Baichuan-Audio: A Unified Framework for End-to-End Speech Interaction

Tianpeng Li Jun Liu Tao Zhang Yuanbo Fang Da Pan Mingrui Wang
Zheng Liang Zehuan Li Mingan Lin Guosheng Dong Jianhua Xu
Haoze Sun* Zenan Zhou* Weipeng Chen

Baichuan Inc.
{sunhaoze, zhouzenan}@baichuan-inc.com

Abstract

We introduce Baichuan-Audio, an end-to-end audio large language model that seamlessly integrates audio understanding and generation. It features a text-guided aligned speech generation mechanism, enabling real-time speech interaction with both comprehension and generation capabilities. Baichuan-Audio leverages a pre-trained ASR model, followed by multi-codebook discretization of speech at a frame rate of 12.5 Hz. This multi-codebook setup ensures that speech tokens retain both semantic and acoustic information. To further enhance modeling, an independent audio head is employed to process audio tokens, effectively capturing their unique characteristics. To mitigate the loss of intelligence during pre-training and preserve the original capabilities of the LLM, we propose a two-stage pre-training strategy that maintains language understanding while enhancing audio modeling. Following alignment, the model excels in real-time speech-based conversation and exhibits outstanding question-answering capabilities, demonstrating its versatility and efficiency. The proposed model demonstrates superior performance in real-time spoken dialogue and exhibits strong question-answering abilities. Our code, model and training data are available at https://github.com/baichuan-inc/Baichuan-Audio

1 Introduction

The development of audio dialogue models has shifted from traditional cascade modeling approaches to end-to-end audio large models [12]. Traditional audio dialogue models operate through a sequential framework, where audio input is processed via Automatic Speech Recognition (ASR) to generate text. This text is then used by a language model to produce responses, which are converted into synthesized speech through Text-to-Speech (TTS) systems. While this approach leverages the capabilities of large language models (LLMs), it overlooks the influence of paralinguistic information and introduces processing delays along with errors accumulated through multiple data conversions. This step-by-step processing chain exacerbates these issues, impacting overall performance. In contrast, contemporary end-to-end audio large models, such as GPT-4o [28], adopt a unified framework to directly process audio input, streamlining the task flow and enhancing both understanding and generation of audio content. This evolution facilitates more natural, fluid audio interactions and improves real-time performance.

Recently, open-source end-to-end audio interaction systems have advanced toward achieving real-time performance. These systems can be categorized into three main types. The first, exemplified by Freeze-Omni [37], aligns audio and text modalities through modality adapters, enabling pre-trained large language models to process audio inputs, with hidden states converted into speech waveforms via a speech decoder. While this approach preserves the original capabilities of LLMs, it lacks a holistic end-to-end understanding of the audio modality. The second type, represented by Moshi [7], utilizes discrete audio tokens as input and adopts a multi-stream architecture to output both audio and text concurrently. The third type, such as GLM-4-Voice [42], differs by generating interleaved audio and text outputs, allowing text to guide audio generation for enhanced output quality. However, a key challenge for the second and third

*Corresponding author.

types lies in the integration of audio modalities, which often results in a noticeable reduction in reasoning capabilities compared to textual large language models.

In this work, we introduce Baichuan-Audio, an end-to-end audio large language model designed for real-time speech interaction. Similar to Moshi and GLM-4-Voice, Baichuan-Audio extends pre-trained LLMs to enable end-to-end audio input and output. This is achieved through the integration of the Baichuan-Audio-Tokenizer and a stream-matching decoder, which discretize audio signals into tokens and decode audio tokens back into speech waveforms, respectively. The tokenizer operates at a frame rate of 12.5 Hz and employs multi-codebook discretization to retain both semantic and acoustic information, enabling effective modeling of the speech modality within the LLM. Baichuan-Audio further incorporates an independent audio head to enhance the capability of model to process and capture unique audio features. We conducted large-scale pretraining on audio-text data comprising approximately 100 billion tokens. Based on an extensive audio corpus comprising 887k hours, we implement an interleaved data processing approach, drawing upon the methodology outlined in [16, 27, 43], to facilitate effective knowledge transfer within the LLM framework. To preserve textual understanding during audio modeling, a two-stage pretraining strategy was introduced, where audio embeddings and the audio head were initially trained independently to maintain language comprehension. Baichuan-Audio demonstrates exceptional performance in real-time speech interactions and exhibits robust question-answering capabilities, highlighting its versatility and efficiency. The main contributions of **Baichuan-Audio** can be summarized as follows:

- **Unified and Outstanding Speech Capabilities**: We design an 8-layer RVQ audio tokenizer (Baichuan-Audio-Tokenizer) achieves an optimal balance between capturing semantic and acoustic information with 12.5 Hz frame rate, which supports high-quality controllable bilingual (Chinese and English) real-time conversations.
- **End-to-end Speech Interaction**: Baichuan-Audio is designed to process text, audio inputs, delivering high-quality text and speech outputs. It is capable of delivering high-quality, seamless speech interaction while maintaining intelligent response. At the same time, we have also open-sourced the training data and foundational model, providing valuable resources and tools to advance research and innovation in the field of voice interaction.

2 Related works

2.1 Audio Large Language Models

The development of audio large language models is driven by advancements in speech tokenizers and large language model research. The speech tokenizers serve as a crucial bridge between audio segments and discrete language models by transforming continuous audio signals into discrete tokens. The self-supervised learning models, such as HuBERT[13] and WavLM[2], effectively capture semantic information from speech. The neural acoustic codecs [41, 6, 18] are designed to preserve the full range of audio signal information through discrete encoding. Recent studies [45, 7] have also utilized distilling semantic features sush as pre-trained HuBERT to ensure that specific layers of residual vector quantization (RVQ) retain enriched semantic content. In audio understanding, methods like Qwen-Audio[4, 3], Wavllm[14], and Pengi[8] combine Whisper encoder[31] with large language models, utilizing multi-task learning strategies across speech and language tasks. These approaches have demonstrated excellent performance in speech-to-text tasks. In text-to-speech (TTS), Wang et al. introduced VALL-E[35], conceptualizing TTS as a conditional language modeling problem. To further enhance the efficiency and fidelity of speech synthesis, decoders such as CosyVoice[9] and Matcha-TTS[24] leverage flow-matching techniques, enabling high-quality speech generation.

