Recent Advances in Speech Language Models: A Survey

Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Yiwen Guo,
and Irwin King, *Fellow, IEEE*

Abstract—**Large Language Models (LLMs) have recently garnered significant attention, primarily for their capabilities in text-based interactions. However, natural human interaction often relies on speech, necessitating a shift towards voice-based models. A straightforward approach to achieve this involves a pipeline of "Automatic Speech Recognition (ASR) + LLM + Text-to-Speech (TTS)", where input speech is transcribed to text, processed by an LLM, and then converted back to speech. Despite being straightforward, this method suffers from inherent limitations, such as information loss during modality conversion, significant latency due to the complex pipeline, and error accumulation across the three stages. To address these issues, Speech Language Models (SpeechLMs)—end-to-end models that generate speech without converting from text—have emerged as a promising alternative. This survey paper provides the first comprehensive overview of recent methodologies for constructing SpeechLMs, detailing the key components of their architecture and the various training recipes integral to their development. Additionally, we systematically survey the various capabilities of SpeechLMs, categorize their evaluation metrics, and discuss the challenges and future research directions in this rapidly evolving field.**[1]**

Index Terms—**Speech Language Models, Speech Interaction, Large Language Models.**

I. INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities in generating text and performing a wide array of natural language processing tasks [1]–[3], serving as powerful foundation models for AI-driven language understanding and generation. Their success has also spurred numerous applications in various other domains, yet the reliance solely on text-based modalities presents a significant limitation. This leads to the development of speech-based generative models, which allow to interact with humans more naturally and intuitively. The inclusion of speech not only facilitates real-time voice interactions but also enriches communication by combining both text and speech information [4], [5].

Given the extensive mutual information between text and speech, it is natural to modify existing LLMs to enable speech interaction capabilities. A straightforward approach is to adopt an "Automatic Speech Recognition (ASR) + LLM + Text-to-Speech (TTS)" framework (Figure 1a) [6], [7]. In this setup, the user's spoken input is first processed by the ASR module,

Wenqian Cui (wenqian.cui@link.cuhk.edu.hk), Dianzhi Yu, and Irwin King (king@cse.cuhk.edu.hk) are with the Chinese University of Hong Kong, Hong Kong, China. Ziqiao Meng is with the National University of Singapore, Singapore. Xiaoqi Jiao and Guangyan Zhang are with the Lightspeed Studios, Tencent, China. Qichao Wang is with the AI Lab, Tencent, China. Yiwen Guo is an Independent Researcher.
[1]Github: https://github.com/dreamtheater123/Awesome-SpeechLM-Survey

which converts it into text. The LLM then generates a text response based on this transcription. Finally, the TTS module transforms the text response back into speech, which is played back to the user. However, this naive solution mainly suffers from the following three problems.

1) **Information loss.** Speech signals not only contain semantic information (i.e., the meaning of the speech) but also paralinguistic information (e.g., pitch, timbre, tonality, etc.). Putting a text-only LLM in the middle will cause the complete loss of paralinguistic information in the input speech [8], [9]. Recognizing paralinguistic features allows for a more engaging and immersive interaction, as the model can respond with greater context. Moreover, it enables the model to interpret a user's intent more accurately in certain situations, as the meaning of speech can vary depending on tone.

2) **Significant latency.** The sequential operation of ASR, LLM, and TTS leads to considerable delays due to the inherently **complex structures and pipelines** of these modules [9]–[11]. For instance, ASR often includes an additional text generator [12], [13], while TTS typically relies on a text tokenizer, both of which increase computational demands. Moreover, implementing advanced decoding techniques, such as beam search for ASR text decoding, can further contribute to delays [12], [13]. Consequently, developing an end-to-end speech generation model can help significantly minimize latency.

3) **Cumulative error.** A staged approach like this can easily lead to cumulative errors throughout the pipeline, particularly in the ASR-LLM stage [14], [15]. Specifically, transcription errors that occur when converting speech to text in the ASR module can negatively impact the language generation performance of the LLM. Additionally, the TTS performance can deteriorate significantly if the LLM generates text that cannot be synthesized. Consequently, developing a unified architecture plays a crucial role in reducing accumulated errors.

The limitations of the naive ASR + LLM + TTS framework have led to the development of Speech Language Models (SpeechLMs, Figure 1b). Unlike the naive framework, SpeechLMs directly encode speech waveforms into tokens or representations, capturing essential features and information from audio (section III-A). Although individual speech tokens may not carry word-level semantic meaning, they capture the semantic information of speech utterances and retain valuable paralinguistic information, which prevents

(a) Illustration of the "ASR + LLM + TTS" framework.
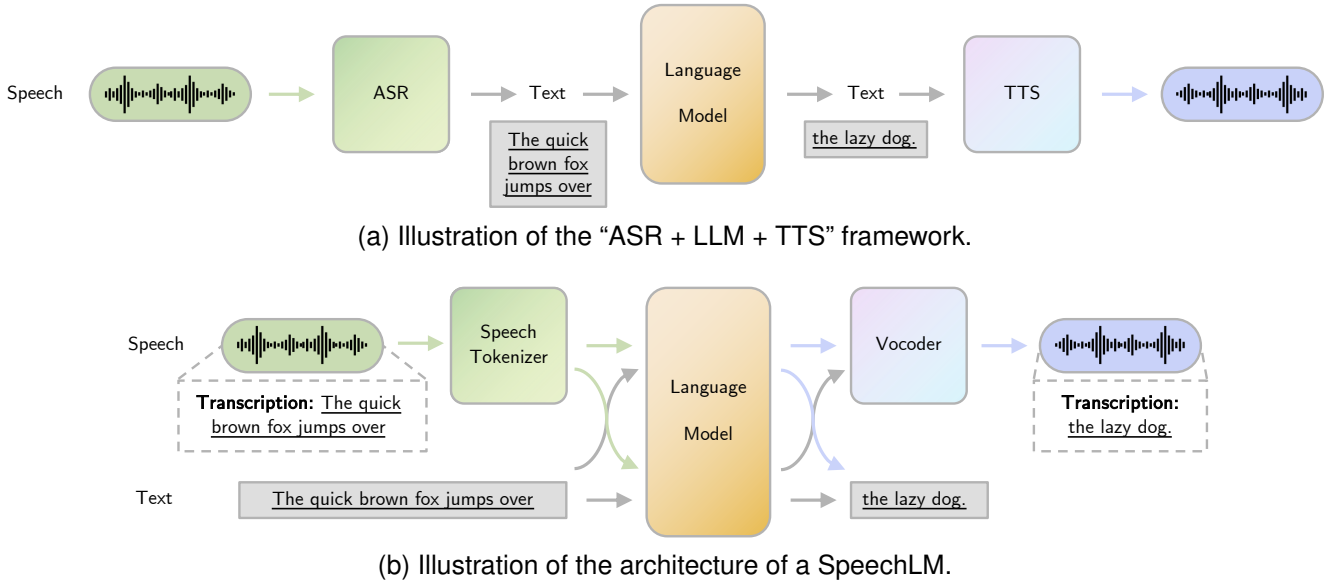
(b) Illustration of the architecture of a SpeechLM.

Fig. 1. Architectures of the "ASR + LLM + TTS" framework and a SpeechLM. We emphasize that, for SpeechLM, the same content can be used across both speech and text modalities, meaning that any input modality will yield any output modality of the same results. This intentional repetition of input/output contents in the figure highlights this point.

TABLE I
NOTATIONS USED IN THIS SURVEY.

Notation	Description
$\mathbf{a}$	A speech audio waveform
$\hat{\mathbf{a}}$	Model-reconstructed speech audio waveform
$\mathbf{t}$	A text span
$\mathbf{M}$	A multi-modal sequence containing speech and/or text
θ	Model parameters
$f_E(\cdot)$	Speech encoder
$d(\cdot)$	Speech quantizer
LM	Language model
$\mathbf{v}$	Encoded speech representations
$\mathbf{s}$	Speech tokens
$\mathbf{m}$	Multi-modal tokens (speech and/or text)
V	Vocabulary of a language model
E	Embedding matrix of a language model
$\mathbf{De}$	Transformer decoder blocks
Vo	Vocoder
G	Generator in a Generative Adversarial Network
D	Discriminator in a Generative Adversarial Network
ms	Mel-spectrogram of a speech audio waveform
F_0	Fundamental frequency

the information loss. SpeechLMs then model these tokens autoregressively, without solely relying on text input, which allows them to use the additional paralinguistic information to generate more expressive and nuanced speech (section III-B). Finally, the generated tokens are synthesized back to speech (section III-C). This integrated approach eliminates the need to chain together three separate modules, significantly reducing latency. Additionally, by working directly with the encoded speech tokens, SpeechLMs effectively mitigate the cumulative errors, as their training is integrated with the speech encoding, whereas the training of LLMs (language modeling) is completely independent of the ASR (speech recognition) module in the naive framework.

Speech Language Models (SpeechLMs) have the capability to go beyond simple conversational tasks and address more intricate and diverse applications. First, SpeechLMs can capture speaker-specific information and emotional nuances (Figure 2), which allows them to distinguish between different speakers during a conversation and to comprehend and generate speech imbued with specific emotional tones. Such advancements are crucial for applications in areas like personalized assistants, emotion-aware systems, and more nuanced human-computer interaction scenarios. Second, SpeechLMs can be designed to enable real-time voice interaction, where the model can be interrupted by humans or choose to speak while the user is still speaking, mimicking the dynamics of human conversations more naturally. Furthermore, because SpeechLMs are trained directly on speech data, they have the potential to facilitate communication in rare languages where spoken content is more prevalent than written material.

Contributions. In this survey, we present the first comprehensive overview of recent endeavors in constructing SpeechLMs. We explore the various components that constitute their architecture (section III) and the training recipes (section IV) involved in their development. we aim to elucidate the current state of the field by analyzing these models from the above perspectives. Additionally, we survey the downstream applications of SpeechLMs (section V), classify metrics to evaluate SpeechLMs (section VI), discuss the challenges encountered in this rapidly evolving area, and outline promising future research directions that could drive further advancements in SpeechLM technology (section VII). Our contributions are summarized as follows:

- We present the first survey in the field of SpeechLMs.
- We propose a novel taxonomy (Figure 3) of classifying SpeechLMs from the underlying components and the training recipes.
- We propose a novel classification system for the evaluation methods for SpeechLMs.
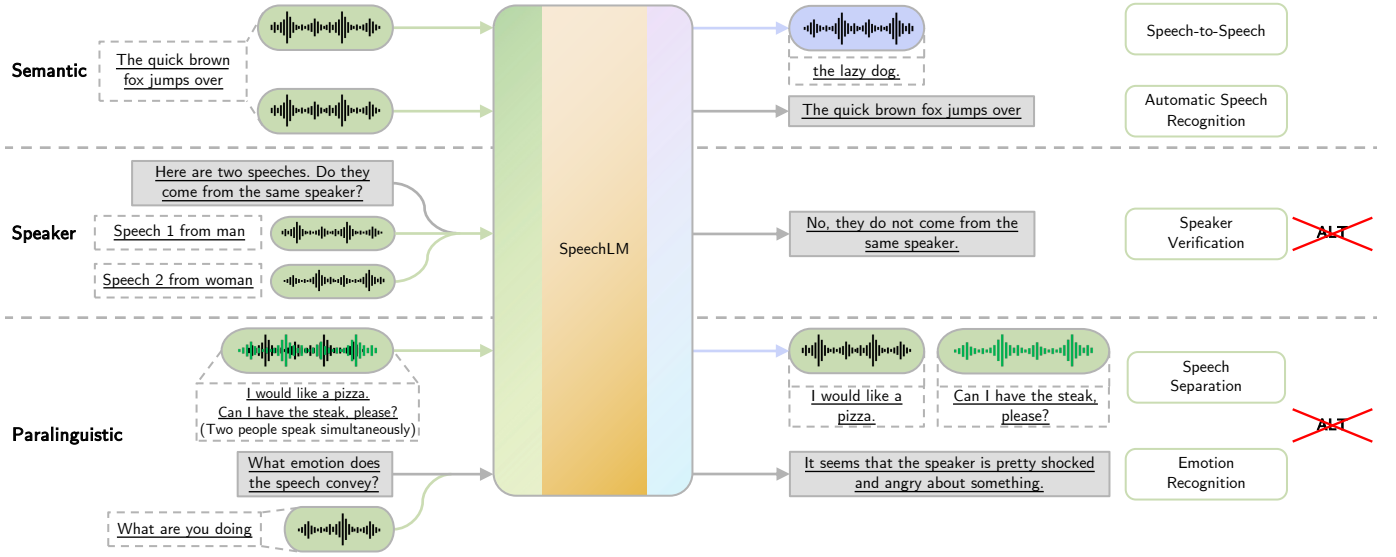- We identify several challenges in building SpeechLMs.

Fig. 2. Applications of a SpeechLM. We use ALT to represent the "ASR + LLM + TTS" framework.

II. PROBLEM FORMULATION

In this section, we provide a formal definition of Speech Language Models. A Speech Language Model (SpeechLM) is an autoregressive foundation model that processes and generates speech end-to-end, utilizing contextual understanding for coherent sequence generation. This capability allows it to perform a variety of tasks through speech-based interactions. Although SpeechLMs are required to perform end-to-end speech interactions, they can also incorporate text, enabling cross-modal functionalities such as speech-in-text-out and vice versa. We note that the concept of SpeechLM is in contrast to traditional text-based language models, such as LLM, where the only modality being processed within the model is text. Therefore, to avoid confusion, we call those text-based language models TextLMs throughout this survey.

