Kimi-Audio Technical Report

Kimi Team

Abstract

We present Kimi-Audio, an open-source audio foundation model that excels in audio understanding, generation, and conversation. We detail the practices in building Kimi-Audio, including model architecture, data curation, training recipe, inference deployment, and evaluation. Specifically, we leverage a 12.5hz audio tokenizer, design a novel LLM-based architecture with continuous features as input and discrete tokens as output, and develop a chunk-wise streaming detokenizer based on flow matching. We curate a pre-training dataset that consists of more than 13 million hours of audio data covering a wide range of modalities including speech, sound, and music, and build a pipeline to construct high-quality and diverse post-training data. Initialized from a pre-trained LLM, Kimi-Audio is continual pre-trained on both audio and text data with several carefully designed tasks, and then fine-tuned to support a diverse of audio-related tasks. Extensive evaluation shows that Kimi-Audio achieves state-of-the-art performance on a range of audio benchmarks including speech recognition, audio understanding, audio question answering, and speech conversation. We release the codes, model checkpoints, as well as the evaluation toolkits in https://github.com/MoonshotAI/Kimi-Audio.

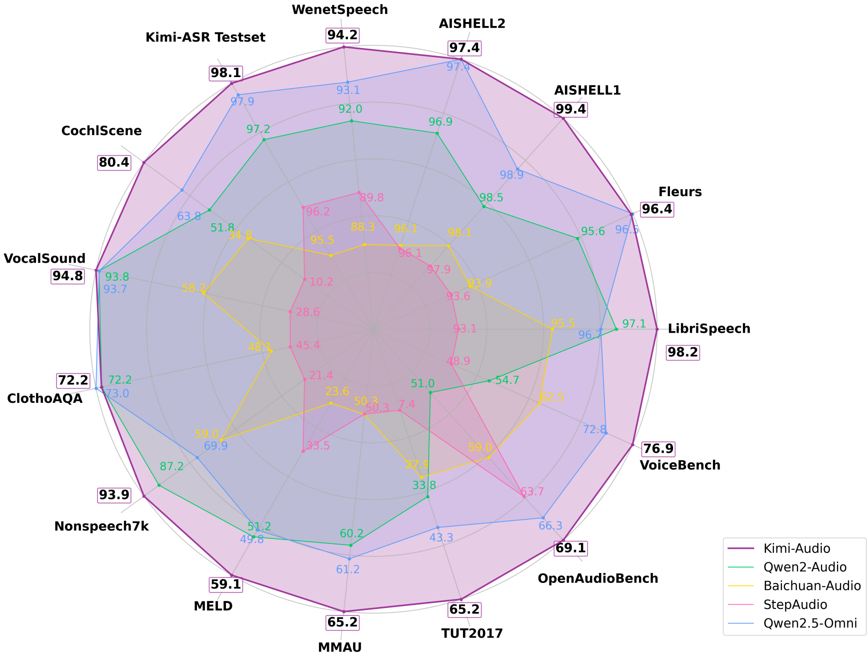

Figure 1: Performance of Kimi-Audio and previous audio langauge models including Qwen2-Audio [11], Baichuan-Audio [41], Step-Audio [28], and Qwen2.5-Omni [73] on various benchmarks.

Kimi-Audio 技术报告

Kimi 团队

摘要

我们提出了Kimi-Audio，一款开源的音频基础模型，在音频理解、生成和对话方面表现出色。我们详细介绍了构建Kimi-Audio的实践，包括模型架构、数据整理、训练方案、推理部署和评估。具体而言，我们利用了一个12.5Hz的音频分词器，设计了一种基于大规模语言模型（LLM）的新颖架构，采用连续特征作为输入，离散标记作为输出，并开发了基于流匹配的分块流式解码器。我们整理了一个预训练数据集，包含超过1300万小时的音频数据，涵盖语音、声音和音乐等多种模态，并构建了一个管道以生成高质量、多样化的后训练数据。Kimi-Audio从预训练的LLM初始化，通过在音频和文本数据上进行多次精心设计的任务的持续预训练，然后进行微调，以支持多样的音频相关任务。大量评估显示，Kimi-Audio在包括语音识别、音频理解、音频问答和语音对话在内的多个音频基准测试中达到了最先进的性能。我们在https://github.com/MoonshotAI/Kimi-Audio上发布了代码、模型检查点以及评估工具包。