2.2 End-to-end audio LLM

End-to-end speech interaction models have emerged as a central research focus within the speech processing community, driven by the increasing demand for seamless and efficient multimodal communication systems. Inspired by GPT-4o[28], significant advancements have been made in the development of open-source models. For instance, Moshi [7] introduces an end-to-end full-duplex spoken dialogue foundation model, capable of simultaneously generating audio tokens and text tokens through a multi-stream output mechanism. GLM-4-Voice [42] leverages interleaved data for pre-training to enable text-guided interleaved generation of speech. Freeze-Omni [37] extends the capabilities of large language models (LLMs) to process speech modalities by incorporating modality-specific adapters. It further employs a speech decoder to transform the LLM's output into audio tokens, enabling high-quality speech generation. Following a similar technical approach to Freeze-Omni, models such as VITA-1.5 [11] and SALMONN-Omni [40]. These developments collectively highlight the ongoing progress in creating robust and versatile frameworks for speech-based applications.

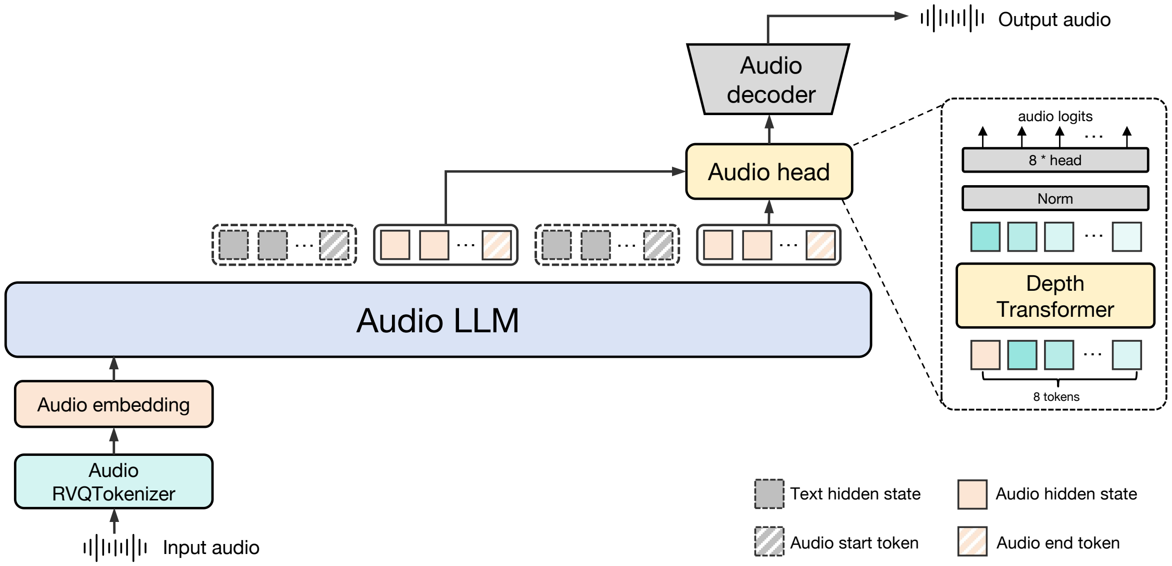

Figure 1: **The overview of Baichuan-Audio.** Our model is an end-to-end large audio language model. When generating audio, the audio LLM alternately predicts text tokens and audio tokens. The audio tokens are then decoded by the flow-matching based audio decoder to produce the final audio.

3 Baichuan-Audio

Baichuan-Audio is an advanced end-to-end audio large language model designed for real-time speech interaction. The overall architecture of the Baichuan-Audio model is presented in Figure 1. The model architecture is structured around three foundational components: the Baichuan-Audio Tokenizer, the audio LLM and an audio decoder. The processing pipeline begins with the audio tokenizer, which converts raw audio input into discrete tokens by capturing both semantic and acoustic information. This is achieved through a combination of high-level feature extraction via the Whisper Encoder and RVQ techniques. The audio LLM then generates aligned text and audio tokens in an alternating manner, facilitated by a specialized token that enables seamless modality switching between text and audio. The audio tokens are subsequently processed by an independent audio head. Finally, a flow-matching based audio decoder reconstructs a high-quality Mel spectrogram from these tokens, which is then converted into an audio waveform via a vocoder. In this section, we will delve into the model architecture of Baichuan-Audio, providing a comprehensive analysis of its training data and the methodologies employed in its development.

3.1 Audio Tokenization

The main challenge for current audio tokenizers lies in achieving an optimal balance between capturing the semantic and acoustic information in the input speech signal [36]. We believe that models such as Baichuan-Omni and Qwen-Audio provide a more direct approach to capturing semantic features, compared to self-supervised learning methods like HuBERT [13]. Meanwhile, audio tokenizer like Encodec [6] and SpeechTokenizer [45] are particularly effective at fully reconstructing audio features. To combine the advantages of these two approaches and inspired by the work of [38], we propose Baichuan-Audio-Tokenizer, an audio tokenizer based on RVQ [6] and multi-objective training, as illustrated in Figure 2. The Baichuan-Audio-Tokenizer retains the audio encoder and the LLM component from Baichuan-Omini [22], while adding an audio decoder structure after the encoder to reconstruct the input Mel spectrogram. The audio tokenizer is trained using a multi-objective optimization approach to effectively capture both semantic and acoustic information.