We offer a unified framework in which SpeechLMs can process and generate speech data, text data, or even interleaved speech and text data. Specifically, a speech audio waveform $\mathbf{a} = (a_1, a_2, \ldots, a_Q)$ consists of a sequence of audio samples $a_i \in \mathbb{R}$ of length Q, where $1 \leq q \leq Q$. Similarly, a text span $\mathbf{t} = (t_1, t_2, \ldots, t_K)$ consists of a sequence of text tokens t_j (word, subword, character, etc.) of length K. Let $\mathbf{M} = (M_1, M_2, \ldots, M_N)$ denote a multimodal sequence of length N, where each element $M_i \in \{a_i, t_j\}$. We define $\mathbf{M}^{\text{in}} = (M_1^{\text{in}}, M_2^{\text{in}}, \ldots, M_{N_{\text{in}}}^{\text{in}})$ as the input multimodal sequence and $\mathbf{M}^{\text{out}} = (M_1^{\text{out}}, M_2^{\text{out}}, \ldots, M_{N_{\text{out}}}^{\text{out}})$ as the output multimodal sequence, where $N_{\text{in}} \geq 0$ and $N_{\text{out}} \geq 0$. Then, A SpeechLM parameterized by θ can then be represented as:

$$\mathbf{M}^{\text{out}} = SpeechLM(\mathbf{M}^{\text{in}}; \theta). \tag{1}$$

III. COMPONENTS IN SPEECHLM

There are three main components within a SpeechLM, namely speech tokenizer, language model, and token-to-speech synthesizer (vocoder), as illustrated in Figure 1. The fundamental reason for such a three-staged design pattern is to use the language modeling architecture (e.g., decoder-only transformer) to model speech autoregressively in the format of audio waveforms. Since both the input and output of a language model are tokens, additional modules need to be attached to the language model to handle the I/O format. Specifically, the speech tokenizer first transforms continuous audio waveforms into tokens or representations to serve as input to the language model, then the language model performs the next-token prediction based on the input speech tokens. Finally, the vocoder transforms the tokens outputted by the language model back into audio waveforms. We note that our focus here is on how the three components are grouped together to form a SpeechLM rather than a comprehensive overview of each component. Therefore, for speech tokenizer and vocoder, we mainly summarize the methods used in existing SpeechLMs. Table II summarizes the popular choices of the three components in various SpeechLM papers.

A. Speech Tokenizer

Speech tokenizer is the first component in SpeechLMs, which encodes continuous audio signals (waveforms) into tokens. Speech tokenizer aims to capture essential features of the audio while reducing its dimensionality and allows the audio input to be effectively processed by a language model for autoregressive generation. Speech tokenizer operates by encoding the audio segment by segment, producing two possible types of tokens (features): **discrete tokens** and **continuous tokens**. Discrete tokens (Section IV-A1) use a specific index to represent each speech segment, while continuous tokens (Section IV-A2) use an embedding to represent the segment[2]. Both token types can be utilized as input for a language model in autoregressive modeling. In this section, we categorize speech tokenizers based on their focus on modeling different aspects of the raw audio.

---

[2] For simplicity, we unify both types of representations as "tokens"

SpeechLM

- **Components (section III)**
  - **Speech Tokenizer (section III-A)**: Contrastive Predictive Coding [16], Wav2vec 2.0 [17], vq-wav2vec [18], w2v-bert [19], HuBERT [20], SpeechTokenizer [21], Mimi [9], Sound-Stream [22], Google USM [23], WavLM [24]
  - **Language Model (section III-B)**: Transformer [25], LLaMA [26], LLaMA 2 [27], LLaMA 3 [2], OPT [3], Qwen2 [28], GLM [29], Mixtral [30]
  - **Vocoder (section III-C)**: WaveNet [31], Tacotron 2 [32], WaveGlow [33], HiFi-GAN [34], Token-based HiFi-GAN [35], EnCodec [36]

- **Training Recipes (section IV)**
  - **Features Modeled (section IV-A)**
    - **Discrete Features (section IV-A1)**
      - **Semantic Tokens:** GSLM [37], TWIST [38], SpeechGPT [8], AudioPaLM [39], OmniFlatten [40], SLAM-Omni [41]
      - **Paralinguistic Token:** pGSLM [42], SPIRIT-LM [5]
      - **Acoustic Token:** VioLA [43], Li et al. [44], Parrot [45]
      - **Mixed Token:** Moshi [9], SpeechGPT-Gen [46]
    - **Continuous Features (section IV-A2)**: Spectron [47], tGSLM [48], Mini-Omni [10], LauraGPT [49], SLAM-Omni [41]
  - **Training Stages (section IV-B)**
    - **LM Pre-Training (section IV-B)**
      - **Cold Initialization:** GSLM [37], SUTLM [50], pGSLM [42], LSLM [51], VioLA [43]
      - **Continued Pre-Training:** TWIST [38], AudioPaLM [39], SPIRIT-LM [5], AudioChatLlama [14], Spectron [47], SLAM-Omni [41], OmniFlatten [40], Mini-Omni [10], Mini-Omni 2 [52], Freeze-Omni [53], Moshi [9], Parrot [45]
    - **LM Instruction-Tuning (section IV-B)**: SpeechGPT [8], SpeechGPT-Gen [46], COSMIC [54], Llama-Omni [11], Moshi [9]
  - **Speech Generation Paradigm (section IV-C)**
    - **Traditional Generation Paradigm (section IV-C)**: TWIST [38], SPIRIT-LM [5], AudioPaLM [39], SpeechGPT [8]
    - **Real-Time Interaction (section IV-C)**: dGSLM [4], Parrot [45], LSLM [51], VITA [55], Moshi [9], Mini-Omni 2 [52], OmniFlatten [40], SALMONN-omni [56]
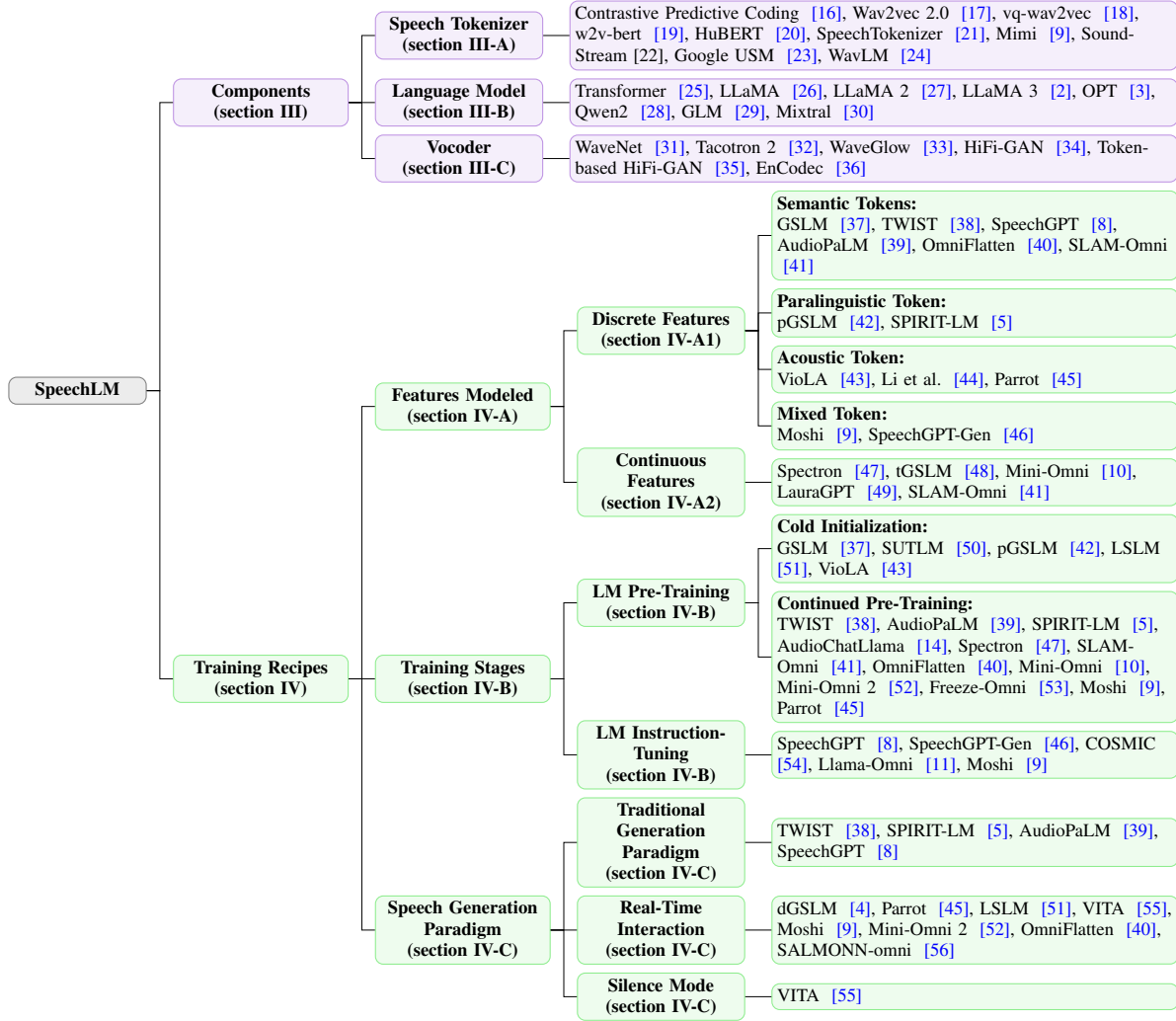   - **Silence Mode (section IV-C)**: VITA [55]

Fig. 3. Taxonomy of Speech Language Models.

1) Semantic Understanding Objective: Speech tokenizers designed with a semantic understanding objective aim to convert speech waveforms into tokens that accurately capture the content and meaning of the speech. These tokenizers focus on extracting semantic features from the waveforms, which enhances tasks like ASR.

A semantic understanding speech tokenizer typically comprises a speech encoder and a quantizer. The speech encoder encodes the essential information from the waveform into continuous embeddings. Then, a quantizer is typically incorporated to convert continuous embeddings into discrete indexes. Let $f_E(\cdot)$ denote the speech encoder parameterized by θ_{f_E}, we have $\mathbf{v} = f_E(\mathbf{a}; \theta_{f_E})$, where $\mathbf{v} = (v_1, v_2, \ldots, v_P)$ represents the encoded embeddings. Since $\mathbf{v}$ is still continuous, a quantizer $d(\cdot)$ is utilized to discretize the embeddings. Depending on different design choices, the speech tokens $\mathbf{s} = (s_1, s_2, \ldots, s_P)$ can either be derived from $\mathbf{a}$ or $\mathbf{v}$. Therefore, we have $\mathbf{s} = d(\mathbf{v}; \theta_d)$ or $\mathbf{s} = d(\mathbf{a}; \theta_d)$ for discrete tokens and $\mathbf{s} = \mathbf{v}$ for continuous tokens. After that, $\mathbf{s}$ can be used to train the speech tokenizer as a target label (such as masking $\mathbf{a}_{\text{mask}} \subset \mathbf{a}$ and reconstructing its corresponding label $\mathbf{s}_{\text{mask}} \subset \mathbf{s}$ [20]) or to train the following language model.

The key design choices lie in how to effectively encode (and quantize) speech into tokens. Wav2vec 2.0 [17] uses a convolutional encoder followed by a product quantization module [57] to discretize the continuous waveform. Then, a portion of the quantized representations is masked and modeled using a contrastive loss. W2v-BERT [19] is built upon wav2vec 2.0 and proposes to use Masked Language Modeling (MLM) loss [58] in addition to contrastive loss. Similarly, HuBERT [20] uses the k-means algorithm to cluster the speech utterances into a number of hidden units, and then perform MLM to predict the target hidden units from the masked speech utterances. To better align the representation of text and speech modalities, Google USM [23] utilizes text-injection loss [59] at the second pre-training stage to improve the performance and robustness of the downstream tasks. WavLM [24] adds the speech denoising objective during pre-training. While the majority of speech tokenizer studies focus on semantic-related tasks such as ASR and TTS, WavLM shows that speech denoising can boost the performance of non-semantic tasks such as speaker verification and speech separation. Section V presents a full list of downstream tasks.

2) Acoustic Generation Objective: Speech tokenizers with an acoustic generation objective focus on capturing the acoustic features necessary for generating high-quality speech waveforms. These tokenizers prioritize the preservation of essential acoustic characteristics over semantic content, making them suitable for speech (re)synthesis tasks.

To generate high-quality speech waveforms, acoustic generation speech tokenizers employ a speech synthesis or speech reconstruction objective. To achieve this, the architecture typically includes an encoder, a quantizer, and a decoder. Same as before, the encoder $f_E(\cdot)$ and quantizer $d(\cdot)$ transform the original waveform into tokens. After that, the decoder $f_D(\cdot)$ reconstructs these tokens back into speech waveforms. This process is represented by $\hat{\mathbf{a}} = f_D(\mathbf{s}; \theta_{f_E})$, where $\hat{\mathbf{a}}$ is the generated or reconstructed waveform.