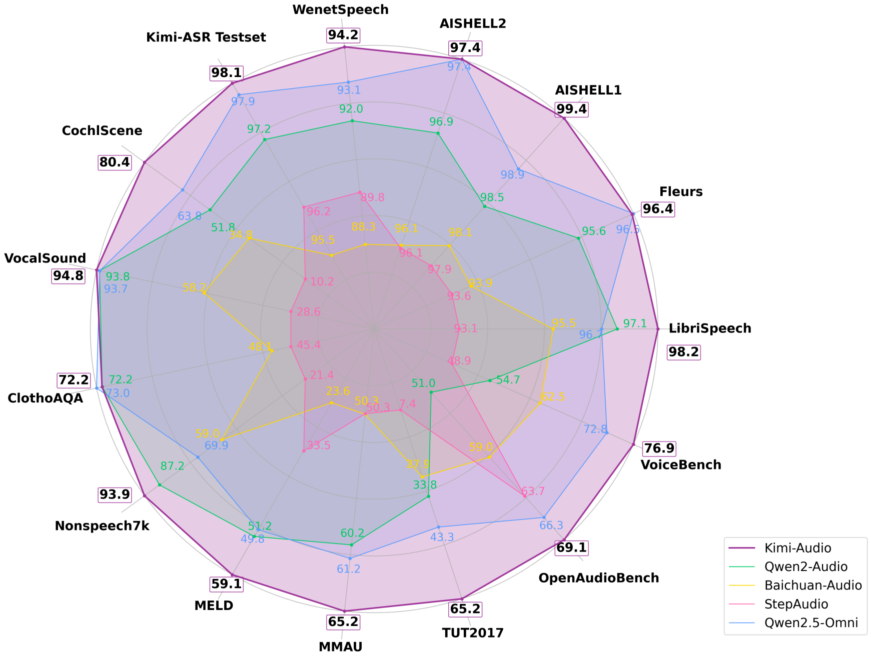

图1：Kimi-Audio及之前的音频语言模型在各种基准测试中的表现，包括Qwen2-Audio [11]、Baichuan-Audio [41]、Step-Audio [28]和Qwen2.5-Omni [73]。

1 Introduction

Audio plays an indispensable role in human daily life, such as environment perception, speech conversation, emotion expression, and music appreciation, and is an important topic in artificial general intelligence. Traditional audio modeling, constrained by the development of artificial intelligence, handles each audio processing task (e.g., speech recognition, emotion recognition, sound event detection, and speech conversation) separately. However, audio is naturally sequential and speech has strict correspondence with text, which makes it suitable to take advantage of the rapid progress in large language models (LLMs) in audio modeling. Just as natural language processing has experienced, audio processing evolves quickly from separate models for separate tasks to a universal model handling a variety of tasks.

For example, pioneer works introduce language models in audio generation [3, 77], audio understanding [10, 63, 9], speech recognition [58, 87], speech synthesis [70, 80], and end-to-end speech conversation [29, 14]. However, previous works fall short of building a universal audio foundation model for a variety of audio processing tasks in several aspects: 1) not universal but only focus on a specific type of tasks, such as audio understanding [10, 11, 21, 63, 88, 35], audio generation [45, 77], or speech conversation [14, 84]; 2) not much emphasis on audio pre-training but only fine-tuning an LLM on downstream audio tasks [10, 63, 9]; 3) no access to source codes and checkpoints, with limited value to the community [29, 7].

In this report, we present Kimi-Audio, an open-source audio foundation model that handles a variety of audio processing tasks. We detail our effort in building a state-of-the-art (SOTA) audio foundation model in three espects: architecture, data, and training.

