proc. Interspeech 2022*, 2022, pp. 5005–5009.

[17] T. Sun, Y. Shao, H. Qian, X. Huang, and X. Qiu, "Black-box tuning for language-model-as-a-service," in *International Conference on Machine Learning*. PMLR, 2022, pp. 20 841–20 855.

[18] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed *et al.*, "On generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021.

[19] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T.-A. Nguyen, M. Rivière, A. Mohamed, E. Dupoux *et al.*, "Text-free prosody-aware generative spoken language modeling," *arXiv preprint arXiv:2109.03264*, 2021.

[20] S. Popuri, P.-J. Chen, C. Wang, J. Pino, Y. Adi, J. Gu, W.-N. Hsu, and A. Lee, "Enhanced Direct Speech-to-Speech Translation Using Self-supervised Pre-training and Data Augmentation," in *Proc. Interspeech 2022*, 2022, pp. 5195–5199.

[21] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W. Hsu, A. Mohamed, and E. Dupoux, "Speech resynthesis from discrete disentangled self-supervised representations," in *Interspeech*, 2021, pp. 3615–3619.

[22] 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," in *Proc. INTERSPEECH 2023*, 2023.

[23] A. Nautsch, A. Jiménez, A. Treiber, J. Kolberg, C. Jasserand, E. Kindt, H. Delgado, M. Todisco, M. A. Hmani, A. Mtibaa *et al.*, "Preserving privacy in speaker and speech characterisation," *Computer Speech & Language*, vol. 58, pp. 441–480, 2019.

[24] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[25] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *ACL*. Association for Computational Linguistics, 2020, pp. 7871–7880.

[26] E. Dunbar, N. Hamilakis, and E. Dupoux, "Self-supervised language learning from raw audio: Lessons from the zero resource speech challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1211–1226, 2022.

[27] 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.

[28] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3915–3924.

[29] M. Riviere, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7414–7418.

[30] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[31] D. Jiang, W. Li, M. Cao, W. Zou, and X. Li, "Speech simclr: Combining contrastive and reconstruction objective for self-supervised speech representation learning," *arXiv preprint arXiv:2010.13991*, 2020.

[32] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3497–3501.

[33] S. Ling and Y. Liu, "Decoar 2.0: Deep contextualized acoustic representations with vector quantization," *arXiv preprint arXiv:2012.06659*, 2020.

[34] A. T. Liu, S.-W. Li, and H.-y. Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.

[35] X. Chang, B. Yan, K. Choi, J. Jung, Y. Lu, S. Maiti, R. Sharma, J. Shi, J. Tian, S. Watanabe *et al.*, "Exploring speech recognition, translation, and understanding with discrete speech units: A comparative study," *arXiv preprint arXiv:2309.15800*, 2023.

[36] 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," *arXiv preprint arXiv:2309.07377*, 2023.

[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[38] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[39] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[40] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "Audiolm: a language modeling approach to audio generation," *CoRR*, vol. abs/2209.03143, 2022.

[41] M. Hassid, T. Remez, T. A. Nguyen, I. Gat, A. Conneau, F. Kreuk, J. Copet, A. Defossez, G. Synnaeve, E. Dupoux *et al.*, "Textually pretrained speech language models," *arXiv preprint arXiv:2305.13009*, 2023.

[42] E. Nachmani, A. Levkovitch, J. Salazar, C. Asawaroengchai, S. Mariooryad, R. Skerry-Ryan, and M. T. Ramanovich, "Lms with a voice: Spoken language modeling beyond speech tokens," *arXiv preprint arXiv:2305.15255*, 2023.

[43] H. Gao, J. Ni, K. Qian, Y. Zhang, S. Chang, and M. Hasegawa-Johnson, "WavPrompt: Towards Few-Shot Spoken Language Understanding with Frozen Language Models," in *Proc. Interspeech 2022*, 2022, pp. 2738–2742.

[44] P. Peng, B. Yan, S. Watanabe, and D. Harwath, "Prompting the Hidden Talent of Web-Scale Speech Models for Zero-Shot Task Generalization," in *Proc. INTERSPEECH 2023*, 2023, pp. 396–400.

[45] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[46] G. F. Elsayed, I. J. Goodfellow, and J. Sohl-Dickstein, "Adversarial reprogramming of neural networks," in *ICLR (Poster)*, 2019.