Neural audio codecs are very suitable for and are primarily employed as acoustic generation speech tokenizers [22], [36]. These codecs utilize the advanced modeling capabilities of deep neural networks to compress audio waveforms into a compact representation, typically in the form of discrete tokens. Using the encoder-quantizer-decoder architecture, the encoder compresses the audio into latent representations, the quantizer discretizes these representations (commonly through vector quantization (VQ) [60] or residual vector quantization (RVQ) [22]), and the decoder reconstructs the discrete tokens back into audio waveforms. Therefore, the encoder and/or the quantizer are utilized as an acoustic speech tokenizer.

3) Mixed Objective: Speech tokenizers with a mixed objective aim to balance both semantic understanding and acoustic generation. The goal is to harness the advantages of both types of tokenizers. Currently, the development of these tokenizers is in its early stages. Most existing mixed speech tokenizers primarily adopt the architecture of acoustic generation speech tokenizers and focus on distilling information from semantic tokenizers into the acoustic tokenizer. SpeechTokenizer [21] utilizes the RVQ-GAN [22], [36] architecture, distilling semantic information from HuBERT [20] to the first layer of RVQ. Inspired by SpeechTokenizer, Mimi [9] employs a single VQ to extract information from WavLM [24] and incorporates another RVQ module to learn the acoustic information.

B. Language Model

Due to the success of TextLMs [1], [2], [86], most SpeechLMs follow their architectures. They primarily employ transformers [25] or decoder-only architectures (such as OPT [3], LLaMA [26]) to generate speech in an autoregressive manner. To formally define it, given $|V_t|$ as the vocabulary size and h as the hidden dimension, a typical text-based decoder-only transformer language model consists of an embedding matrix $E_t \in \mathbb{R}^{|V_t| \times h}$, a sequence of L transformer decoder blocks $\mathbf{De} = \{De_1, De_2, \ldots, De_L\}$, and an output embedding matrix $E_t' \in \mathbb{R}^{h \times |V_t|}$. Therefore, the language model (LM) can be represented as

$$\mathbf{t}^{\text{out}} \sim \text{LM}(\mathbf{t}^{\text{in}}, (E_t, \mathbf{De}, E_t')). \tag{2}$$

To adapt the language model to generate speech, the original text tokenizer is changed to the speech tokenizers illustrated in section III-A. When using **discrete tokens**, $E_t \in \mathbb{R}^{|V_t| \times h}$ is changed to a speech embedding matrix $E_s \in \mathbb{R}^{|V_s| \times h}$, where $|V_s|$ represents the vocabulary size of the speech tokenizer. The output embedding matrix is also changed from $E_t' \in \mathbb{R}^{h \times |V_t|}$ to $E_s' \in \mathbb{R}^{h \times |V_s|}$. As a result, the language model in a SpeechLM is represented as

$$\mathbf{s}^{\text{out}} \sim \text{LM}(\mathbf{s}^{\text{in}}, (E_s, \mathbf{De}, E_s')). \tag{3}$$

Because the language model architecture of SpeechLMs is borrowed from TextLMs, it is natural that the resulting model can jointly model both text and speech modalities [5], [8]. To achieve this, a naive and most adopted approach is to expand the vocabulary of the original TextLM to incorporate both text and speech tokens. Specifically, the speech embedding matrix is usually appended to the end of the text embedding matrix, resulting in a larger embedding matrix $E_m \in \mathbb{R}^{(|V_t|+|V_s|) \times h}$. Let $\mathbf{m}$ be a token sequence containing both speech and text tokens, the resulting language model becomes

$$\mathbf{m}^{\text{out}} \sim \text{LM}(\mathbf{m}^{\text{in}}, (E_j, \mathbf{De}, E_j')). \tag{4}$$

By doing so, the model can generate both text and speech in a single sequence, enabling much more diverse applications (see section V). In contrast, when modeling with **continuous tokens**, the embeddings derived from the speech tokenizer are directly fed into the language model. In this case, the architecture of the language model remains unchanged.

C. Token-to-Speech Synthesizer (Vocoder)

After the tokens have been autoregressively generated by the language model component, a token-to-speech module, often known as vocoder, is utilized to synthesize all the speech tokens back into speech waveforms. This process involves converting the linguistic and paralinguistic information represented by the generated speech tokens into audio waveforms that can be heard. This can be seen as a reverse process to the speech tokenizer and therefore can be represented as

$$\mathbf{a} = Vo(\mathbf{s}; \theta_{V_o}), \tag{5}$$

where Vo is the vocoder model parameterized by θ_{V_o}.

The pipeline of the SpeechLM vocoder can vary depending on the underlying vocoder model. There are two main pipelines: Direct synthesis and input-enhanced synthesis. **Direct synthesis** is the pipeline where the vocoder directly converts speech tokens generated by the language model into audio waveforms. For example, Polyak *et al.* [35] adapts the HiFi-GAN [34] architecture and takes speech tokens as inputs. In contrast, **input-enhanced synthesis** employs an additional module to transform the tokens into a continuous latent representation before they are fed into the vocoder [87], [88]. The main reason for using this pipeline is that vocoders typically require intermediate audio representations, such as mel-spectrograms [34], [89], [90], as input. For example, CosyVoice [61] introduces a Conditional Flow-Matching (CFM) model to convert speech tokens into mel-spectrogram, and then leverages a HiFi-GAN to synthesize the final waveform. When comparing the two pipelines, direct synthesis is generally simpler and faster than input-enhanced synthesis.

TABLE II
SUMMARIZATION OF THE ARCHITECTURAL CHOICE OF SPEECH TOKENIZER, LANGUAGE MODEL, AND VOCODER IN POPULAR SPEECHLMS. "-"
REPRESENTS NON-EXISTENCE OR NOT INDICATED, * MEANS THE ARCHITECTURE IS MAINLY BASED ON THE WRITTEN ONE, "A, B" MEANS THE
AUTHORS EXPERIMENTED WITH BOTH A AND B AS THE COMPONENT, AND "A + B" MEANS "A" AND "B" ARE COMBINED TO SERVE AS THE
COMPONENT.

Approach	Speech Tokenizer	Language Model	Vocoder
SLAM-Omni [41]	Whisper Encoder [12] + Linear Projector	Qwen2 [28]	-
OmniFlatten [40]	CosyVoice Encoder [61]	Qwen2	CosyVoice Decoder
SyncLLM [62]	HuBERT [20]	LLaMA-3 [2]	HiFi-GAN [34], [35]
EMOVA [63]	SPIRAL [64]	LLaMA-3	VITS [65]
Freeze-Omni [53]	Transformer [25]	Qwen2	TiCodec [66]
IntrinsicVoice [67]	HuBERT	Qwen2	HiFi-GAN
Mini-Omni2 [52]	Whisper	Qwen2	Mini-Omni [10]
SALMONN-omni [56]	Mamba Streaming Encoder [68]	-	VoiceCraft [69] + Codec Decoder
Zeng et al. [70]	Whisper + ASR	GLM [29]	CosyVoice
Parrot [45]	VQ-VAE	LLaMA-3, Mistral, Gemma 2	HiFi-GAN
GPST [71]	EnCodec [36]	Transformer	Codec Decoder
Moshi [9]	Mimi [9]	Transformer*	Mimi
VITA [55]	CNN + Transformer + MLP [55]	Mixtral [30]	Text-to-Speech Toolkit [55]
LSLM [51]	vq-wav2vec [18]	Decoder-Only Transformer	UniVATS [72]
SPIRIT-LM [5]	HuBERT, VQ-VAE [60], speechprop	LLaMA-2 [27]	HiFi-GAN
TWIST [38]	HuBERT	OPT [3], LLaMA [26]	HiFi-GAN
VOXTLM [73]	HuBERT	OPT [3]	HiFi-GAN
Voicebox [74]	EnCodec	Transformer* [25]	HiFi-GAN
VioLA [43]	EnCodec	Transformer*	Codec Decoder [36]
FunAudioLLM [75]	SAN-M [76]	Transformer*	HiFTNet [77]
SpeechGPT-Gen [46]	SpeechTokenizer [21]	LLaMA-2	SpeechTokenizer decoder [21]
LauraGPT [49]	Conformer*	Qwen [78]	Transformer + Codec Decoder
Spectron [47]	Conformer*	PaLM 2* [79]	WaveFit [80]
AudioLM [81]	w2v-BERT [19]	Decoder-Only Transformer*	SoundStream* [22]
UniAudio [82]	EnCodec, Hifi-codec [83], Improved RVQGAN [84]	Transformer*	Codec Decoder
Llama-Omni [11]	Whisper	LLaMA-3.1	HiFi-GAN
Mini-Omni [10]	Whisper + ASR Adapter [10]	Qwen2	TTS Adapter [10]
tGSLM [48]	Segmentation + SSE [85] + Lexical embedder	Transformer*	Tacotron-2 + Waveglow [32], [33]
SpeechGPT [8]	HuBERT	LLaMA	HiFi-GAN
dGSLM [4]	HuBERT	Dialogue Transformer [4]	HiFi-GAN
SUTLM [50]	HuBERT	Transformer*	-
pGSLM [42]	HuBERT	MS-TLM [42]	HiFi-GAN
GSLM [37]	HuBERT, CPC [16], Wav2vec 2.0 [17]	Transformer*	Tacotron-2 + Waveglow

However, the choice of pipeline depends on the type of tokens used as input. Tokens from acoustic generation tokenizers contain sufficient acoustic information, making them suitable for direct synthesis. Conversely, tokens from semantic understanding tokenizers provide rich semantic information but lack fine acoustic details, particularly in higher frequencies. Therefore, these tokens are better enhanced into an acoustic-rich representation, such as mel-spectrograms, before synthesizing the final speech.

Vocoders can be categorized by their architectural choice. In the following sections, we summarize vocoders that are mostly adopted in the development of SpeechLMs.

1) GAN-based Vocoder: Generative Adversarial Network (GAN) is the most adopted architecture of the vocoders [34], [35], [89]–[91]. It is well known for its fast and high-fidelity generation in speech synthesis tasks. The architecture of GAN includes a generator and a discriminator. Specifically, the generator creates realistic audio waveforms from random noise or input features, while the discriminator evaluates the authenticity of the generated audio against real audio samples.

To utilize GAN to synthesize high-fidelity speech, various training objectives are designed, focusing on different aspects. First, **GAN loss** is utilized as the fundamental objective for the operation of the generator and the discriminator. Specifically, the typical choice of GAN loss for the generator (G) and discriminator (D) is to use the least squares loss function. The GAN loss for the generator ($\mathcal{L}_{\text{GAN}}(G; D)$) and the discriminator ($\mathcal{L}_{\text{GAN}}(D; G)$) are

$$\mathcal{L}_{\text{GAN}}(G; D) = \mathbb{E}_{ms}\left[(D(G(ms)) - 1)^2\right] \quad (6)$$

and

$$\mathcal{L}_{\text{GAN}}(D; G) = \mathbb{E}_{(x,ms)}\left[(D(x) - 1)^2 + (D(G(ms)))^2\right], \quad (7)$$

respectively. In these loss functions, x represents the ground truth audio and ms represents its mel-spectrogram. Second, most GAN-based vocoders synthesize speech waveform from mel-spectrograms, so **mel-spectrogram loss** is proposed to align the mel-spectrogram synthesized by the generator and the mel-spectrogram transformed from the ground-truth waveform, in order to improve the fidelity of the generated speech. Mel-spectrogram loss ($\mathcal{L}_{\text{Mel}}(G)$) works by minimizing the L1 distance between the two versions of mel-spectrograms mentioned above. Its formula is shown below:

$$\mathcal{L}_{\text{Mel}}(G) = \mathbb{E}_{(x,ms)}\left[\|\phi(x) - \phi(G(ms))\|_1\right], \quad (8)$$

where $\phi(\cdot)$ is the function to transform a waveform into the corresponding mel-spectrogram. Third, to further enhance the generation fidelity, **feature matching loss** ($\mathcal{L}_{FM}(G; D)$) is proposed to align the discriminator-encoded features of the ground truth sample and the generated sample with L1 distance, which has the following formula:

$$\mathcal{L}_{FM}(G; D) = \mathbb{E}_{(x,ms)}\left[\sum_{i=1}^{T} \frac{1}{N_i} \left\|D^i(x) - D^i(G(ms))\right\|_1\right], \quad (9)$$

where $D^i(\cdot)$ and N_i denote the features and the number of features in the i-th layer of the discriminator, respectively.

For architectural choices, GAN-based vocoders focus on injecting inductive biases to generate audio waveforms. Mel-GAN [89] adds residual blocks with dilations in the generator

to model the long-range correlation among the audio time steps and proposes a multi-scale architecture for the discriminator to model the different frequency ranges of the audio. Based on the idea of the multi-scale discriminator, HiFi-GAN [34] proposes a multi-period discriminator to model the diverse periodic patterns within the audio waveforms. To preserve high-frequency content, Fre-GAN [91] employs the Discrete Wavelet Transform (DWT) to downsample and learn spectral distributions across multiple frequency bands. Unlike traditional approaches like Average Pooling (AP), DWT efficiently decomposes the signal into low-frequency and high-frequency sub-bands. BigVGAN [90] introduces a periodic activation function called snake function along with an anti-aliased representation to reduce the high-frequency artifacts in the synthesized audio.