- Architecture. Our model consists of three components: an audio tokenizer and detokenizer as audio I/O, and an audio LLM as the core processing part (see Section 2.1). We use discrete semantic audio tokens as the basic representation for both the input and output of the audio LLM. Meanwhile, we concatenate the semantic audio token with continuous acoustic vectors in the input to enhance perception capability, and concatenate with discrete text tokens in the output to enhance the generation capability. In this way, we can achieve good audio perception and generation capabilities at the same time, facilitate universal audio modeling. We reduce the number of token per second in audio to bridge the gap between text and audio sequence and set the compression rate of both the semantic and acoustic audio tokens as 12.5Hz. The detailed design of the audio tokenizer for both discrete semantic tokens and continuous acoustic vectors, as well as the generation of both discrete semantic tokens and text tokens are introduced in Section 2.2 and 2.3 respectively.
- Data. To achieve SOTA universal audio modeling, we need to pre-train the model on a large amount of audio data to see diverse scenarios. To this end, we crawl and process a large-scale audio pre-training dataset. We develop a data processing pipeline consisting of speech enhancement, diarization, transcription, filtering, etc, to enable high data quality (see Section 3.1). To support diverse audio processing tasks, we curate a large amount of task-specific data for supervised fine-tuning (SFT). We demonstrate an economic way to construct most of SFT data with pure open and accessible data sources and processing tools to achieve SOTA performance, without relying on any data purchase (see Section 3.2).
- Training. To achieve good audio understanding/generation capability while maintaining high knowledge capacity and intelligence, we initialize the audio LLM with a pre-trained LLM, and carefully design a series of pre-training tasks to fully learn the audio data and bridge the gap

1 引言

音频在人的日常生活中扮演着不可或缺的角色，例如环境感知、语音对话、情感表达和音乐欣赏，是人工通用智能中的一个重要课题。传统的音频建模受到人工智能发展的限制，分别处理每个音频处理任务（例如语音识别、情感识别、声音事件检测和语音对话）。然而，音频具有自然的序列性，语音与文本之间具有严格的对应关系，这使得利用在音频建模中快速发展的大型语言模型（LLMs）成为可能。正如自然语言处理所经历的那样，音频处理也在快速演变，从为各个任务单独建立模型，发展到一个通用模型能够处理多种任务。

例如，先驱工作在音频生成 [3, 77]、音频理解 [10, 63, 9]、语音识别 [58, 87]、语音合成 [70 , 80] 和端到端语音对话 [29, 14] 中引入了语言模型。然而，之前的工作在多个方面未能构建一个通用的音频基础模型，以应对各种音频处理任务：1）不是通用的，而仅关注特定类型的任务，例如音频理解 [10, 11, 21, 63, 88, 35]、音频生成 [45, 77] 或语音对话 [14, 84] ；2）对音频预训练的重视不够，仅在下游音频任务上进行微调大型语言模型（LLM） [10 , 63, 9]；3）无法访问源代码和检查点，社区价值有限 [29, 7]。

在本报告中，我们介绍了Kimi-Audio，一款开源的音频基础模型，能够处理各种音频处理任务。我们详细描述了在架构、数据和训练三个方面构建最先进（SOTA）音频基础模型的努力。

- 架构。我们的模型由三个部分组成：一个音频标记器和解标记器作为音频输入输出，以及一个音频大模型作为核心处理部分（见第2.1节）。我们使用离散的语义音频标记作为音频大模型输入和输出的基本表示。同时，我们在输入中将语义音频标记与连续的声学向量连接，以增强感知能力；在输出中将其与离散的文本标记连接，以增强生成能力。通过这种方式，我们可以同时实现良好的音频感知和生成能力，促进通用音频建模。我们将每秒的标记数减少，以弥合文本和音频序列之间的差距，并将语义和声学音频标记的压缩率设为12.5Hz。第2.2节和第2.3节分别介绍了离散语义标记和连续声学向量的音频标记器的详细设计，以及离散语义标记和文本标记的生成方法。

- 数据。为了实现最先进的通用音频建模，我们需要在大量音频数据上进行预训练，以涵盖多样的场景。为此，我们爬取并处理了一个大规模的音频预训练数据集。我们开发了一个由语音增强、说话人识别、转录、过滤等组成的数据处理流程，以确保高数据质量（见第3.1节）。为了支持多样的音频处理任务，我们整理了大量针对特定任务的监督微调（SFT）数据。我们展示了一种经济高效的方法，利用纯开源和可访问的数据源及处理工具，构建大部分SFT数据，从而实现最先进的性能，而无需依赖任何数据购买（见第3.2节）。