The Baichuan-Audio-Tokenizer utilizes a frame rate design of 12.5 tokens per second. The high-level audio features are extracted from Mel spectrograms using the Whisper Large Encoder [30], followed by a residual convolutional network performing 4× downsampling to obtain low frame rate audio features. Since the audio features output by the Whisper Encoder are high-dimensional, it is essential to use an 8-layer RVQ to minimize information loss during the quantization process. We design the codebook sizes in a decreasing manner as $\{8K, 4K, 2K, 1K, 1K, 1K, 1K, 1K\}$. The audio decoder employs a fully symmetric structure to the Whisper Encoder, processing its input through a deconvolution module for 4× upsampling. After passing through a series of Transformer layers, the sequence undergoes an additional 2× upsampling, resulting in a coarse Mel-spectrogram representation with 100 tokens per second. Inspired by [20, 25], we design a refined network to improve the accuracy of Mel-spectrogram reconstruction, yielding high-quality refined Mel-spectrogram features. In the design of the audio reconstruction loss, we refer to [25] and adopt a combination of L2 and L1 losses as the reconstruction loss. The reconstruction loss is defined as follows:

$$\begin{aligned}\text{Loss}_{reconstruct} &= L_1(\text{Mel}_{gt}, \text{Mel}_{coarse}) + L_1(\text{Mel}_{gt}, \text{Mel}_{refined}) \\ &+ L_2(\text{Mel}_{gt}, \text{Mel}_{coarse}) + L_2(\text{Mel}_{gt}, \text{Mel}_{refined})\end{aligned} \tag{1}$$

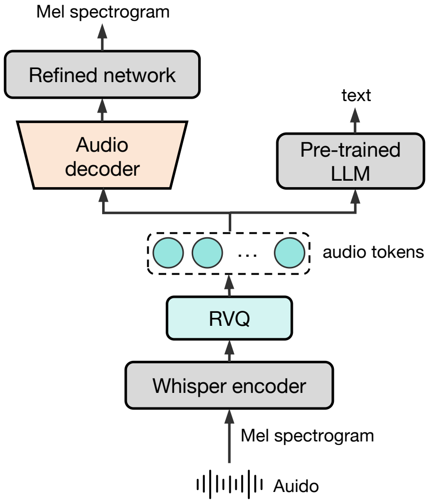

Figure 2: Baichuan-Audio-Tokenizer.

To enhance audio reconstruction quality, we introduce a multi-scale Mel loss approach [18], utilizing two distinct hop lengths and window sizes. This effectively mitigates information loss caused by dimensionality reduction and downsampling interpolation during the transformation from the decoder output to the Mel-spectrogram. By optimizing across multiple scales, the method preserves more detailed audio features, enhancing both reconstruction fidelity and training stability. For the pretrained LLM, the objective is to maximize the softmax probability of text outputs in audio understanding tasks. To ensure semantic alignment, the pretrained LLM maximizes the text softmax probability for audio understanding tasks, with its parameters kept fixed during training to preserve the alignment between the audio tokenizer and the textual LLM space. For the selection of LLM size, we observed during the training of audio understanding models that different LLM sizes have minimal impact on ASR performance metrics. Therefore, we chose a pretrained LLM with 1.5 billion parameters for continued pretraining. This LLM size also aligns well with the Audio Decoder, resulting in smaller gradient norm differences between the two components and improved training stability. We utilize an Exponential Moving Average (EMA) strategy to update the codebook and employ a Straight-Through Estimator (STE) for backpropagating gradients to the encoder. Additionally, we utilize a Vector Quantization (VQ) commitment loss to ensure the encoder outputs align closely with the codebook entries. The VQ commitment loss is defined as:

$$\text{Loss}_{commit} = |z - \text{quantize}(z)|_2^2, \tag{2}$$

The total loss is a weighted combination of the multi-scale reconstruction loss, text-audio alignment loss (for the LLM), and VQ commitment loss:

$$\text{Loss}_{total} = \lambda_1 \cdot \text{Loss}'_{reconstruct} + \lambda_2 \cdot \text{Loss}_{llm} + \lambda_3 \cdot \text{Loss}_{commit} \tag{3}$$

RVQ Training Details. During the VQ training process, we introduce layerwise dropout, randomly discarding the VQ outputs of all layers starting from the second layer onward. This approach ensures that key semantic information is concentrated in the top layers of the codebook, achieving an effect similar to distilling the first-layer codebook in SpeechTokenizer [45]. To enhance codebook utilization, we implement a restart strategy alongside Gumbel sampling. Specifically, if a cluster within the codebook remains unused for a certain number of steps, it is randomly replaced with an input from the current batch. Meanwhile, we employ Gumbel-Softmax sampling during training to introduce stochasticity, ensuring effective exploration of the codebook space. Additionally, to constrain the distribution of the codebook within a specific range, the L2 norm of the codebook is maintained during EMA updates. The final update formula is:

$$c_{j,t} = c_{j,t-1} \times \alpha + \frac{1}{N} \sum_{x_i \in c_{j,t}} x_i \tag{4}$$

$$c_{j,t} = (1 - \beta) \times c'_{j,t} \tag{5}$$

where $c_{j,t}$ denotes the codebook cluster at time step t, $c_{j,t-1}$ is its previous state, α is the EMA decay factor, N is the number of inputs assigned to the cluster, and x_i represents the input samples in the current batch associated with $c_{j,t}$. Here, $c'_{j,t}$ represents the intermediate update of the cluster before normalization, and β is the constraint factor used to regulate the L2 norm of the codebook vector, ensuring its distribution remains within a bounded range. For the L2 norm constraint, $c'_{j,t}$ is the intermediate updated cluster before normalization, and β is the constraint factor controlling the compression of the distribution of codebook vector. We observed that L2 norm constraints significantly reduce the mutual information of the codebook, concentrating the VQ distribution on a few top token IDs. This helps simplify downstream tasks, such as TTS, by improving the prediction accuracy of audio tokens. However, excessively strong L2 norm constraints can lead to codebook redundancy and reduce overall codebook utilization. Therefore, the β parameter in the formula requires careful fine-tuning. To address this, we adopt a multi-stage progressive training strategy, gradually increasing the proportion of Whisper Encoder outputs replaced by VQ results, from 10% to 100%, while simultaneously adjusting the weight of the commit loss. We found that replacing the entire sample with VQ is more effective than randomly replacing individual tokens with VQ, i.e., instance-level replacement is better than token-level replacement.