2) GAN-based Neural Audio Codec: Given that many neural audio codecs employ a GAN architecture, they can be effectively discussed within the context of GAN-based vocoders. In contrast to speech tokenizers, the decoder in the codec is leveraged as the vocoder [22], [36]. Polyak *et al.* [35] utilizes HiFi-GAN [34] as the vocoder backbone and proposes to disentangle the input features of a vocoder into distinct properties [35], which include semantic tokens, pitch tokens, and speaker embeddings. Such a design choice enables the codec to better perform on pitch and speaker-related tasks such as voice conversion and F_0 manipulation.

3) Other Types of Vocoder: The variety of vocoders is not restricted to the ones mentioned earlier, as those are the ones commonly employed in SpeechLMs. This section briefly outlines other potential vocoder types that are seldom explored as a component in SpeechLMs.

Pure Signal Processing Vocoder. Pure signal processing vocoders are traditional methods relying on deterministic algorithms rather than deep learning models to synthesize speech [92], [93]. However, this kind of vocoder introduces noticeable artifacts in the synthesized audio and is rarely used.

Autoregressive Vocoder. Autoregressive vocoders generate audio waveforms one sample at a time, with each sample conditioned on the previously generated samples [31]. This approach allows for high-quality audio synthesis due to its sequential nature and the ability to capture intricate temporal dependencies within the audio signal. However, the sequential generation process can be computationally expensive and time-consuming, making autoregressive models less efficient compared to parallelized methods like GAN-based vocoders.

Flow-based Vocoder. Flow-based vocoder aims to establish a series of invertible transformations that map a simple distribution, such as a Gaussian, to the complex distribution of audio samples. This mechanism allows for efficient sampling and density evaluation, enabling the model to synthesize audio in parallel rather than sequentially, which significantly enhances both speed and quality [33]. Compared to GAN-based vocoders, Flow-based vocoders typically need more parameters and memory to train the model, which hinders them from being effectively utilized [89].

VAE-based Vocoders. Variational Autoencoders (VAEs) are powerful generative models that learn to encode input data into a compressed latent space while allowing for the

Dataset	Type	Phase	Hours	Year
LibriSpeech [98]	ASR	Pre-Training	1k	2015
Multilingual LibriSpeech [99]	ASR	Pre-Training	50.5k	2020
LibriLight [100]	ASR	Pre-Training	60k	2019
People dataset [101]	ASR	Pre-Training	30k	2021
VoxPopuli [102]	ASR	Pre-Training	1.6k	2021
Gigaspeech [103]	ASR	Pre-Training	40k	2021
Common Voice [104]	ASR	Pre-Training	2.5k	2019
VCTK [105]	ASR	Pre-Training	0.3k	2017
WenetSpeech [106]	ASR	Pre-Training	22k	2022
LibriTTS [107]	TTS	Pre-Training	0.6k	2019
CoVoST2 [108]	S2TT	Pre-Training	2.8k	2020
CVSS [109]	S2ST	Pre-Training	1.9k	2022
VoxCeleb [110]	Speaker Identification	Pre-Training	0.4k	2017
VoxCeleb2 [111]	Speaker Identification	Pre-Training	2.4k	2018
Spotify Podcasts [112]	Podcast	Pre-Training	47k	2020
Fisher [113]	Telephone conversation	Pre-Training	2k	2004
SpeechInstruct* [8]	Instruction-following	Instruction-Tuning	-	2023
InstructS2S-200K* [11]	Instruction-following	Instruction-Tuning	-	2024
VoiceAssistant-400K* [10]	Instruction-following	Instruction-Tuning	-	2024

reconstruction of the original data [60], [94]. However, VAE is seldom explored as the underlying architecture of vocoders.

Diffusion-based Vocoder. Diffusion models have emerged in recent years as a powerful class of generative models that can be used for high-fidelity speech synthesis. They work by gradually adding noise to the input data (e.g. audio waveforms) to create a sequence of increasingly noisy representations, then learning to reverse this process to generate new samples [95]–[97]. For instance, DiffWave [95] uses Denoising Diffusion Probabilistic Models (DDPM) to synthesize audio.

IV. TRAINING RECIPES

In this section, we categorize and summarize the commonly used training recipes found in recent SpeechLM papers. This includes an overview of the types of features modeled in SpeechLMs, the various training stages along with the techniques employed in each stage, and the different paradigms for generating speech.

A. Features Modeled

The features modeled refer to the types of features or tokens outputted by the speech tokenizer and modeled by the language model component within a SpeechLM. These features play a crucial role in determining the capabilities and performance of SpeechLMs. Different features model the speech waveforms from different aspects. In this section, we summarize commonly used features in SpeechLMs and focus on how different features affect the performance of SpeechLMs. Based on recent developments, we can categorize the features modeled by SpeechLMs into two main types: discrete features and continuous features.

1) Discrete Features: Discrete features (or discrete tokens) refer to quantized representations of speech signals that can be represented as distinct, countable units or tokens. These features are typically derived from speech signals through various encoding and quantization processes, resulting in a finite set of possible values. Discrete features are the most

used features by SpeechLMs as they can be represented as tokens and be modeled exactly the same as the text tokens within a TextLM.

Most SpeechLMs only employ *semantic tokens* (generated by semantic understanding tokenizers, Section III-A1) to represent speech, as semantic information plays the most crucial role in spoken communication. GSLM [37], the first-ever SpeechLM, compares three tokenizers, which include Contrastive Predictive Coding (CPC) [16], wav2vec 2.0 [17], and HuBERT [20]. It concludes that HuBERT performs the best on various tasks such as speech resynthesis and speech generation. A large number of works follow this setting and use HuBERT as the speech tokenizer [5], [8], [38]. AudioPaLM [39] experiments the choice between w2v-bert [19], USM-v1 [23], and USM-v2 [39] (a modified version of USM-v1), and it concludes that USM-v2 is the best-performing speech tokenizer on ASR and Speech Translation (ST) tasks.

Although semantic tokens excel at generating semantically meaningful speech because of the modeling of the contextual information within speech waveforms, researchers find out that the speech generated solely upon semantic tokens lacks expressive information such as prosody and different pitches or timbres [5], [114]. To conquer this limitation, *paralinguistic tokens* can be integrated into the modeling process to capture expressive information with speeches. pGSLM [42] proposes to use the fundamental frequency (F0) and unit duration as prosody features to complement the HuBERT semantic tokens, and trains a multi-stream transformer language model to predict the semantic tokens, pitch (F0), and unit duration separately. Similarly, SPIRIT-LM [5] complements the HuBERT semantic tokens with pitch and style tokens [115]. This incorporation of extra acoustic tokens allows SpeechLMs to more effectively capture expressive elements without significantly compromising semantic understanding [5].

Another type is *acoustic tokens*, which are tokens aiming to capture essential acoustic features to reconstruct high-fidelity speech, primarily obtained from neural audio codec models (Section III-A2). Some studies directly model the codec tokens in a language model, which is often regarded as Codec Language Models (CodecLMs). For example, Viola [43] trains a CodecLM capable of performing ASR, TTS, and Machine Translation. Parrot [45] trains on VQ-VAE [60] tokens for modeling dual-channel spoken dialogue data.

Discussion. Different types of tokens influence the speech quality of SpeechLMs in different ways, often resulting in trade-offs [81]. For example, while semantic tokens align well with text and excel in producing semantically coherent speech, the generated speech often lacks acoustic details, such as high-frequency information. Recovering and enhancing these details typically requires post-processing, like a diffusion model, which significantly increases the model's latency. Conversely, acoustic tokens can facilitate the generation of high-fidelity audio but often struggle with inaccuracies in content generation [21]. Researchers have tried two ways to balance these trade-offs. The first involves combining semantic and acoustic tokens into a single sequence. AudioLM [81] proposes a hierarchical modeling scheme that first models semantic tokens from w2v-bert [19] and then uses these tokens to predict acoustic tokens

from SoundStream [22], which ultimately generates speech. However, this kind of approach increases sequence length, which increases modeling complexity. The second strategy leverages *mixed tokens* (Section III-A3) to jointly model semantic and acoustic information, showing promising results in Moshi [9] and SpeechGPT-Gen [46].

2) Continuous Features: Continuous features (or continuous tokens), in contrast to discrete features, are unquantized, real-valued representations of speech signals that exist on a continuous scale (continuous tokens). Continuous features can include spectral representations like mel-spectrograms or latent representations extracted from neural networks. Spectron [47] performs speech continuation by predicting the spectrograms frame-by-frame. Mini-Omni [10] and SLAM-Omni [41] extract intermediate representations from a frozen Whisper encoder as input for the SpeechLM, whereas LauraGPT [49] employs an audio encoder trained alongside the language model to derive latent representations from input speech. Continuous features can capture fine-grained, nuanced aspects of speech that may be lost in discretization processes. However, utilizing these features often necessitates modifying the off-the-shelf training pipeline of language models, as traditional text-based models are built to handle discrete units. Moreover, continuous features demand more storage capacity compared to their discrete counterparts.

B. Training Stages

Training a SpeechLM involves training the three main components: speech tokenizer, language model, and vocoder. Similar to TextLMs, the key to training SpeechLMs lies in effectively modeling speech continuation, which is primarily the responsibility of the language model. The speech tokenizer and vocoder usually rely on established methods and are trained using distinct training datasets specific to each SpeechLM approach. Therefore, This section reviews the main techniques used to train the language model component. Following TextLMs, we divide the training process of SpeechLMs into three stages, including pre-training, instruction-tuning, and post-alignment.

1) Language Model Pre-Training: The pre-training of the language model in SpeechLMs is a critical phase that significantly influences the model's ability to generate coherent and contextually relevant speech. This phase typically involves training the language model to autoregressively predict the next token on a large corpus of speech tokens. The primary objective during this stage is to learn the statistical patterns and dependencies inherent in the speech data, enabling the model to predict the next token in a sequence based on the preceding context.

Training data. SpeechLMs pre-training mainly leverages large-scale open-sourced speech data. Commonly used datasets include those for ASR [98], [100]–[102], TTS [107], ST [102], [109], podcasts [112], and dialogues [113]. Table III includes popular datasets used in the pre-training stage. Some datasets consist solely of speech data, while others include both speech and corresponding text transcripts. The inclusion of text transcripts can enhance the model's representation by

TABLE IV
FOUR DIFFERENT METHODS OF MODELING SPEECH AND TEXT TOKENS.

Modeling Method	Example	Explanation
Speech-only	[SPEECH] S12 S34 S33 ... S11 S59	Only the speech sequence is provided.
Text-only	[TEXT] A quick brown fox jumps over a lazy dog.	Only the text sequence is provided.
Concatenated speech-text	[SPEECH] S12 S34 S33 ... S11 S59 [TEXT] A quick brown fox jumps over a lazy dog.	The speech sequence and text sequence are concatenated together.
Alternating speech-text	[SPEECH] S12 S34 S33 [TEXT] brown fox jumps over a lazy [SPEECH] S11 S59	The sequence is interleaved with speech and text tokens.

allowing it to learn the relationship between spoken language and its written form, which will be discussed later.

Cold Initialization. Some SpeechLMs use cold initialization during the pre-training phase, where model parameters are initialized randomly. The pioneering SpeechLM—GSLM [37]—trained a transformer [25] from scratch to serve as the language model. This study demonstrated the effectiveness of the SpeechLM pipeline and compared performance across various speech tokenizer options. They found that HuBERT [20] outperformed CPC [16] and wav2vec 2.0 [17] in understanding speech content and generating natural speech. SUTLM [50] also uses a transformer as the language model. They studied the critical problem of jointly modeling speech and text tokens by comparing four different modeling methods: speech-only, text-only, concatenated speech-text, and alternating (interleaving) speech-text. They showed that the setting of alternating speech-text performs the best in cross-modal evaluations. Table IV illustrates the four modeling methods.

Some works leverage a different architecture from the standard transformer. These models are typically trained from scratch when the architecture deviates from a standard transformer or TextLM too much. For example, pGSLM [42] proposes a multi-stream transformer language model (MS-TLM) that takes multiple streams of input and predicts multiple streams of output to generate speech units, duration, and pitch embeddings simultaneously. dGSLM [4] introduced a dialogue transformer language model (DLM) to jointly model the dialogue speech data from the two speakers. To enable the listening ability of SpeechLMs while speaking, LSLM [51] proposes to attach a streaming self-supervised learning (SSL) Encoder to an autoregressive token-based TTS Model.

Continued Pre-Training. In contrast to cold initialization, continued Pre-Training involves initializing the language model with pre-trained weights from a TextLM and then adapting it to handle speech tokens. This approach leverages the linguistic knowledge embedded in TextLMs, allowing for more efficient and effective training of SpeechLMs. Hassid *et al.* [38] found that starting with a textually pre-trained language model (OPT [3] and LLaMA [26]) can enhance the model's convergence rate and significantly improve its speech understanding capabilities. They also demonstrated that while training from text-pretrained checkpoints outperforms cold initialization, training from image-pretrained checkpoints yields poorer results compared to cold initialization. This indicates that not all pre-trained checkpoints are equally effective. Additionally, AudioPaLM [39] trained the SpeechLM using PaLM and PaLM-2 [116], [117], showing that the

SpeechLM benefits from both an increased size of the pre-trained checkpoint and a larger training dataset.