- 训练。为了在保持高知识容量和智能水平的同时实现良好的音频理解/生成能力，我们用预训练的LLM初始化音频大模型，并精心设计一系列预训练任务，以充分学习音频数据并弥合差距{v*} between text and audio. Specifically, the pre-training tasks can be divided into three categories: 1) text-only and audio-only pre-training, which aims to learn the knowledge from text and audio domains separately; 2) audio-to-text mapping, which encourages the conversion between audio and text; 3) audio-text interleaving, which further bridge the gap between text and audio (see Section 4.1). In the supervised fine-tuning stage, we develop a training recipe to improve fine-tuning efficiency and task generalization (see Section 4.2).

Furthermore, we introduce the practices for deploying and serving our audio foundation model for inference in Kimi APP, as described in Section 5. Evaluating and benchmarking an audio foundation model in various downstream tasks such as speech recognition, audio understanding, and speech conversation is challenging. We encounter tricky issues in fairly comparing different audio models, such as non-standardized metric, evaluation protocol, and inference hyper-parameters. Therefore, we develop an evaluation toolkit that can faithfully evaluate audio LLMs on comprehensive benchmarks (see Section 6.1). We open-source this toolkit to facilitate a fair comparison in the community.

Based on our evaluation toolkit, we conduct a comprehensive evaluation of Kimi-Audio and other audio LLMs on a variety of audio benchmarks (see Section 6.2). Evaluation results demonstrate that Kimi-Audio achieves SOTA performance in a series of audio tasks, including speech recognition, audio understanding, audio-to-text chat, and speech conversation. We open source the codes and checkpoints of Kimi-Audio, as well as the evaluation toolkit in `https://github.com/MoonshotAI/Kimi-Audio`, to boost the development of the community.

2 Architecture

2.1 Overview

Kimi-Audio is an audio foundation model designed to perform comprehensive audio understanding, generation, and conversation tasks within a unified architecture. As illustrated in Figure 2, our system comprises three primary components: (1) an audio tokenizer that converts input audio into discrete semantic tokens derived through vector quantization with a 12.5Hz frame rate. The audio tokenizer additionally extracts continuous acoustic vectors to enhance perception capability. (2) an audio LLM that generates semantic tokens together with text token to improve the generation capability, featuring shared transformer layers that process multimodal inputs before branching into specialized parallel heads for text and audio generation; and (3) an audio detokenizer that converts the discrete semantic tokens predicted by the audio LLM back into coherent audio waveforms using a flow matching approach. This integrated architecture enables Kimi-Audio to seamlessly handle diverse audio-language tasks from speech recognition and understanding to speech conversation within a single unified model framework.

2.2 Audio Tokenizer

Our audio foundation model employs a hybrid audio tokenization strategy, integrating discrete semantic tokens and complementary continuous vectors of acoustic information to effectively represent speech signals for downstream tasks. This tokenization allows the model to leverage the efficiency and semantic focus of discrete tokens while benefiting from the rich acoustic details captured by continuous representations.

We incorporate the discrete semantic tokens proposed by GLM-4-Voice [84]. This component utilizes a supervised speech tokenizer derived from an automatic speech recognition (ASR) model.

在文本和音频之间。具体而言，预训练任务可以分为三类：1）仅文本和仅音频预训练，旨在分别从文本和音频领域学习知识；2）音频到文本的映射，鼓励音频与文本之间的转换；3）音频-文本交错，进一步弥合文本和音频之间的差距（见第4.1节）。在有监督的微调阶段，我们制定了一套训练方案，以提高微调效率和任务的泛化能力（见第4.2节）。

此外，我们介绍了在Kimi APP中部署和服务我们的音频基础模型以进行推理的实践，如第5节所述。在各种下游任务中评估和基准测试音频基础模型，例如语音识别、音频理解和语音对话，是具有挑战性的。我们在公平比较不同音频模型时遇到一些棘手的问题，例如非标准化的指标、评估协议和推理超参数。因此，我们开发了一个评估工具包，能够在全面的基准测试中忠实地评估音频LLMs（见第6.1节）。我们将此工具包开源，以促进社区中的公平比较。