Training Data. In addition to traditional tasks such as Automatic Speech Recognition (ASR), Audio Query Answering (AQA), and Speech-to-Text Translation (S2TT), we incorporate a proportion of audio-text interleaved data into the training process. This strategy aims to enhance the VQ module's capability to model complex contextual scenarios. Specifically, the training dataset consists of 135k hours of ASR data, 11k hours of AQA data, 9k hours of S2TT translation data, and 52k hours of audio-text interleaved data. Further details about the dataset are provided in Section 3.3.

Evaluation of Baichuan-Audio-Tokenizer. To evaluate the performance of the tokenizer in terms of pre-training and post-training losses, we trained a non-VQ version of the audio understanding model as a baseline using the same data and base model. For both the VQ and non-VQ models, the parameters of the LLM were kept frozen during training to ensure a fair comparison and to isolate the impact of the VQ mechanism on the overall performance. From Table 1, we can see that the 8-layer vq is closer to the baseline and has the least loss of semantic content. As shown in Table 2, the ASR results of the 8-layer VQ model and the baseline across multiple datasets demonstrate that the trained 8-layer VQ model achieves competitive performance.

Table 1: **Comparison of ASR Performance across VQ Models with Different Layer Counts.** Librispeech* results are averages across dev-clean, dev-other, test-clean, and test-other. AiShell1* is the average of eval and dev results.

Model	LibriSpeech* WER (%)	AiShell1* WER (%)	LibriSpeech* Mel MAE	AiShell1* Mel MAE
Baseline	3.8	2.0	-	-
8 Layers VQ	5.3	2.7	0.466	0.403
6 Layers VQ	5.5	3.0	0.485	0.423
4 Layers VQ	6.3	3.9	0.516	0.488
1 Layer VQ	26.2	19.2	0.677	0.553

Table 2: **Comparison of ASR Performance between the Baseline and the 8-layer VQ Model.** The evaluation metric is WER(%).

Dataset	**Baseline**	**VQ Model (8 Layers)**
Fleurs-ZH	3.54	4.15
Fleurs-EN	5.98	8.64
Med-ZH-inhouse	4.39	6.03
Wenet testnet	7.32	8.29
Wenet testmeeting	8.54	9.03
Covost-2 en2zh BLEU	33.94	30.4
Covost-2 zh2en BLEU	21.7	19.7
Clotho AQA	40.4	37.2

3.2 Flow-matching based audio decoder

To improve the quality and fidelity of synthesized audio, the audio decoder module is enhanced using a flow-matching model [23], trained on 24 kHz audio to generate target Mel spectrograms. The flow-matching decoder includes a Pre-Net and a conditional decoder, as shown in Figure 3. The Pre-Net maps intermediate representations to a prior for the vocoder, using an MLP and a 12-layer transformer. The MLP projects 1280-dimensional, 50 Hz features to 512 dimensions, refined by the transformer, and a final linear layer converts them into 80-dimensional Mel spectrograms.

The flow-matching conditional decoder uses a U-Net structure trained with OT-CFM, inspired by Matcha-TTS [24] and CosyVoice [9]. The U-Net includes one down-sampling block, one up-sampling block, and 12 intermediate blocks, each with a ResNet1D and transformer layer (256 dimensions). A linear layer projects features to 80-dimensional Mel spectrograms. As the model already encodes acoustic information (e.g., speaker timbre) via reconstruction loss, no additional speaker embeddings are used. The generated Mel spectrograms are converted into waveforms using the HiFi-GAN [17, 9] vocoder[2].

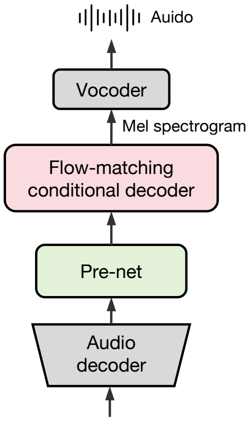

Figure 3: Flow-matching based audio decoder.

Training Details. The flow-matching model was trained on about 270k hours of audio, including Mandarin, English, various dialects, and multilingual data. Data quality was refined using ensemble ASR and MOS filtering. During training, the AudioEncoder, VQ layers, and AudioDecoder were fixed, while the flow-matching Pre-Net and decoder were trained with a prior loss added to the Pre-Net.

Reconstruction result. We evaluated the performance of the trained flow-matching network on the LibriSpeech-dev set. UTMOS [32] was used as a proxy for subjective audio perception, with the score improving from 3.43 to 4.05, closely approaching the ground truth score of 4.08. Content quality was assessed using Whisper ASR [31], where the Word Error Rate (WER) decreased from 2.84 to 2.78.

Table 3: **Evaluation of reconstruction performance**. We evaluate the performance of subjective audio perception and content quality on librispeech-dev set.

	UTMOS[2]	ASR-WER
groundtruth	4.08	2.26
VQ	3.43	2.84
+Flow-matching	4.05	2.78

3.3 Audio LLM

The Baichuan-Audio extends a pre-trained LLM by incorporating the newly introduced Baichuan-Audio-Tokenizer, which includes audio embedding layers and an independent audio head. Specifically, the audio tokens from Baichuan-Audio-Tokenizer are first transformed into audio embeddings through audio embedding layers. The audio LLM alternately generates aligned text tokens and audio tokens, facilitated by a special token that enables modality switching between text and audio. The generated audio tokens are processed by an independent audio head, consisting of 3 layers of depth transformers and 8 classification heads. Finally, the audio embeddings are passed through an audio encoder, such as a flow-matching-based audio encoder and vocoder, to reconstruct audio waveforms.

Audio embedding. First, the 8 discrete audio token are summed through a corresponding number of embedding layers to obtain audio embeddings. Each embedding layer takes an input dimension that is one greater than the size of the corresponding codebook, due to the inclusion of an additional special token that signifies the end of audio token generation.