The performance of SpeechLMs can be further enhanced by **aligning the text and speech modality representations.** Some works align the text and speech representations in a **single sequence**. SPIRIT-LM [5] found that continually pretraining on TextLM checkpoints using interleaving text and speech tokens can significantly boost the model's performance on speech understanding and generation. Additionally, their visualizations demonstrate that the similarity between text and speech features is notably higher in models trained with interleaved token sequences compared to those trained without this approach. Spectron [47] solve the text-speech representation alignment problem by jointly supervising multiple objectives. Specifically, the input speech prompt is first transcribed into its text tokens, and then the model predicts the text token response. Finally, the text response is synthesized to output speech. SpeechGPT [8] also adopts this concept but applies it during the instruction-tuning phase. Some other methods perform **multi-sequence** representation alignment. This approach simultaneously generates the text sequence and speech sequence(s). For example, Llama-Omni uses the LLM output hidden state to decode text tokens and generate discrete speech tokens simultaneously. Mini-Omni [10] generates a single sequence of text tokens and seven sequences of acoustic tokens in parallel, all aligned at the sentence level. Similarly, Moshi [9] generates one sequence of text tokens, one sequence of semantic tokens, and seven sequences of acoustic tokens in parallel, which are aligned at the word level.

Discussion. The primary goal of aligning text and speech representations is to leverage the strengths of text-based models to enhance speech-based models. Researchers have found that training a SpeechLM is significantly more challenging than training a TextLM. This difficulty arises because text serves as a concentrated form of knowledge, while speech requires models to independently learn the rules of spoken language. Aligning text and speech representations has demonstrated effectiveness, but it involves various trade-offs. First, text primarily conveys semantic information, which can improve a SpeechLM's semantic modeling capabilities but may compromise its ability to capture paralinguistic features, such as tone and emotion, during alignment. Second, there are two main inference approaches for the aligned models: text-present and text-independent. Text-present inference decodes text and speech simultaneously, which may increase latency but enhances the SpeechLM's reasoning abilities [10] and reduces possible hallucinations [9]. Conversely, text-independent

inference is more efficient but may lack stability. Furthermore, the question of whether to incorporate text modality to enhance SpeechLM performance remains an open question, especially considering that humans typically acquire spoken language skills before mastering written language.

2) Language Model Instruction-Tuning: Instruction-tuning refers to the process of fine-tuning SpeechLMs to follow specific instructions to perform a wide range of tasks. This phase is crucial for enhancing the pre-trained model's generalization capabilities and making it more adaptable to diverse applications. Therefore, the key focus is on creating effective instruction-following datasets.

Several approaches have been proposed to construct instruction-following datasets for SpeechLMs. SpeechGPT [8] and SpeechGPT-Gen [46] propose a two-stage instruction-tuning, including cross-modal instruction fine-tuning and chain-of-modality instruction fine-tuning. In the first stage, instruction data are generated based on ASR datasets by appending the instruction to paired ASR data, asking the model to convert speech into text. Similarly, paired data is also used to create instruction data for performing TTS. In the second stage, they construct a speech-in-speech-out dataset by transforming a text-based instruction-following dataset using TTS. Llama-Omni [11] also creates instruction-following data by synthesizing text-based datasets, adhering to specific constraints. First, they transform the input text prompt into a format that mimics natural speech patterns. Next, they discard the original text response and employ a TextLM to generate answers to the converted prompts, ensuring these responses also follow natural speech patterns. Finally, they synthesize the prompt/response pairs using TTS. COSMIC [54] constructed speech QA data by asking GPT-3.5 to generate question-answer pairs based on the transcriptions of English TED talk speeches. They showed the model trained on their proposed speech QA dataset can generalize to unseen tasks such as speech-to-text translation using in-context learning.

3) Language Model Post-Alignment: Post-alignment is the critical process of refining a language model's behavior to align with human preferences, ensuring that its outputs are both safe and reliable. This stage is typically regarded as the final phase of language model training. It often employs techniques like Reinforcement Learning from Human Feedback (RLHF), specifically methods such as Proximal Policy Optimization (PPO) [118] and Direct Preference Optimization (DPO) [119].

Post-alignment in SpeechLMs focuses on addressing the unique challenges inherent in the speech interaction pipeline. Align-SLM [120] identifies that SpeechLMs often generate inconsistent semantic content when given the same prompt. It tackles this by using a TextLM to choose the preferred response from SpeechLMs after transcribing them via ASR and then aligning those preferences using DPO. On the other hand, SpeechAlign [121] concentrates on the acoustic quality of SpeechLMs. It observes that differences between the "golden" speech tokens and those generated by the language model lead to subpar acoustic quality in the generated speech as the vocoder synthesizes speech from generated tokens during inference. To mitigate this, it employs various optimization

techniques to align the language model's output with the distribution of the "golden" tokens. Despite its importance, the post-alignment of SpeechLMs remains under-explored. A critical application of post-alignment is to mitigate the safety risks associated with generative models. Thus, future research should prioritize identifying and addressing the unique safety challenges posed by SpeechLMs (see Section VII-D).

C. Speech Generation Paradigm

Most approaches covered in earlier sections follow the traditional generation paradigm of SpeechLMs, which involves taking a predefined input sequence and generating a complete response. However, this approach does not reflect the natural flow of voice interactions. For instance, during a conversation, one person may interrupt another, switching from listening to speaking. Additionally, a person might choose not to respond if the other is engaged in a conversation with someone else. Based on these observations, we identify two key aspects of advanced voice interaction skills for SpeechLMs: real-time interaction and interactive period recognition.

Real-time Interaction refers to the capability of SpeechLMs to engage with users instantaneously. This interaction consists of two key components. 1) User Interruption: SpeechLMs should be able to be interrupted by users and should respond appropriately to new instructions provided during the conversation. 2) Simultaneous Response: SpeechLMs should be capable of generating responses while the user is still speaking. This requires the model to effectively perform speech understanding (processing input) and speech generation (producing output) simultaneously. Some literature refers to the ability as full-duplex modeling. dGSLM [4] introduces a dual-transformer architecture to model two-speaker dialogues, using one transformer to handle speech from each speaker. A cross-attention transformer layer is included to capture the interactions between the speakers' content. However, most approaches rely on a single language model architecture to manage both channels in a dialogue. Parrot [45] introduces the "next-token-pair prediction" approach, which uses a single decoder-only transformer to model both channels simultaneously by predicting tokens for both channels at each time step. Moshi [9] concatenates the user input channel and the model response channel and employs an RQ-Transformer that first processes data along the temporal dimension and then along the channel dimension. LSLM [51] utilizes a decoder-only Transformer focusing on modeling one speaker's speech in the dialogue. This model incorporates a streaming SSL encoder that continuously processes input from the listening channel and fuses its embeddings with those from the speaking channel.

Interactive Period Recognition (IPR) refers to the ability to recognize whether the users are interacting with it or not. SpeechLMs should provide response during the interactive period and remain silent during the non-interactive period. IPR is essential for creating a natural conversational flow, allowing the model to avoid unnecessary interruptions. It is crucial for situations where a small group of users is having a discussion, as the SpeechLM needs to discern when to join in and when to

TABLE V
EXAMPLES OF THE VARIOUS CAPABILITIES OF SPEECHLMS.

Task	Input	Output
Semantic-related Applications		
Spoken Dialogue	Speech(I am so angry right now!)	Speech(Sorry to hear that. What happened?)
Speech Translation	Speech(Good morning my friend!) Please translate the speech into French.	Speech(Bonjour mon ami !)
Automatic Speech Recognition	Speech(A river is on your left side.) Please perform automatic speech recognition to the speech	The transcribed speech is: A river is on your left side.
Keyword Spotting	Speech(How's your day? (Speaking to somebody else) Hey, Siri.) Please detect if the given speech contains any pre-defined keywords.	Keyword detected: "Hey, Siri."
Text-to-Speech Synthesis	Please synthesize the following text into speech: "A river is on your left side."	Sure. Here is the synthesized speech: Speech(A river is on your left side.)
Intent Classification	Speech(Can you help me book a flight?) Please detect the intent of the given speech.	Intent detected: "Book a flight".
Slot Filling	Speech(I want to fly from New York to San Francisco on June 5th.) Please fill the following slots based on the given speech: departure city, destination city, and date	Sure. Here are the filled slots. Departure city: New York. Destination city: San Francisco. Date: June 5th.
Query by Example Spoken Term Detection	Speech(Can you retrieve the audio documents that contain the word "happy"?) [A batch of audio documents]	Sure. I have found documents 2 and 4 contain the word happy.
Speaker-related Applications		
Speaker Identification	The following speech comes from Lucy: "[Lucy's speech]". The following speech comes from Bob: "Bob's speech". Question: Who is the speaker of the following speech [Unknown speech]	The speaker of the following speech: "[Unknown speech]" is Bob.
Speaker Verification	Here are two speeches. Speech 1: [Speech 1]. Speech 2: [Speech 2]. Do they come from the same speaker?	No, they do not come from the same speaker.
Speaker Diarization	[An audio file of the meeting] Please provide the speaker diarization output in the following format: [Start Time] - [End Time]: Speaker [Speaker].	Sure. Here is the speaker diarization result: 00:00 - 00:15: Alice, 00:16 - 00:40: Bob, 00:41 - 01:00: Carol, 01:01 - 01:20: Alice, 01:21 - 01:45: Bob and Carol (overlapping speech), 01:46 - 02:00: Carol.
Voice-Conditioned Speech Generation	Can you use a robotic voice to say "Hello world!" to me?	Sure. Speech([In robotic voice] Hello world!)
Paralinguistic Applications		
Emotion Recognition	Speech(What are you doing) What emotion does the speech convey?	It seems that the speaker is pretty shocked and angry about something.
Speech Separation	Speech([Two overlapping speeches]) Can you separate the two speeches?	Sure. Here is the speech separation result. Speech 1: Speech(Should I go for this one?). Speech 2: Speech(That one seems good.)
Paralinguistics-Enhanced Generation	Speech(Please generate a speech that sounds happy.)	Speech([In happy mood] My friend just gave me a candy!!)

stay silent. Additionally, it is important for the model to learn when to disregard instructions when users are not speaking at it. One approach to achieving IPR is through a Voice Activity Detection (VAD) module. MiniCPM-o 2.6 [122] integrates a VAD module to ensure the model responds only when the input audio surpasses a predefined VAD threshold. Inputs below this threshold are considered noise and ignored. VITA [55] takes a different approach by training the SpeechLM to distinguish between query speech and non-query audio. The model learns to output an end-of-sequence token to terminate its response when non-query audio is detected.

V. DOWNSTREAM APPLICATIONS

Unlike traditional speech systems like ASR and TTS, which usually focus on specific tasks, SpeechLMs function as generative foundation models. They can handle a diverse array of speech-only, text-only, and multi-modal tasks by following various instructions. In this section, we explore the primary downstream applications of SpeechLMs. The tasks discussed here primarily consist of traditional speech-related tasks, along with some that are unique to SpeechLMs. In contrast to TextLMs, which generate text containing only semantic information, SpeechLMs can model both semantic and paralinguistic information—such as pitch and timbre—making them more powerful models. We thereby categorize the downstream applications of SpeechLMs into three main classes: semantic-related applications, speaker-related applications, and paralinguistic applications. Table V provides an example for each downstream task.

A. Semantic-Related Applications

Semantic-related applications encompass key tasks that facilitate meaningful interactions between humans and machines. These applications require SpeechLMs to understand the semantic meaning of the input and generate responses that are not only contextually relevant but also logically coherent. The primary Semantic-related applications of SpeechLMs are as follows.

Spoken Dialogue. Spoken dialogue is the most natural application of SpeechLMs. Spoken dialogue systems are designed to facilitate natural conversations between humans and machines in spoken format. They can engage users in

interactive exchanges, understanding and generating responses based on the context of the conversation. Unlike TextLMs, SpeechLMs are able to perform conversations with humans directly in speech, which is a more natural way of communication. Note that SpeechLMs can not only perform speech-only dialogues but also perform cross-modal dialogues, such as taking texts as input and responding in speech format.

Speech Translation. Speech translation (ST) is the process of converting spoken language from one language to another. Similar to Spoken dialogue, SpeechLMs can perform ST in both single-modal and cross-modal settings. Specifically, the input and output of the ST task can be either in text or speech format.

Automated Speech Recognition. Automatic speech recognition (ASR) enables systems to convert spoken language into text. The input of ASR is a speech waveform, and the system outputs the transcription in textual form. For SpeechLMs, the input would be a combination of the speech waveform and the instruction to tell the model to perform ASR on the given speech.

Keyword Spotting. Keyword spotting can be considered a special type of ASR, where its primary objective is to identify specific words or phrases within continuous speech. While traditional ASR systems aim to transcribe entire spoken utterances into text, keyword spotting focuses specifically on identifying and extracting predefined keywords or phrases within continuous speech. The primary application of keyword spotting is to build voice-activated assistants in smart home devices. Those devices are activated when the specific keywords are triggered. Therefore, although SpeechLMs are capable of spotting and understanding more than just a couple of words, keyword spotting can be used to efficiently trigger SpeechLMs to respond to user inputs.

Text-to-Speech Synthesis. Text-to-speech synthesis (TTS) enables systems to synthesize written text into spoken language. In contrast to ASR, TTS takes text as input and outputs the converted speech waveform. Similarly, the input of the SpeechLMs would be a combination of the text to synthesize and the instruction, and the output is the synthesized speech.