基于我们的评估工具包，我们对Kimi-Audio及其他音频大模型在各种音频基准测试中进行了全面评估（见第6.2节）。评估结果显示，Kimi-Audio在一系列音频任务中实现了最先进的性能，包括语音识别、音频理解、音频转文本聊天和语音对话。我们在https://github.com/MoonshotAI/Kimi-Audio开源了Kimi-Audio的代码和检查点，以及评估工具包，以促进社区的发展。

2 架构

2.1 概述

Kimi-Audio 是一种音频基础模型，旨在在统一架构中执行全面的音频理解、生成和对话任务。如图 2 所示，我们的系统由三个主要组件组成：(1) 音频分词器，将输入音频转换为通过向量量化获得的离散语义标记，采样率为 12.5Hz。音频分词器还额外提取连续声学向量，以增强感知能力；(2) 音频大模型（LLM），与文本标记一起生成语义标记，以提升生成能力，具有共享的变换器层，处理多模态输入后分支为专门的平行头，用于文本和音频生成；以及 (3) 音频去标记器，使用流匹配方法将由音频 LLM 预测的离散语义标记转换回连贯的音频波形。这一集成架构使 Kimi-Audio 能够在单一的统一模型框架内无缝处理从语音识别和理解到语音对话的多样音频语言任务。

2.2 音频标记器

我们的音频基础模型采用混合音频标记策略，结合离散语义标记和互补的连续声学信息向量，有效地表示语音信号以用于下游任务。这种标记方式使模型能够利用离散标记的高效性和语义焦点，同时受益于连续表示所捕获的丰富声学细节。