Audio head. The generated audio tokens are processed using an independent audio head, which consists of 3 layers of depth transformers and 8 classification heads. The depth transformer has a depth of 8, predicting audio embeddings for 8 codebooks. Finally, the classification heads are used to obtain the logits for each codebook corresponding to the audio tokens.

Compared to pure text-based large models, speech language models often struggle to generate semantically coherent outputs. Research in [36] indicates that this issue primarily arises from the introduction of duration and paralinguistic information in speech. To address this issue, two types of interleaved data are employed during pre-training: Audio-Text Interleaved (INTLV) and Interleaved Text-to-Speech (ITTS), which contribute to enhancing audio understanding and generation capabilities. During inference, discrete audio tokens are fed into the LLM, and the model alternately generates aligned text tokens and audio tokens. Special tokens facilitate modality switching between text and audio. This

[2] https://www.modelscope.cn/models/iic/CosyVoice2-0.5B

[2] https://github.com/sarulab-speech/UTMOS22

forced alignment approach ensures that the model generates coherent and complete textual content before synthesizing the corresponding audio tokens, effectively guiding the generation of audio tokens and mitigating the issue of semantic degradation.

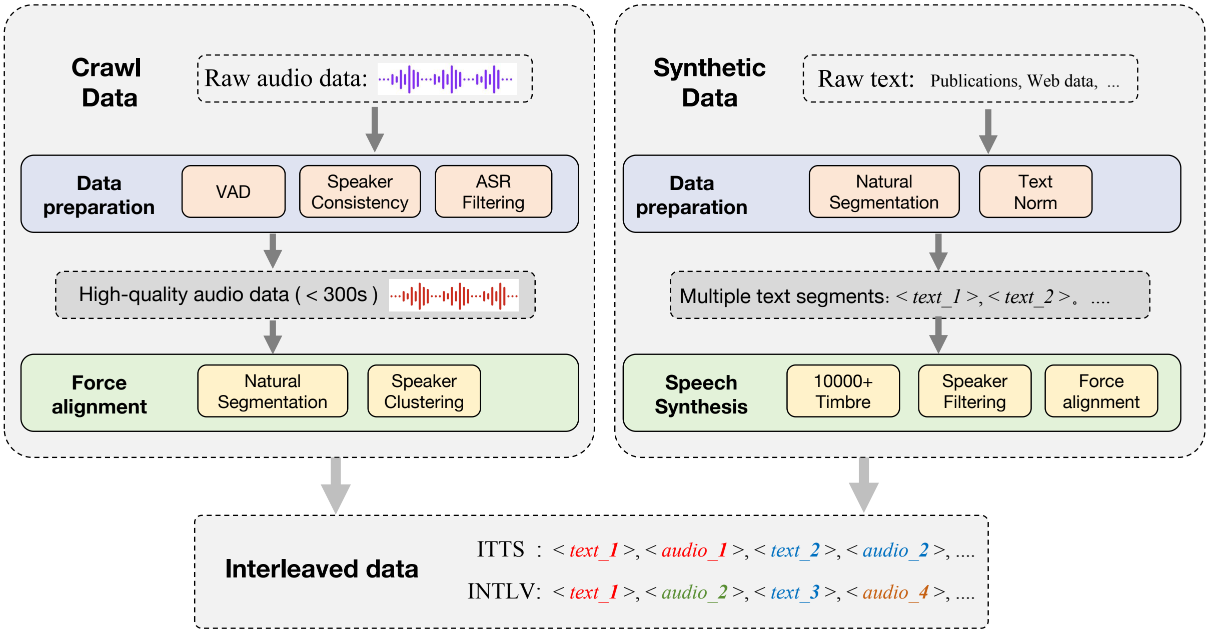

Figure 4: Pipeline of interleaved data collection.

3.3.1 Pre-training details

Pre-training data. The detailed statistics of the training data for audio pre-training are shown in Table 4. Audio data can be broadly categorized into two primary types: audio understanding data and audio generation data. Audio understanding data includes Automatic Speech Recognition (ASR), Audio Question Answering (AQA), Speech-to-Text Translation, and Audio-Text Interleave data. Audio generation data encompasses Text-to-Speech (TTS), Interleaved Text-to-Speech data, and pure audio data. Interleaved data consists of alternating text and audio modalities, segmented by punctuation marks to facilitate cross-modal knowledge transfer. The interleaved aligned generation data composed of fully aligned text and audio content, designed to enhance the model's ability to generate audio tokens under text supervision. The audio-text paired data (e.g., ASR and TTS data) improve the performance on fundamental speech tasks. Pure audio data, on the other hand, enhances the capability to independently process audio modalities. The interleaved data collection process is shown in Figure 4, which is divided into two types: crawl data and synthetic data. In total, we obtained 142k hours of ITTS data and 393k hours of INTLV data. The interleaved data is segmented using LLM, which performs natural segmentation based on punctuation or natural pauses in the text content. For the segmented text data of synthetic data, we also use a large language model for text normalization [46]. During pre-training, we excluded the loss

Table 4: Detailed statistics of the training data of audio pretrain.

Type	Task	Data Format	Hours (k)
Audio Understanding	Automatic Speech Recognition (ASR)	`<prompt, audio, transcript>`	185
	Audio Query Answer (AQA)	`<prompt, audio, response>`	21
	Speech-to-Text Translation (S2TT)	`<prompt, audio, translated_text>`	15
	Audio-Text Interleaved (INTLV)	`<audio_1, text_2, audio_3, text_4, ...>`	393
Audio Generation	Text-to-Speech (TTS)	`<text, audio>`	51
	Interleaved Text-to-Speech (ITTS)	`<text_1, audio_1, text_2, audio_2, ...>`	142
	Pure Audio	`<audio>`	80
Total	-	-	887

computation for the audio segments in the Audio-Text Interleaved data, a design choice that differs from GLM-4-Voice. Empirical observations under the current about 50B training audio data scale revealed that computing loss for the audio segments in INTLV data led to performance degradation. This decision is further justified by the inherent modality conflict between audio and text, as well as the absence of a requirement for text-to-audio continuation during inference. Consequently, we omitted the loss calculation for the audio segments in INTLV data. For ITTS data, loss was computed for both audio and text segments, except for the initial text segment, to enhance the model's capability in text-guided audio generation.