Intent Classification. Intent classification is a critical task that identifies the underlying intention behind a user's input speech. The AI system can then perform certain actions based on the identified user intent (e.g., book a flight). Intent classification is particularly important in applications such as virtual assistants, customer service bots, and interactive voice response systems. To perform Intent Classification, it is more natural for SpeechLMs to take speech inputs and classify the results in text since it is easier to parse and classify the intent classification result in text than speech.

Slot Filling. Slot filling is an important task in spoken language understanding that involves identifying and extracting specific pieces of information from user inputs into predefined classes, such as intents, entities, and parameters that are essential for completing a task. For example, slot filling extracts the phrase "I want to fly from New York to San Francisco on June 5th." into distinct slots like "departure city" (New York), "destination city" (San Francisco), and "date" (June 5th). Similar to Intent Classification, it is more natural for

SpeechLMs to take speech inputs and extract the pieces in texts.

Query by Example Spoken Term Detection. Another spoken term detection task is query by example spoken term detection (QbE-STD), which allows users to identify specific spoken terms or phrases within a larger audio stream by providing an example of the desired term. Unlike traditional keyword spotting methods that rely on predefined lists of keywords, QbE-STD leverages the flexibility of example-based querying, enabling users to specify their search terms through audio samples.

B. Speaker-Related Applications

Speaker-related applications refer to the tasks that involve the processing of information related to speaker identity. It could involve classification tasks such as identifying, verifying, and distinguishing individual speakers based on their unique vocal characteristics, as well as generation tasks such as maintaining or modifying the timbre of a given speech. While we acknowledge that voice characteristics can be considered paralinguistic information, we believe that speaker-related applications are unique because they enable SpeechLMs to function in complex scenarios such as participating in multi-speaker conversations. In this section, we survey common speaker-related applications of SpeechLMs.

Speaker Identification. Speaker identification is the process of recognizing a person's identity based on their voice characteristics. It is a multi-class classification of a given speech as input. SpeechLMs can perform this task by taking an input speech and outputting the classification result in text or speech format. Moreover, SpeechLMs can also identify different speakers implicitly. Specifically, it can chat with multiple speakers at the same time, distinguishing the words from different speakers and responding to each speaker appropriately.

Speaker Verification. Speaker verification involves determining whether the speakers of a pair of speeches match with each other. Unlike speaker identification, which is a multi-class classification process, speaker verification is a binary classification process.

Speaker Diarization. Speaker diarization is the process of partitioning an audio stream into segments according to the identity of the speakers. It predicts "who is speaking when" for each timestamp [123]. A natural way to integrate speaker diarization into SpeechLMs is to have the model generate the transcript of each audio segment along with the identification of the speaker.

Voice-Conditioned Speech Generation. Voice-conditioned speech generation involves synthesizing speech based on the vocal characteristics of a specific speaker. This could involve voice cloning and voice conversion. Voice cloning utilizes a sample of the speaker's voice as a reference, enabling the model to reproduce the speaker's timbre when generating speech from input text. Voice conversion, on the other hand, modifies an existing speech signal to sound like it was produced by a different speaker while retaining the original content. Additionally, instead of giving the target vocal charac-

TABLE VI
A SUMMARY OF POPULAR BENCHMARKS FOR THE EVALUATION OF
SPEECHLMS. I/O, A, AND T REPRESENT INPUT/OUTPUT FORMAT, AUDIO,
AND TEXT, RESPECTIVELY.

Name	Eval Type	# Tasks	Audio Type	I/O
ABX [124]–[126]	Representation	1	Speech	$A \rightarrow -$
sWUGGY [126]	Linguistic	1	Speech	$A \rightarrow -$
sBLIMP [126]	Linguistic	1	Speech	$A \rightarrow -$
sStoryCloze [38]	Linguistic	1	Speech	$A/T \rightarrow -$
STSP [5]	Paralinguistic	1	Speech	$A/T \rightarrow A/T$
MMAU [127]	Downstream	27	Speech, Sound, Music	$A \rightarrow T$
Audiobench [128]	Downstream	8	Speech, Sound	$A \rightarrow T$
AIR-Bench [129]	Downstream	20	Speech, Sound, Music	$A \rightarrow T$
SD-Eval [130]	Downstream	4	Speech	$A \rightarrow T$
SUPERB [131]	Downstream	10	Speech	$A \rightarrow T$
Dynamic-SUPERB [131]	Downstream	180	Speech, Sound, Music	$A \rightarrow T$
SALMON [132]	Downstream	8	Speech	$A \rightarrow -$
VoiceBench [133]	Downstream	8	Speech	$A \rightarrow A$
VoxEval [134]	Downstream	56	Speech	$A \rightarrow A$

teristics, SpeechLMs should also be able to adapt their output timbre based on various speech or text instructions.

C. Paralinguistic Applications

Paralinguistics refers to the non-verbal elements of communication that accompany spoken language. It encompasses various vocal attributes that convey meaning beyond the actual words spoken. These elements can significantly influence how messages are interpreted and understood. The key elements of paralinguistics include pitch, timbre, column, rate of speech, pauses, etc. Since combining elements in paralinguistics in different ways can result in a speech with different emotions, we include emotion-related tasks as paralinguistic applications as well.

Emotion Recognition. Emotion recognition task involves identifying and classifying the emotion carried by a given speech into predefined classes. Similar to speaker identification, SpeechLMs are capable of not only directly performing this task but also implicitly recognizing users' emotions through their speech queries and responding accordingly.

Speech Separation. Speech separation refers to the process of isolating individual speech signals from a mixture of sounds, such as when multiple speakers are talking simultaneously. When separating the input speech, SpeechLMs can not only output the contents of each person in speech but also in text format (i.e., transcriptions).

Paralinguistics-Enhanced Generation. Paralinguistics-enhanced generation refers to the process of instructing SpeechLMs to produce speech that exhibits specific paralinguistic characteristics. Users can define these characteristics in their prompts, allowing the model to generate speech that aligns with their specifications. Examples of paralinguistics-enhanced generation include synthesizing speech with a specific style, speaking at a fast pace, and even singing. This capability distinguishes SpeechLMs from TextLMs and facilitates a more engaging and interactive form of communication with the AI models.

VI. EVALUATIONS

Similar to TextLMs, SpeechLMs have a wide range of capabilities, making it challenging to compare different SpeechLMs. Consequently, it's essential to evaluate

SpeechLMs from various perspectives to determine their effectiveness. In this section, we review the commonly used methods and benchmarks (Table VI) for evaluating SpeechLMs. We categorize these evaluation methods into automatic and human assessments, each containing distinct evaluation aspects.

A. Automatic (Objective) Evaluation

Automatic evaluation methods are essential for providing quick and consistent assessments of SpeechLMs. These methods typically rely on quantitative metrics that can be computed without human intervention. Below, we outline some of the most commonly used automatic evaluation techniques.

Representation Evaluation. Representation (embedding) is a crucial component in SpeechLMs (and TextLMs). It refers to how input data, such as speech or text, is transformed into a format that the model can understand and process. Effective representation lays a solid foundation for models to understand lexical, syntax, and contextual information, which are vital for generating coherent and contextually relevant outputs.

In the context of SpeechLMs, representation evaluation focuses on how well the model encodes speech features into meaningful vectors. GSLM [37] uses *between-speaker ABX score* to measure the embedding similarity. It quantifies how well-separated the phonetic categories are. Specifically, It works by comparing three sound samples: two from the same category (A) and one from a different category (B). The test measures how often the system correctly identifies that two sounds from category A are more similar to each other than one sound from A is to a sound from B. Another way of evaluating representations is through speech resynthesis [37]. Specifically, an input speech is encoded into tokens and then synthesized back to speech. Then, word error rate (WER) or character error rate (CER) can be computed on the ASR results of the input and resynthesized speech. This measures the information loss caused by discretizing the input speech into speech tokens, thereby evaluating the robustness of the latent representations.

Linguistic Evaluation. Linguistics, including lexical, syntactic, and semantic evaluation methods, assess the model's ability to generate and understand the rules for constructing words, sentences, and meaningful contents. These evaluations focus on the correctness and appropriateness of word choices, the grammatical structure of the outputs, and the coherence and relevance of the generated content. In terms of benchmark datasets, sWUGGY [126] assesses at the lexical level by determining if the model can distinguish a real word from a (real, non-real) word pair. sBLIMP [126] evaluates at the syntactic level by determining if the model can identify the grammatically correct sentence from a (grammatical, ungrammatical) sentence pair. Spoken StoryCloze [38] evaluates semantic comprehension by assessing the model's capability to select the genuine ending of a story from a pair of ending choices. All the evaluation is conducted by comparing the model's negative log-likelihood of the data pair.

Paralinguistic Evaluation. In contrast to linguistic evaluation, paralinguistic evaluation focuses on the non-verbal aspects of communication that accompany speech. Some works

choose to utilize paralinguistic tokens alongside semantic tokens to enhance the paralinguistic abilities of SpeechLMs [5], [42], so one way is to evaluate the paralinguistic tokens. pGSLM [42] measures the correctness, consistency, and expressiveness of the prosodic tokens. Correctness evaluates the model's ability to generate accurate prosodic profiles by calculating the minimal mean absolute error (min-MAE) of the prosodic tokens from 20 generated samples against the prosodic tokens from the reference, consistency is assessed through the Pearson correlation between the mean values of the prompt prosodic and its generated continuation prosodic tokens, and expressiveness is measured by the standard deviation of the generated prosody token values, with the expectation that it matches the variability of the ground truth. We note that the same metrics can also be applied to other paralinguistic tokens. Instead of evaluating from the token level, SPIRIT-LM [5] proposes to measure on the perceptual level. They introduced a speech-text sentiment preservation benchmark (STSP), which requires the model to generate a text or speech sequence of tokens that preserves the sentiment of the prompt. A sentiment classifier is used to assess the sentiment in the generated speech. It should be noted that although they only apply the preservation approach on sentiment, this idea can be generalized to other paralinguistic features, such as timbre or prosody.

Generation Quality and Diversity. Quality and diversity are two crucial aspects of model generation. Typically, there is a trade-off between these dimensions when sampling model responses at different temperatures, so GSLM [37] suggests using the Area Under the Curve (AUC) with various temperature values. Specifically, AUC on perplexity and VERT are employed to assess these factors, where VERT represents the geometric mean of the ratio of k-grams in the generated speech that appears at least once. Additionally, the ChatGPT score can be utilized to evaluate the quality of the generated speech. In this process, the generated speech is transcribed using state-of-the-art ASR models and then sent to ChatGPT for quality (and diversity) assessment.

Real-Time Interaction Evaluation. Real-time interaction evaluation involves assessing the ability of SpeechLMs to interact in real-time, which is crucial for those models that facilitate full-duplex communication. Current research focuses on evaluating the naturalness and usefulness of speech generated in real-time. dGSLM [4] examines the naturalness of dialogues between two speakers by introducing different turn-taking events, such as speech segments (Inter-Pausal Unit, IPU), pauses within speech (pause), pauses between speeches (gaps), and overlapping speech (overlap). The generated speech is considered more natural if the statistics of these turn-taking events closely resemble those found in human dialogues. Another method involves speech continuation, where the generated speech is considered more natural if the turn-taking statistics of the speech prompts align closely with those of the subsequent continuations. Additionally, evaluating SpeechLM's usefulness as an AI assistant in real-time interaction is crucial. The Parrot model [45] suggests using reflective pauses and interruptions to assess usefulness, where reflective pause evaluates the SpeechLM's ability to remain silent while the user speaks, and interruption measures the SpeechLM's capability to cease speaking when interrupted.

Downstream Evaluation. Downstream evaluation refers to evaluating the ability of SpeechLMs to perform specific tasks, such as ASR, TTS, Speaker Identification, etc. The evaluation can be performed on pre-trained models by adding few-shot example(s) at the start of the prompt or on the instruction-tuned models by directly instructing them to do so. Several benchmarks compile a broad array of downstream tasks to provide a comprehensive assessment of SpeechLMs. SUPERB [123] includes a variety of speech understanding tasks. SD-Eval [130] assesses the paralinguistic understanding of SpeechLMs using the emotion, age, environment, and age classification tasks. SALMON [132] tests the SpeechLMs' ability to generate speech with consistent paralinguistic and environmental characteristics. Dynamic-SUPERB [131], MMAU [127], AirBench [129], and AudioBench [128] extend beyond traditional speech tasks to include sound and/or music-related challenges as well.

While these benchmarks offer comprehensive coverage of audio-related tasks, they primarily require models to respond in text, which creates a barrier to end-to-end speech interaction evaluation. To address this limitation, newer benchmarks have emerged that support speech output. VoxEval [134] focuses on benchmarking the knowledge understanding capabilities of SpeechLMs, providing question-answer pairs in comprehensive subjects in speech format, along with an evaluation pipeline tailored for speech output. Additionally, it presents questions under various input audio conditions for evaluating the model's robustness. Similarly, VoiceBench [133] supports speech output while focusing on assessing general knowledge, instruction following, and response safety of SpeechLMs.

B. Human (Subjective) Evaluation.

Human evaluation plays a crucial role in assessing the performance of SpeechLMs, as ultimately, speech is designed to be heard and perceived by humans. This type of evaluation relies on human judgment to assess the quality of the outputs generated by SpeechLMs. Below, we outline several commonly used human evaluation methods.