我们结合了由GLM-4-Voice [84]提出的离散语义标记。该组件采用由自动语音识别（ASR）模型衍生的有监督语音分词器。

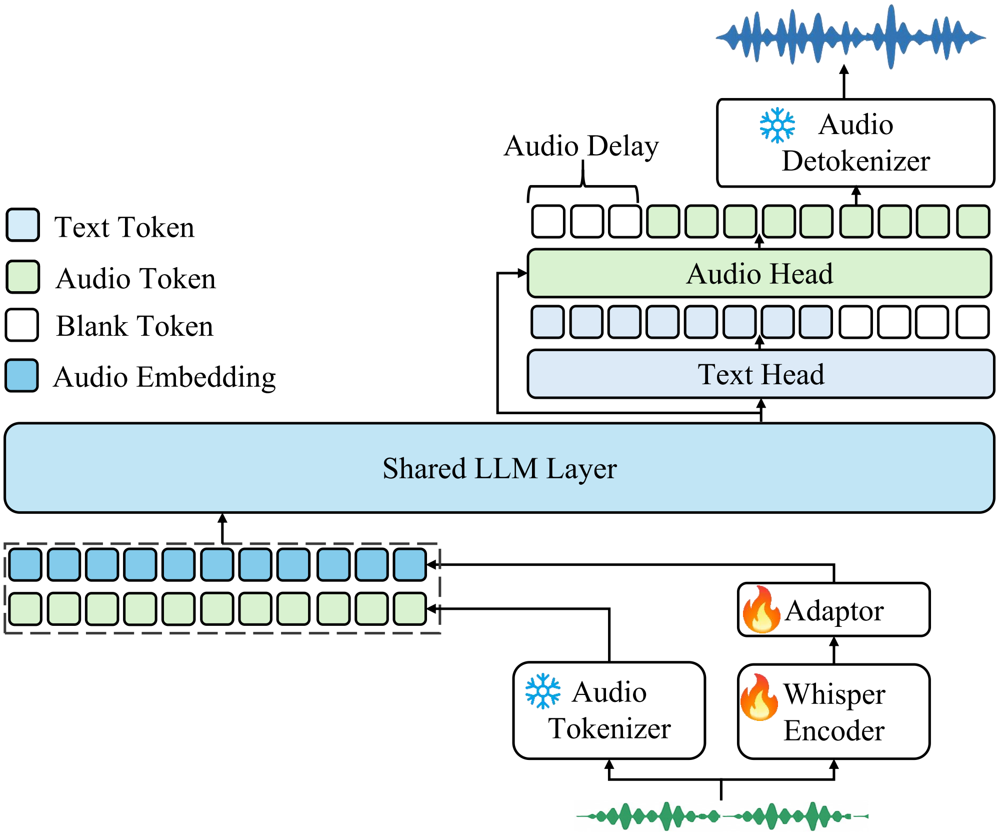

Figure 2: Overview of the Kimi-Audio model architecture: (1) an audio tokenizer that extracts discrete semantic tokens and a Whisper encoder that generates continuous acoustic features; (2) an audio LLM that processes audio inputs and generates text and/or audio outputs; (3) an audio detokenizer converts audio tokens into waveforms.

By introducing a vector quantization layer within the whisper encoder architecture [58], we can transform continuous speech representations into a sequence of discrete tokens at a low frame rate (i.e. 12.5Hz) using a single codebook.

Complementing the discrete semantic tokens, we incorporate a continuous feature representation derived from a pre-trained whisper model [58] to enhance the perception capability of our model. Since the whisper feature has a frame rate of 50Hz, we additionally introduce an adaptor upon the whisper feature extractor to downsample the feature from 50Hz to 12.5Hz. The downsampled features are added to the embeddings of discrete semantic tokens to serve as the input of the audio LLM.

By combining discrete semantic tokens with continuous whisper features, our model benefits from efficient, semantically grounded representation and detailed acoustic modeling, providing a comprehensive foundation for diverse audio processing tasks.

2.3 Audio LLM

The core of our system is an audio LLM designed to process the audio representations generated by the tokenization strategy described in Section 2.2 and produce multimodal outputs, which include the discrete semantic tokens of audio and the corresponding text tokens to improve the generation capability.