Two stage training strategy. To address the potential disruption of the original textual knowledge in the LLM caused by the distinct characteristics of speech features compared to text features, we propose a two-stage training strategy to mitigate training conflicts between modalities. In the first stage, the parameters of the LLM remain fixed, allowing only the parameters of the audio embedding layer and the audio head to be updated. In the second stage, all parameters, except for those of the text embedding layer and the LM head, are made trainable.

3.3.2 Supervised fine-tuning details

The supervised fine-tuning stage aims to enhance the model's ability to follow complex instructions across a range of tasks. The audio SFT data are derived from a large collection of textual instructions. High-quality instructions are selected using a filtering strategy based on instruction type, diversity, and overall quality. Audio instructions are synthesized using a curated dataset of 10,000 distinct voice tones. Corresponding text responses are generated and segmented at natural conversational pauses before being converted into audio using the designated voice tones. These datasets cover multiple tasks and contain approximately 242k audio data pairs.

To ensure the quality of the synthesized audio, Automatic Speech Recognition (ASR) is applied to the generated audio files. The ASR outputs are compared against the original text to validate quality. This process results in the creation of high-quality end-to-end conversational datasets. Synthesized audio files with errors are added to the Text-to-Speech (TTS) dataset, while cases with ASR errors are incorporated into the ASR training dataset. This iterative approach of incorporating challenging examples enhances both TTS and ASR performance.

Special attention is required to address cases where text-to-audio conversion makes the original textual response unsuitable as an audio reply. This issue arises due to differences in tone, speed, and expression between text and audio. Some textual content may fail to convey the intended meaning or introduce ambiguity when converted into audio. Consequently, careful review and adjustment of such cases are essential during the generation process. This ensures that the synthesized data accurately reflects real-world voice interaction scenarios, enhancing data reliability and improving the model's practical applicability.

4 Experiment

4.1 General Intelligence Evaluation

A major challenge faced by speech-based dialogue models is that, compared to pure text-based dialogue models, their performance tends to degrade. To evaluate the "intelligence" of speech models, we use text-to-text modeling capabilities as a baseline and assess the performance of pre-trained speech-to-text models. The evaluation dataset consists of two types: story continuation ability and commonsense reasoning ability.

Sentence continuation. For the continuation ability evaluation, we use the sStoryCloze dataset [12]. Additionally, we introduce the zh-sStoryCloze dataset, which is created by translating the English version of sStoryCloze into its Chinese counterpart via a translation engine and replacing English names with Chinese ones to better suit the Chinese context. Each sample in both evaluation sets consists of five sentences, divided into positive and negative samples. The last sentence differs between the two, with the last sentence of the positive sample being the correct continuation. A prediction is considered correct if the perplexity of the last sentence in the positive sample is lower than that of the negative sample.

Commonsense reasoning. For the commonsense reasoning ability evaluation, the goal is to assess whether the model possesses domain-specific knowledge. Drawing inspiration from the design of sStoryCloze, we use the GPT-4o API to rewrite and filter the CMMLU dataset [19], ultimately creating the sCMMLU dataset with 4,743 commonsense questions. For each multiple-choice question in the original CMMLU, we rewrite it into four statements with the same first half and different second halves according to the answer options. A prediction is considered correct if the perplexity of the correct option's statement is lower than that of the other options.

Results. The general intelligence evaluation results in two aspects are shown in Table 5. It can be observed that Baichuan-Audio consistently outperforms previous models in the sentence continuation evaluation task. Given the

Table 5: **Performance Comparison on Various Evaluation Tasks.** ∗: Evaluations were performed using the instruct model as no base model was provided.

Model	**Modality**	**Params**	**Evaluation Datasets**		
			sStoryCloze	**zh-sStoryCloze**	**sCMMLU**
TWIST	S → T	7B	53.3	-	-
Moshi	S → T	7B	60.8	-	-
GLM-4-Voice	S → T	9B	76.3	70.3∗	64.3∗
Baseline	T → T	7B	83.0	76.1	70.3
Our (single stage)	S → T	7B	77.5	70.1	67.0
Our (two stage)	S → T	7B	79.6	72.4	69.3

current architecture, the intelligence of the speech multimodal model is inherently dependent on the reasoning capability of the pure text LLM. Consequently, the accuracy of the T→T mode serves as the upper bound for end-to-end speech models, while the accuracy of the S→T mode is inherently lower than that of the T→T mode. Our goal is to bridge this gap by improving the accuracy of the S→T mode to approach that of the T→T mode, thereby enhancing the intelligence of end-to-end speech models. Moreover, we observe that a two-stage training strategy effectively mitigates the degradation in model intelligence compared to single-stage training.

4.2 Performance in ASR/TTS Tasks

For ASR evaluation in the general scene, we report results on the Fleurs [5] Chinese (*zh*) and English (*en*) test sets, as well as the WenetSpeech [44] *test_net* dataset. To assess performance in more challenging ASR scenarios, we include results from the WenetSpeech [44] *test_meeting* dataset and the KeSpeech [33] test set, which evaluate the model's ASR capabilities in 'Meeting' and 'Chinese dialect' contexts. Baichuan-Audio exhibits a strong audio transcription capacity in Table 6. On the Fleurs dataset (test-zh), it achieves a WER of 3.2%, significantly lower than Whisper-large-v3 (12.4%) and Qwen2-Audio-Base (4.3%). For the WenetSpeech dataset, Baichuan-Audio-Base achieves a WER of 7.2% on test_net and 8.5% on test_meeting. On the KeSpeech dataset, Baichuan-Audio-Base excels across multiple Chinese dialects. In addition to ASR, Baichuan-Audio excels in both S2TT and TTS tasks. For S2TT task, which aims to translate the audio signal in the source to the target language. We evaluate the model's S2TT performance between Chinese and English using the zh2en and en2zh subsets of the Covost2 [34] dataset, with BLEU [29] scores as the evaluation metric. The evaluation results of S2TT and TTS tasks are summarized in Table 7.