Mean Opinion Score. Mean opinion score (MOS) is a widely used metric in the field of speech evaluation that quantifies the perceived quality of speech output as judged by human listeners. Typically, a group of evaluators listens to a series of audio samples generated by the SpeechLM and rates each sample on a predefined scale, often from 1 (poor quality) to 5 (excellent quality).

MOS is calculated by averaging the scores given by all evaluators for each audio sample, providing a single score that reflects the overall quality as perceived by humans. Variations of MOS focus on different aspects of speech quality, including MMOS, PMOS, and SMOS [42], [46]. They evaluate the aspects of naturalness, prosody, and timbre similarity of the given speech, respectively.

Typically, evaluating naturalness or timbre similarity involves collecting human opinions. However, this process can be complicated due to the challenges of recruiting participants

and gathering their evaluations. As a result, researchers often turn to machine-based evaluations. They commonly employ neural network models specifically trained for these tasks. For instance, a naturalness prediction model [135] can assess the naturalness of generated outputs, while a speaker identification model can evaluate timbre similarity.

VII. CHALLENGES AND FUTURE DIRECTIONS

While SpeechLMs have demonstrated impressive abilities, the research in this area is still under explored. In this section, we survey challenges, unsolved questions, and possible directions for future research in the study of SpeechLMs.

A. Understanding Different Component Choices

Current research on SpeechLMs encompasses key components such as speech tokenizers, language models, and vocoders, each offering a diverse range of options. While some studies have compared various component choices—primarily focusing on speech tokenizers—the comparisons tend to be limited in scope and depth [37], [39]. Consequently, there remains a significant gap in understanding the advantages and disadvantages of different component selections. Therefore, studies aimed at comprehensively comparing these choices are essential. Such an investigation would yield valuable insights and serve as a guide for selecting more efficient components when developing SpeechLMs.

B. End-to-End Training

Although SpeechLMs can generate speech directly without relying on text signals, some studies train the three components separately. This separate optimization may hinder the model's overall potential. Consequently, it would be worthwhile to investigate whether training can be conducted in an end-to-end manner, allowing gradients to be back-propagated from the vocoder's output to the tokenizer's input. By exploring this fully end-to-end approach, we could potentially enable SpeechLMs to produce more coherent, contextually relevant, and high-fidelity speech outputs.

C. Real-Time Speech Generation

Enabling real-time speech generation is crucial in SpeechLM as it fosters a more interactive way of engaging with humans. However, the most adopted approaches described in section III still result in noticeable delays between input and output speech generation. This delay occurs because a typical vocoder must wait for the entire sequence of output tokens to be generated by the language model before functioning, making it the most time-consuming process in the inference pipeline. One potential solution to improve latency is to develop a streamable pipeline, allowing the speech input and output to be processed and generated in chunks. Another option could involve the SpeechLM autonomously generating audio samples in waveform. Overall, this area of real-time speech generation remains underexplored and requires further investigation.

D. Safety Risks in SpeechLMs

Safety is a highly significant subject in the field of Machine Learning, particularly when it comes to large-scale generative AI models. While there has been extensive research on safety concerns in TextLMs, the safety issues in SpeechLMs have not been thoroughly investigated. The safety challenges in SpeechLMs present both similarities and unique aspects compared to TextLMs, as highlighted in OpenAI's recent report on the safety issues of GPT-4o's voice model [136]. Therefore, it is crucial for future research to explore safety vulnerabilities in SpeechLMs and develop safer SpeechLMs.

Primary concerns for the safety issues in SpeechLMs include but are not limited to *toxicity* and *privacy*. *Toxicity* refers to the harmful nature of the content generated by SpeechLMs. For instance, these models might produce semantically dangerous content, such as instructions for making explosives. Additionally, they could generate acoustically inappropriate content, like erotic speech [136], which presents a unique challenge. *Privacy* involves the risk of revealing personal information from the speech input after it has been processed by a SpeechLM. For example, the model might infer the speaker's identity based on the semantic content or acoustic features of the input. Even more concerning is the potential for the model to make biased inferences about the speaker, such as their ethnicity or religious beliefs, based on insufficient (e.g., acoustic) information [136].

E. Performance on Rare Languages

SpeechLMs directly model speech data, which allows them to more effectively handle "low-resource" languages compared to TextLMs. "Low-resource" languages are those that lack extensive textual data, making it challenging for TextLMs to model them efficiently. In contrast, SpeechLM provides a better solution by modeling the speech data of these "low-resource" languages, which often have more available audio data than text [37]. Therefore, future research could focus on training SpeechLMs in "low-resource" languages or dialects to expand their capabilities.

VIII. CONCLUSIONS

This survey provides a comprehensive overview of recent advancements in Speech Language Models (SpeechLMs). We begin by addressing the limitations of the naive framework that combines Automatic Speech Recognition (ASR), Large Language Models (LLMs), and Text-to-Speech (TTS) systems for voice interactions. Next, we highlight the key advantages offered by SpeechLMs. Following this, we explore the architectures of SpeechLMs, detailing the components involved and their training recipes. We also discuss their capabilities in various downstream applications as well as their different evaluation methods. Finally, we identify the major challenges in developing SpeechLMs and outline potential directions for future research. We hope this survey will illuminate the field and assist the research community in creating more powerful Speech Language Models.

REFERENCES

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[3] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[4] T. A. Nguyen, E. Kharitonov, J. Copet, Y. Adi, W.-N. Hsu, A. Elkahky, P. Tomasello, R. Algayres, B. Sagot, A. Mohamed *et al.*, "Generative spoken dialogue language modeling," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 250–266, 2023.

[5] T. A. Nguyen, B. Muller, B. Yu, M. R. Costa-Jussa, M. Elbayad, S. Popuri, P.-A. Duquenne, R. Algayres, R. Mavlyutov, I. Gat *et al.*, "Spirit-lm: Interleaved spoken and written language model," *arXiv preprint arXiv:2402.05755*, 2024.

[6] R. Huang, M. Li, D. Yang, J. Shi, X. Chang, Z. Ye, Y. Wu, Z. Hong, J. Huang, J. Liu *et al.*, "Audiogpt: Understanding and generating speech, music, sound, and talking head," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 802–23 804.

[7] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[8] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, and X. Qiu, "SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 15 757–15 773. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.1055

[9] A. Défossez, L. Mazaré, M. Orsini, A. Royer, P. Pérez, H. Jégou, E. Grave, and N. Zeghidour, "Moshi: a speech-text foundation model for real-time dialogue," Kyutai, Tech. Rep., September 2024. [Online]. Available: http://kyutai.org/Moshi.pdf

[10] Z. Xie and C. Wu, "Mini-Omni: Language Models Can Hear, Talk While Thinking in Streaming," *arXiv preprint arXiv:2408.16725*, 2024.

[11] Q. Fang, S. Guo, Y. Zhou, Z. Ma, S. Zhang, and Y. Feng, "Llama-omni: Seamless speech interaction with large language models," *arXiv preprint arXiv:2409.06666*, 2024.

[12] 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.

[13] H. Le, J. Pino, C. Wang, J. Gu, D. Schwab, and L. Besacier, "Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 3520–3533. [Online]. Available: https://aclanthology.org/2020.coling-main.314

[14] Y. Fathullah, C. Wu, E. Lakomkin, K. Li, J. Jia, Y. Shangguan, J. Mahadeokar, O. Kalinli, C. Fuegen, and M. Seltzer, "Audiochatllama: Towards general-purpose speech abilities for llms," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 5522–5532.

[15] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, "SALMONN: Towards Generic Hearing Abilities for Large Language Models," in *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: https://openreview.net/forum?id=14rn7HpKVk

[16] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[17] 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.

[18] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. [Online]. Available: https://openreview.net/forum?id=rylwJxrYDS

[19] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 244–250.

[20] 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.

[21] X. Zhang, D. Zhang, S. Li, Y. Zhou, and X. Qiu, "Speechtokenizer: Unified speech tokenizer for speech large language models," in *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: https://openreview.net/forum?id=AF9Q8Vip84

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

[23] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang *et al.*, "Google usm: Scaling automatic speech recognition beyond 100 languages," *arXiv preprint arXiv:2303.01037*, 2023.

[24] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[26] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[27] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[28] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[29] T. GLM, A. Zeng, B. Xu, B. Wang, C. Zhang, D. Yin, D. Zhang, D. Rojas, G. Feng, H. Zhao *et al.*, "Chatglm: A family of large language models from glm-130b to glm-4 all tools," *arXiv preprint arXiv:2406.12793*, 2024.

[30] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mixtral of Experts," Jan. 2024.

[31] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, 2016, p. 125.

[32] 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.

[33] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[34] 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.

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

[36] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *Transactions on Machine Learning Research*, 2023.

[37] 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.

[38] 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," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[39] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. d. C. Quitry, P. Chen, D. E. Badawy, W. Han, E. Kharitonov *et al.*, "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv:2306.12925*, 2023.

[40] Q. Zhang, L. Cheng, C. Deng, Q. Chen, W. Wang, S. Zheng, J. Liu, H. Yu, and C. Tan, "Omniflatten: An end-to-end gpt model for seamless voice conversation," *arXiv preprint arXiv:2410.17799*, 2024.

[41] W. Chen, Z. Ma, R. Yan, Y. Liang, X. Li, R. Xu, Z. Niu, Y. Zhu, Y. Yang, Z. Liu *et al.*, "Slam-omni: Timbre-controllable voice interaction system with single-stage training," *arXiv preprint arXiv:2412.15649*, 2024.

[42] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T. A. Nguyen, M. Riviere, A. Mohamed, E. Dupoux, and W.-N. Hsu, "Text-free prosody-aware generative spoken language modeling," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds.   Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8666–8681. [Online]. Available: https://aclanthology.org/2022.acl-long.593

[43] T. Wang, L. Zhou, Z. Zhang, Y. Wu, S. Liu, Y. Gaur, Z. Chen, J. Li, and F. Wei, "Viola: Conditional language models for speech recognition, synthesis, and translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[44] J. Li, D. Wang, X. Wang, Y. Qian, L. Zhou, S. Liu, M. Yousefi, C. Li, C.-H. Tsai, Z. Xiao *et al.*, "Investigating neural audio codecs for speech language model-based speech generation," *arXiv preprint arXiv:2409.04016*, 2024.

[45] Z. Meng, Q. Wang, W. Cui, Y. Zhang, B. Wu, I. King, L. Chen, and P. Zhao, "Parrot: Autoregressive spoken dialogue language modeling with decoder-only transformers," in *Audio Imagination: NeurIPS 2024 Workshop AI-Driven Speech, Music, and Sound Generation*, 2024.

[46] D. Zhang, X. Zhang, J. Zhan, S. Li, Y. Zhou, and X. Qiu, "Speechgpt-gen: Scaling chain-of-information speech generation," *arXiv preprint arXiv:2401.13527*, 2024.

[47] E. Nachmani, A. Levkovitch, R. Hirsch, J. Salazar, C. Asawaroengchai, S. Mariooryad, E. Rivlin, R. Skerry-Ryan, and M. T. Ramanovich, "Spoken question answering and speech continuation using spectrogram-powered llm," in *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: https://openreview.net/forum?id=izrOLJov5y

[48] R. Algayres, Y. Adi, T. Nguyen, J. Copet, G. Synnaeve, B. Sagot, and E. Dupoux, "Generative spoken language model based on continuous word-sized audio tokens," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds.   Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3008–3028. [Online]. Available: https://aclanthology.org/2023.emnlp-main.182

[49] Z. Du, J. Wang, Q. Chen, Y. Chu, Z. Gao, Z. Li, K. Hu, X. Zhou, J. Xu, Z. Ma, W. Wang, S. Zheng, C. Zhou, Z. Yan, and S. Zhang, "LauraGPT: Listen, Attend, Understand, and Regenerate Audio with GPT," https://arxiv.org/abs/2310.04673v4, Oct. 2023.

[50] J.-C. Chou, C.-M. Chien, W.-N. Hsu, K. Livescu, A. Babu, A. Conneau, A. Baevski, and M. Auli, "Toward joint language modeling for speech units and text," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds.   Singapore: Association for Computational Linguistics, Dec. 2023, pp. 6582–6593. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.438

[51] Z. Ma, Y. Song, C. Du, J. Cong, Z. Chen, Y. Wang, Y. Wang, and X. Chen, "Language model can listen while speaking," *arXiv preprint arXiv:2408.02622*, 2024.

[52] Z. Xie and C. Wu, "Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities," *arXiv preprint arXiv:2410.11190*, 2024.

[53] X. Wang, Y. Li, C. Fu, L. Xie, K. Li, X. Sun, and L. Ma, "Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm," *arXiv preprint arXiv:2411.00774*, 2024.

[54] J. Pan, J. Wu, Y. Gaur, S. Sivasankaran, Z. Chen, S. Liu, and J. Li, "Cosmic: Data efficient instruction-tuning for speech in-context learning," *arXiv preprint arXiv:2311.02248*, 2023.

[55] C. Fu, H. Lin, Z. Long, Y. Shen, M. Zhao, Y. Zhang, X. Wang, D. Yin, L. Ma, X. Zheng *et al.*, "Vita: Towards open-source interactive omni multimodal llm," *arXiv preprint arXiv:2408.05211*, 2024.

[56] W. Yu, S. Wang, X. Yang, X. Chen, X. Tian, J. Zhang, G. Sun, L. Lu, Y. Wang, and C. Zhang, "Salmonn-omni: A codec-free llm for full-duplex speech understanding and generation," *arXiv preprint arXiv:2411.18138*, 2024.