[47] P.-Y. Chen, "Model reprogramming: Resource-efficient cross-domain machine learning," *arXiv preprint arXiv:2202.10629*, 2022.

[48] C.-H. H. Yang, Y.-Y. Tsai, and P.-Y. Chen, "Voice2series: Reprogramming acoustic models for time series classification," in *International conference on machine learning*. PMLR, 2021, pp. 11 808–11 819.

[49] H. Yen, P. Ku, C. H. Yang, H. Hu, S. M. Siniscalchi, P. Chen, and Y. Tsao, "A study of low-resource speech commands recognition based on adversarial reprogramming," *CoRR*, vol. abs/2110.03894, 2021.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[51] T. Schick, H. Schmid, and H. Schütze, "Automatically identifying words that can serve as labels for few-shot text classification," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 5569–5578.

[52] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?" in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 11 048–11 064.

[53] P. Warden, "Speech commands: A public dataset for single-word speech recognition." *Dataset available online*, 2017.

[54] Y. Tian and P. J. Gorinski, "Improving end-to-end speech-to-intent classification with reptile," in *INTERSPEECH*, 2020, pp. 891–895.

[55] A. Kolesau and D. Šešok, "Unsupervised pre-training for voice activation," *Applied Sciences*, vol. 10, no. 23, p. 8643, 2020.

[56] Y.-Y. Lin, W.-Z. Zheng, W. C. Chu, J.-Y. Han, Y.-H. Hung, G.-M. Ho, C.-Y. Chang, and Y.-H. Lai, "A speech command control-based recognition system for dysarthric patients based on deep learning technology," *Applied Sciences*, vol. 11, no. 6, p. 2477, 2021.

[57] L. T. Benamer and O. A. Alkishriwo, "Database for arabic speech commands recognition," in *CEST*, 2020.

[58] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," in *Proc. of Interspeech*, G. Kubin and Z. Kacic, Eds., 2019.

[59] K. MacLean, "Voxforge," 2018, *Ken MacLean.[Online]. Available: http://www.voxforge.org/home.[Acedido em 2012]*.

[60] A. Ray, S. Mishra, A. Nunna, and P. Bhattacharyya, "A multimodal corpus for emotion recognition in sarcasm," in *Proceedings of the Thirteenth LREC*, 2022, pp. 6992–7003.

[61] F. Jia, S. Majumdar, and B. Ginsburg, "Marblenet: Deep 1d time-channel separable convolutional neural network for voice activity detection," in *ICASSP*, 2021, pp. 6818–6822.

[62] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[63] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: A platform for the creation of open audio datasets," in *ISMIR*, 2017, pp. 486–493.

[64] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[65] C. Lai, Y. Chuang, H. Lee, S. Li, and J. R. Glass, "Semi-supervised spoken language understanding via self-supervised speech and language model pretraining," in *ICASSP*. IEEE, 2021, pp. 7468–7472.

[66] C. Wang, A. Wu, and J. Pino, "Covost 2 and massively multilingual speech-to-text translation," *arXiv preprint arXiv:2007.10310*, 2020.

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

[68] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, "Direct speech-to-speech translation with a sequence-to-sequence model," *Interspeech 2019*, 2019.

[69] A. Lee, P.-J. Chen, C. Wang, J. Gu, S. Popuri, X. Ma, A. Polyak, Y. Adi, Q. He, Y. Tang *et al.*, "Direct speech-to-speech translation with discrete units," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 3327–3339.

[70] D. Wells, H. Tang, and K. Richmond, "Phonetic Analysis of Self-supervised Representations of English Speech," in *Proc. Interspeech 2022*, 2022, pp. 3583–3587.

[71] K.-W. Chang, M.-H. Chen, Y.-P. Lin, J. N. Hsu, P. K.-M. Huang, C.-y. Huang, S.-W. Li, and H.-y. Lee, "Prompting and adapter tuning for self-supervised encoder-decoder speech model," *arXiv preprint arXiv:2310.02971*, 2023.

[72] F. Wu, K. Kim, S. Watanabe, K. J. Han, R. McDonald, K. Q. Weinberger, and Y. Artzi, "Wav2seq: Pre-training speech-to-text encoder-decoder models using pseudo languages," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.