4.3 Performance in Audio Understanding Tasks

Baselines. We compare Baichuan-Audio with the following baselines: proprietary model (GPT-4o-Audio [28]), open-source voice model (GLM-4-Voice [42]), and open-source models for omni-modal (VITA-1.5 [11], MiniCPM-o 2.6 [39]).

Evaluation Benchmarks. To assess the audio understanding capabilities of Baichuan-Audio, we have built and open-sourced an OpenAudioBench and use GPT-4o [28] to evaluate the results, including Reasoning QA(self-constructed), Spoken Llama Questions [26], Web Questions [1], TriviaQA [15], and AlpacaEval [21]. For AlpacaEval, we select two subsets `helpful base` and `vicuna` from the original AlpacaEval dataset and remove questions related to math and code. This process follows Llama-Omni [10], with the aim of obtaining questions more suitable for speech scenarios, and the final AlpacaEval benchmark in our report comprises 199 questions in total. Considering the substantial size of the Web Questions and TriviaQA datasets, a full evaluation is impractical. Therefore, we randomly sample 1,000 questions from each original dataset. The instructions for these three benchmarks were synthesized using our TTS model.

For Reasoning QA, we use GPT-4o to evaluate the score of the answers based on the given reference answers, and then calculate the accuracy rate. For Llama Questions, Web Questions, and TriviaQA, we provide reference answers and use GPT-4o to assess the correctness of the model's responses. The final score is the percentage of answers judged as correct.

For all audio benchmarks, we consider two different settings: 1) speech-to-speech generation in a non cascaded manner (denoted as S→S), where the input is audio and the output is interleaved text and audio. The output text is then merged and used for evaluation. 2) speech-to-text generation (denoted as S→T, where the input is audio and the output is text, which is used for evaluation.

Table 6: **Major results on Fleurs, WenetSpeech, and KeSpeech**. The test sets are evaluated with WER. The rest unlabeled results are reproduced by ourselves, and any performance divergence may be attributed to differences in decoding parameters.

Scene	**Dataset**	**Model**	**Results** WER (%) ↓
General	Fleurs *test-zh*	Whisper-large-v3 (1.55B) Qwen2-Audio-Base (7B) Baichuan-Audio-Base (7B)	12.4 4.3 **3.2**
	WenetSpeech *test_net*	Whisper-large-v3 (1.55B) Qwen2-Audio-Base (7B) Baichuan-Audio-Base (7B)	17.5 7.3 **7.2**
Meeting	WenetSpeech *test_meeting*	Whisper-large-v3 (1.55B) Qwen2-Audio-Base (7B) Baichuan-Audio-Base (7B)	30.8 **7.7** 8.5
Chinese dialect	KeSpeech *mandarin* | *beijing* | *southwest* *lan-yin* | *zhongyuan* | *northeast* *jiang-huai* | *ji-lu* | *jiao-liao*	Whisper-large-v3 (1.55B)	18.7 | 44.8 | 52.9 54.8 | 50.1 | 22.9 54.7 | 47.0 | 50.4
		Qwen2-Audio-Base (7B)	3.0 | 7.5 | **6.2** **6.7** | **5.0** | 5.5 **9.1** | **6.6** | **7.0**
		Baichuan-Audio-Base (7B)	**2.7** | **6.9** | 6.8 7.2 | 5.3 | **5.0** 9.9 | 6.7 | 7.4

Table 7: **Evaluation the capabilities of the automatic speech translation and text-to-speech on base model**.

Model	**Translation (BLEU)**		**TTS (ASR)**
	Covost-2 zh-CN2en	Covost-2 en2zh-CN	MED-TTS
Qwen2-Audio-Base (7B)	22.17	43.58	-
Baichuan-Audio-Base (7B)	24.37	45.96	2.71

Results. As shown in Table 8, our model performs excellently on audio understanding benchmarks, outperforming the latest open-source models. In the S→T setting, Baichuan-Audio significantly outperforms models of the same size in AlpacaEval, achieving score of 77.4. In the S→S setting, Baichuan-Audio surpasses GLM-4-Voice across the board, particularly leading by 11.4 and 20.7 in Reasoning QA and AlpacaEval.

Table 8: **Results on audio understanding benchmarks.** ∇: The modalities parameter is set to ["text", "audio"], evaluation based on the output text. ♢: Supports only text-audio interleaved output. □: Cascade output method, evaluation based on the output text.

Model	**Audio Comprehensive Capacity**									
	Reasoning QA		Llama Questions		Web Questions		TriviaQA		AlpacaEval	
	$s \to t$	$s \to s$	$s \to t$	$s \to s$	$s \to t$	$s \to s$	$s \to t$	$s \to s$	$s \to t$	$s \to s$
GPT-4o-Audio$^{\nabla}$	**55.6**	-	**88.4**	-	**81.0**	-	**90.6**	-	**80.1**	-
GLM-4-Voice (9B)$^{\diamondsuit}$	-	26.5	-	71.0	-	51.5	-	46.6	-	48.9
VITA-1.5 (7B)$^{\square}$	41.0	-	74.2	-	57.3	-	46.8	-	68.2	-
MiniCPM-o 2.6 (7B)$^{\square}$	38.6	-	77.8	-	68.6	-	61.9	-	51.8	-
Baichuan-Audio (7B)	41.9	**37.9**	78.4	**74.5**	64.5	**60.3**	61.7	**54.2**	77.4	**69.6**

5 Conclusion

In this paper, we introduce Baichuan-Audio, an end-to-end large language model designed for audio that integrates both speech comprehension and generation. The model employs a multi-codebook discretization of speech signals at 12.5 Hz via a pre-trained ASR model, which preserves both semantic and acoustic information in speech tokens. Additionally, an independent audio head is specifically designed to process these tokens efficiently. To balance audio modeling and language capability preservation, a two-stage pre-training strategy with interleaved data is adopted. The proposed framework supports speech interaction through text-guided aligned speech generation, thereby further retaining the model's foundational cognitive abilities. With open-sourced training data and models, Baichuan-Audio makes a significant contribution to the advancement of real-time speech interaction systems.