[57] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds.   Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[59] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, P. J. Moreno, A. Bapna, and H. Zen, "Maestro: Matched speech text representations through modality matching," in *Proceedings of Interspeech*, 2022, pp. 4093–4097.

[60] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[61] Z. Du, Q. Chen, S. Zhang, K. Hu, H. Lu, Y. Yang, H. Hu, S. Zheng, Y. Gu, Z. Ma *et al.*, "Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens," *arXiv preprint arXiv:2407.05407*, 2024.

[62] B. Veluri, B. N. Peloquin, B. Yu, H. Gong, and S. Gollakota, "Beyond turn-based interfaces: Synchronous LLMs as full-duplex dialogue agents," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds.   Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 21 390–21 402. [Online]. Available: https://aclanthology.org/2024.emnlp-main.1192/

[63] K. Chen, Y. Gou, R. Huang, Z. Liu, D. Tan, J. Xu, C. Wang, Y. Zhu, Y. Zeng, K. Yang *et al.*, "Emova: Empowering language models to see, hear and speak with vivid emotions," *arXiv preprint arXiv:2409.18042*, 2024.

[64] W. Huang, Z. Zhang, Y. T. Yeung, X. Jiang, and Q. Liu, "Spiral: Self-supervised perturbation-invariant representation learning for speech pre-training," *arXiv preprint arXiv:2201.10207*, 2022.

[65] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*.   PMLR, 2021, pp. 5530–5540.

[66] Y. Ren, T. Wang, J. Yi, L. Xu, J. Tao, C. Y. Zhang, and J. Zhou, "Fewer-token neural speech codec with time-invariant codes," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.   IEEE, 2024, pp. 12 737–12 741.

[67] X. Zhang, X. Lyu, Z. Du, Q. Chen, D. Zhang, H. Hu, C. Tan, T. Zhao, Y. Wang, B. Zhang *et al.*, "Intrinsicvoice: Empowering llms with intrinsic real-time voice interaction abilities," *arXiv preprint arXiv:2410.08035*, 2024.

[68] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=tEYskw1VY2

[69] P. Peng, P.-Y. Huang, S.-W. Li, A. Mohamed, and D. Harwath, "VoiceCraft: Zero-shot speech editing and text-to-speech in the wild," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds.   Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 12 442–12 462. [Online]. Available: https://aclanthology.org/2024.acl-long.673/

[70] A. Zeng, Z. Du, M. Liu, L. Zhang, S. Jiang, Y. Dong, and J. Tang, "Scaling speech-text pre-training with synthetic interleaved data," *arXiv preprint arXiv:2411.17607*, 2024.

[71] Y. Zhu, D. Su, L. He, L. Xu, and D. Yu, "Generative pre-trained speech language model with efficient hierarchical transformer," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds.   Bangkok, Thailand: Association

for Computational Linguistics, Aug. 2024, pp. 1764–1775. [Online]. Available: https://aclanthology.org/2024.acl-long.97

[72] C. Du, Y. Guo, F. Shen, Z. Liu, Z. Liang, X. Chen, S. Wang, H. Zhang, and K. Yu, "Unicats: A unified context-aware text-to-speech framework with contextual vq-diffusion and vocoding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 924–17 932.

[73] S. Maiti, Y. Peng, S. Choi, J.-w. Jung, X. Chang, and S. Watanabe, "Voxtlm: Unified decoder-only models for consolidating speech recognition, synthesis and speech, text continuation tasks," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 13 326–13 330.

[74] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, and J. Mahadeokar, "Voicebox: Text-guided multilingual universal speech generation at scale," *Advances in neural information processing systems*, vol. 36, 2024.

[75] T. SpeechTeam, "FunAudioLLM: Voice Understanding and Generation Foundation Models for Natural Interaction Between Humans and LLMs," *arXiv preprint arXiv:2407.04051*, 2024.

[76] Z. Gao, S. Zhang, M. Lei, and I. McLoughlin, "San-m: Memory equipped self-attention for end-to-end speech recognition," *arXiv preprint arXiv:2006.01713*, 2020.

[77] Y. A. Li, C. Han, X. Jiang, and N. Mesgarani, "Hiftnet: A fast high-quality neural vocoder with harmonic-plus-noise filter and inverse short time fourier transform," *arXiv preprint arXiv:2309.09493*, 2023.

[78] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[79] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[80] Y. Koizumi, K. Yatabe, H. Zen, and M. Bacchiani, "Wavefit: An iterative and non-autoregressive neural vocoder based on fixed-point iteration," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 884–891.

[81] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 31, pp. 2523–2533, 2023.

[82] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, X. Chang, J. Shi, S. Zhao, J. Bian, and X. Wu, "Uniaudio: An audio foundation model toward universal audio generation," *arXiv preprint arXiv:2310.00704*, 2023.

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

[84] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved RVQGAN," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=qjnl1QUnFA

[85] R. Algayres, A. Nabli, B. Sagot, and E. Dupoux, "Speech sequence embeddings using nearest neighbors contrastive learning," in *Interspeech 2022*, 2022, pp. 2123–2127.

[86] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[87] P. Anastassiou, J. Chen, J. Chen, Y. Chen, Z. Chen, Z. Chen, J. Cong, L. Deng, C. Ding, L. Gao *et al.*, "Seed-tts: A family of high-quality versatile speech generation models," *arXiv preprint arXiv:2406.02430*, 2024.

[88] J. Betker, "Better speech synthesis through scaling," *arXiv preprint arXiv:2305.07243*, 2023.

[89] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[90] S.-g. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "Bigvgan: A universal neural vocoder with large-scale training," in *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023. [Online]. Available: https://openreview.net/forum?id=iTtGCMDEzS_

[91] J.-H. Kim, S.-H. Lee, J.-H. Lee, and S.-W. Lee, "Fre-gan: Adversarial frequency-consistent audio synthesis," in *Proceedings of Interspeech*, 2021, pp. 2197–2201.

[92] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[93] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[94] W.-C. Huang, Y.-C. Wu, H.-T. Hwang, P. L. Tobing, T. Hayashi, K. Kobayashi, T. Toda, Y. Tsao, and H.-M. Wang, "Refined wavenet vocoder for variational autoencoder based voice conversion," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[95] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=a-xFK8Ymz5J

[96] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=NsMLjcFaO8O

[97] S.-g. Lee, H. Kim, C. Shin, X. Tan, C. Liu, Q. Meng, T. Qin, W. Chen, S. Yoon, and T.-Y. Liu, "Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior," in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://openreview.net/forum?id=_BNiN4IjC5

[98] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[99] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," in *Interspeech 2020*, 2020, pp. 2757–2761.

[100] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen *et al.*, "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7669–7673.

[101] D. Galvez, G. Diamos, J. Ciro, J. F. Cerón, K. Achorn, A. Gopi, D. Kanter, M. Lam, M. Mazumder, and V. J. Reddi, "The people's speech: A large-scale diverse english speech recognition dataset for commercial usage," in *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[102] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," 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)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 993–1003. [Online]. Available: https://aclanthology.org/2021.acl-long.80

[103] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 6. International Speech Communication Association, 2021, pp. 4376–4380.

[104] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, May 2020, pp. 4218–4222. [Online]. Available: https://aclanthology.org/2020.lrec-1.520

[105] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *2013 international conference oriental COCOSDA held jointly with 2013 conference on Asian spoken language research and evaluation (O-COCOSDA/CASLRE)*. IEEE, 2013, pp. 1–4.

[106] B. Zhang, H. Lv, P. Guo, Q. Shao, C. Yang, L. Xie, X. Xu, H. Bu, X. Chen, C. Zeng *et al.*, "Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6182–6186.

[107] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," in *Proc. Interspeech 2019*, 2019, pp. 1526–1530. [Online]. Available: http://www.openslr.org/60/

[108] C. Wang, A. Wu, and J. Pino, "Covost 2 and massively multilingual speech-to-text translation," in *Proc. Interspeech 2021*, 2021, pp. 2247–2251.

[109] Y. Jia, M. Tadmor Ramanovich, Q. Wang, and H. Zen, "CVSS corpus and massively multilingual speech-to-speech translation," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 6691–6703. [Online]. Available: https://aclanthology.org/2022.lrec-1.720

[110] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[111] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech 2018*, 2018, pp. 1086–1090.

[112] A. Clifton, A. Pappu, S. Reddy, Y. Yu, J. Karlgren, B. Carterette, and R. Jones, "The spotify podcast dataset," *arXiv preprint arXiv:2004.04270*, 2020.

[113] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: A resource for the next generations of speech-to-text." in *LREC*, vol. 4, 2004, pp. 69–71.

[114] T. A. Nguyen, W.-N. Hsu, A. d'Avirro, B. Shi, I. Gat, M. Fazel-Zarani, T. Remez, J. Copet, G. Synnaeve, M. Hassid *et al.*, "Expresso: A benchmark and analysis of discrete expressive speech resynthesis," in *Proceedings of INTERSPEECH 2023*, 2023, pp. 4823–4827.

[115] P.-A. Duquenne, K. Heffernan, A. Mourachko, B. Sagot, and H. Schwenk, "Sonar expressive: Zero-shot expressive speech-to-speech translation," *Meta AI Research*, 2023.

[116] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, and S. Gehrmann, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[117] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[118] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[119] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.

[120] G.-T. Lin, P. G. Shivakumar, A. Gourav, Y. Gu, A. Gandhe, H.-y. Lee, and I. Bulyko, "Align-slm: Textless spoken language models with reinforcement learning from ai feedback," *arXiv preprint arXiv:2411.01834*, 2024.

[121] D. Zhang, Z. Li, S. Li, X. Zhang, P. Wang, Y. Zhou, and X. Qiu, "Speechalign: Aligning speech generation to human preferences," in *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, December 2024, poster presentation on December 11, 2024, from 11 a.m. to 2 p.m. PST.

[122] OpenBMB, "Minicpm-o 2.6: A gpt-4.0 level mllm for vision, speech, and multimodal live streaming on your phone," https://huggingface.co/openbmb/MiniCPM-o-2_6-int4, 2024.

[123] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," in *Proc. Interspeech*, 2021, pp. 1194–1198.

[124] M. Versteegh, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015: Proposed approaches and results," *Procedia Computer Science*, vol. 81, pp. 67–72, 2016, sLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091630045X

[125] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X.-N. Cao, L. Miskic, C. Dugrain, L. Ondel, A. W. Black, L. Besacier, S. Sakti, and E. Dupoux, "The zero resource speech challenge 2019: Tts without t," in *Interspeech 2019*, 2019, pp. 1088–1092.

[126] T. A. Nguyen, M. de Seyssel, P. Rozé, M. Rivière, E. Kharitonov, A. Baevski, E. Dunbar, and E. Dupoux, "The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling," in *Proceedings of the Workshop on Self-Supervised Learning for Speech and Audio Processing*. NeurIPS, 2020.

[127] S. Sakshi, U. Tyagi, S. Kumar, A. Seth, R. Selvakumar, O. Nieto, R. Duraiswami, S. Ghosh, and D. Manocha, "Mmau: A massive multi-task audio understanding and reasoning benchmark," *arXiv preprint arXiv:2410.19168*, 2024.

[128] B. Wang, X. Zou, G. Lin, S. Sun, Z. Liu, W. Zhang, Z. Liu, A. Aw, and N. F. Chen, "Audiobench: A universal benchmark for audio large language models," *arXiv preprint arXiv:2406.16020*, 2024.

[129] Q. Yang, J. Xu, W. Liu, Y. Chu, Z. Jiang, X. Zhou, Y. Leng, Y. Lv, Z. Zhao, C. Zhou *et al.*, "Air-bench: Benchmarking large audio-language models via generative comprehension," *arXiv preprint arXiv:2402.07729*, 2024.

[130] J. Ao, Y. Wang, X. Tian, D. Chen, J. Zhang, L. Lu, Y. Wang, H. Li, and Z. Wu, "Sd-eval: A benchmark dataset for spoken dialogue understanding beyond words," *arXiv preprint arXiv:2406.13340*, 2024.

[131] C.-y. Huang, W.-C. Chen, S.-w. Yang, A. T. Liu, C.-A. Li, Y.-X. Lin, W.-C. Tseng, A. Diwan, Y.-J. Shih, J. Shi *et al.*, "Dynamic-superb phase-2: A collaboratively expanding benchmark for measuring the capabilities of spoken language models with 180 tasks," *arXiv preprint arXiv:2411.05361*, 2024.

[132] G. Maimon, A. Roth, and Y. Adi, "A suite for acoustic language model evaluation," *arXiv preprint arXiv:2409.07437*, 2024.

[133] Y. Chen, X. Yue, C. Zhang, X. Gao, R. T. Tan, and H. Li, "Voicebench: Benchmarking llm-based voice assistants," *CoRR*, vol. abs/2410.17196, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2410.17196

[134] W. Cui, X. Jiao, Z. Meng, and I. King, "Voxeval: Benchmarking the knowledge understanding capabilities of end-to-end spoken language models," *arXiv preprint arXiv:2501.04962*, 2025.

[135] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," in *Proceedings of Interspeech 2021*, 2021, pp. 2127–2131.

[136] OpenAI, "Gpt-4o system card," 2024, online; Accessed on 6-September-2024. [Online]. Available: https://openai.com/index/gpt-4o-system-card/