References

[1] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, 2013.

[2] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[3] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

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

[5] Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. Fleurs: Few-shot learning evaluation of universal representations of speech. *arXiv preprint arXiv:2205.12446*, 2022.

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

[7] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

[8] Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. Pengi: An audio language model for audio tasks. *Advances in Neural Information Processing Systems*, 36:18090–18108, 2023.

[9] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024.

[10] Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*, 2024.

[11] Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Yangze Li, Zuwei Long, Heting Gao, Ke Li, et al. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025.

[12] Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, et al. Textually pretrained speech language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[13] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

[14] Shujie Hu, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Hongkun Hao, Jing Pan, Xunying Liu, Jinyu Li, Sunit Sivasankaran, et al. Wavllm: Towards robust and adaptive speech large language model. *arXiv preprint arXiv:2404.00656*, 2024.

[15] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[16] Heeseung Kim, Soonshin Seo, Kyeongseok Jeong, Ohsung Kwon, Jungwhan Kim, Jaehong Lee, Eunwoo Song, Myungwoo Oh, Sungroh Yoon, and Kang Min Yoo. Unified speech-text pretraining for spoken dialog modeling. *arXiv preprint arXiv:2402.05706*, 2024.

[17] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

[18] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*, 2023.

[20] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6706–6713, 2019.

[21] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

[22] Yadong Li, Haoze Sun, Mingan Lin, Tianpeng Li, Guosheng Dong, Tao Zhang, Bowen Ding, Wei Song, Zhenglin Cheng, Yuqi Huo, et al. Baichuan-omni technical report. *arXiv preprint arXiv:2410.08565*, 3(7), 2024.

[23] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[24] Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. Matcha-tts: A fast tts architecture with conditional flow matching. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11341–11345. IEEE, 2024.

[25] Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, et al. Autoregressive speech synthesis without vector quantization. *arXiv preprint arXiv:2407.08551*, 2024.

[26] Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. Spoken question answering and speech continuation using spectrogram-powered llm. *arXiv preprint arXiv:2305.15255*, 2023.

[27] Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, et al. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025.

[28] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[30] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[31] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[32] Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*, 2022.

[33] Zhiyuan Tang, Dong Wang, Yanguang Xu, Jianwei Sun, Xiaoning Lei, Shuaijiang Zhao, Cheng Wen, Xingjun Tan, Chuandong Xie, Shuran Zhou, Rui Yan, Chenjia Lv, Yang Han, Wei Zou, and Xiangang Li. Kespeech: An open source speech dataset of mandarin and its eight subdialects. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[34] Changhan Wang, Anne Wu, and Juan Pino. Covost 2 and massively multilingual speech-to-text translation. 2020.

[35] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

[36] Hankun Wang, Haoran Wang, Yiwei Guo, Zhihan Li, Chenpeng Du, Xie Chen, and Kai Yu. Why do speech language models fail to generate semantically coherent outputs? a modality evolving perspective. *arXiv preprint arXiv:2412.17048*, 2024.

[37] Xiong Wang, Yangze Li, Chaoyou Fu, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*, 2024.

[38] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

[39] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[40] Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. Salmonn-omni: A codec-free llm for full-duplex speech understanding and generation. *arXiv preprint arXiv:2411.18138*, 2024.

[41] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

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

[43] Aohan Zeng, Zhengxiao Du, Mingdao Liu, Lei Zhang, Shengmin Jiang, Yuxiao Dong, and Jie Tang. Scaling speech-text pre-training with synthetic interleaved data. *arXiv preprint arXiv:2411.17607*, 2024.

[44] Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, Di Wu, and Zhendong Peng. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. 2022.

[45] Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechtokenizer: Unified speech tokenizer for speech language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] Yang Zhang, Travis M Bartley, Mariana Graterol-Fuenmayor, Vitaly Lavrukhin, Evelina Bakhturina, and Boris Ginsburg. A chat about boring problems: Studying gpt-based text normalization. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10921–10925. IEEE, 2024.

A Appendix

A.1 Open-Source Data for training

Table 9: Open-Source Dataset Summary

Dataset	Type	Sizes
AISHELL 1	Chinese	500 hours
AMI-IHM/SDM	English	100 hours
AudioCaps	English	46,000 audios
AudioSet	Multilingual	5,800 hours
Clotho	English	4981 audios
CMU-MOSEI	-	65 hours
Common Voice	Multilingual	31,000
covost2	Multilingual	2900
Emilia	Multilingual	101k
earnings22	English	119 hours (57k clips)
Fluent Speech Commands	English	10 hours
fma	Music	-
fsd50k	-	51k clips (108h)
GigaSpeech	English	10,000 hours
gigaspeech2	Southeast Asian	30,000 hours
Google FLEURS	Multilingual	10 hours
kespeech	Chinese Dialects	1,500+ hours
LibriSpeech	English	1,000 hours
LibriTTS	English	585 hours
LJSpeech	English	24 hours
MAGICDATA	Chinese	700+ hours
Multilingual LibriSpeech	Multilingual	-
Multilingual TEDx	Multilingual	-
NMSQA_audio	English	-
Parler TTS	Multilingual	54k hours
peoples_speech	Multilingual	30,000+ hours
SPGISpeech	English	5,000+ hours
TED-LIUM	English	452 hours
vggsound	-	550+ hours
wenetspeech4tts	Chinese	12.8k+ hours
WenetSpeech	Chinese	10,000 hours
zhvoice	Chinese	960 hours