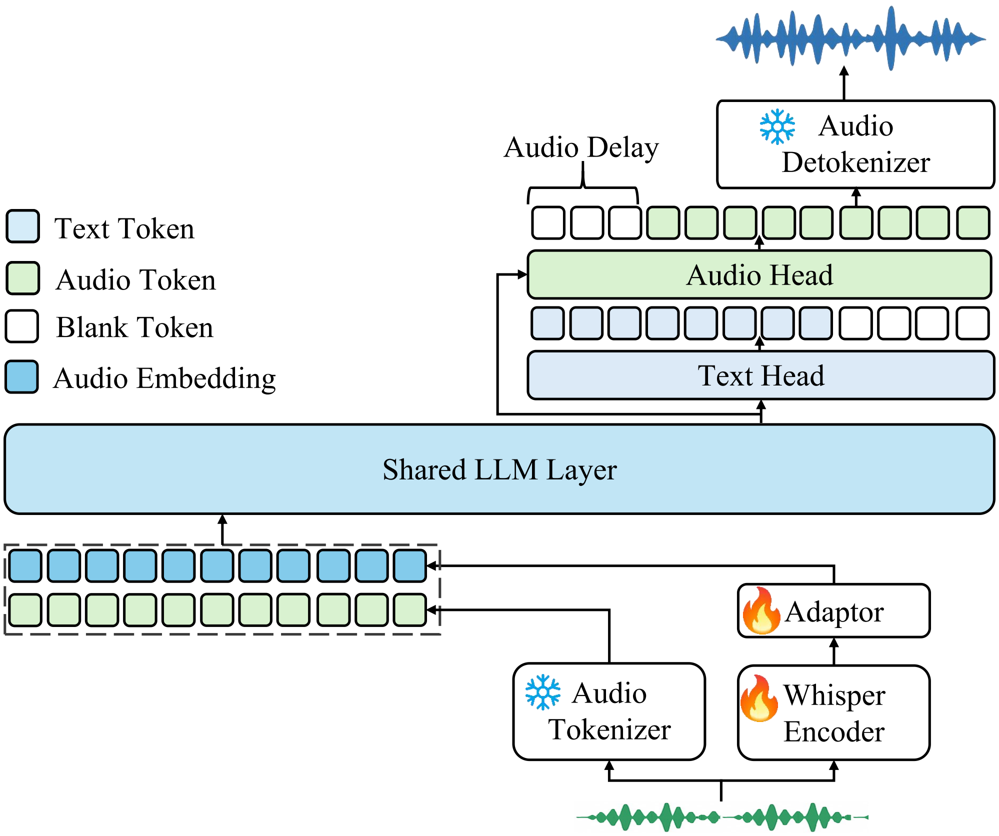

图2：Kimi-Audio模型架构概述：(1) 一个音频分词器，用于提取离散的语义标记，以及一个Whisper编码器，用于生成连续的声学特征；(2) 一个音频大模型（LLM），用于处理音频输入并生成文本和/或音频输出；(3) 一个音频去标记器，将音频标记转换为波形。

通过在 whisper 编码器架构中引入向量量化层 [58]，我们可以使用单一的码本将连续的语音表示转换为低帧率（即 12.5Hz）的一系列离散标记。

补充离散语义标记，我们还引入了一个来自预训练的 whisper 模型 [58] 的连续特征表示，以增强模型的感知能力。由于 whisper 特征的帧率为 50Hz，我们还在 whisper 特征提取器上添加了一个适配器，将特征从 50Hz 降采样到 12.5Hz。降采样后的特征被加入到离散语义标记的嵌入中，作为音频 LLM 的输入。

通过将离散语义标记与连续的 whisper 特征相结合，我们的模型受益于高效、语义基础的表示和详细的声学建模，为各种音频处理任务提供了全面的基础。

2.3 音频大模型

我们系统的核心是一个音频大模型，旨在处理由第2.2节描述的标记策略生成的音频表示，并产生多模态输出，包括音频的离散语义标记和相应的文本标记，以提高生成能力。

To enable the model to generate both audio semantic tokens and the corresponding textual responses, we adapt the standard LLM architecture by structuring it into components with shared and specialized functions. A significant portion of the original transformer bottom layers, i.e., the first several layers, are utilized as shared layers. These layers process the input sequence and learn cross-modal representations, integrating information from both text and audio modalities present in the input or context. Based on these shared layers, the architecture diverges into two parallel heads containing transformer layers. The first head is a text head that is specifically responsible for autoregressively predicting text tokens, forming the textual output of the model. The second head is an audio head to predict the discrete audio semantic tokens. These predicted audio tokens are subsequently passed to an audio detokenizer module to synthesize the final output audio waveform.

To take advantage of the strong language capabilities of the pre-trained text LLMs [76, 24, 13], the parameters of the shared transformer layers and the text head are initialized directly from the weights of the pre-trained text LLM. The audio head layers are initialized randomly. This initialization strategy ensures that the model retains robust text understanding and generation capabilities while learning to effectively process and generate audio information.

2.4 Audio Detokenizer

The audio detokenizer aims to generate high-quality and expressive speech conditioned on discrete semantic audio tokens. We employ the same detokenizer architecture as in MoonCast [32], which contains two parts: 1) a flow-matching module which converts 12.5Hz semantic tokens to 50Hz mel-spectrograms; 2) a vocoder which generates waveforms from mel-spectrograms. To reduce speech generation latency, we design a chunk-wise streaming detokenizer. Intuitively, we can split the semantic tokens into chunks and decode them separately, which, however, in our preliminary experiments, faces an intermittent issue in the chunk boundaries. Thus, we propose a chunk-wise autoregressive streaming framework with a look-ahead mechanism.

Chunk-wise Autoregressive Streaming Framework. We split the audio into chunks (e.g., 1 second per chunk): $\{c_1, c_2, ..., c_i, ..., c_N\}$, where N is the number of chunks. Firstly, to match the sequence length between semantic tokens (12.5Hz) and mel-spectrograms (50Hz), we upsample the semantic tokens by 4x rate. Secondly, we apply a chunk-wise causal mask during training and inference, i.e., for chunk c_i, all previous chunks c_j with $j < i$ are prompts. We denote chunk c_i's mel-spectrograms as m_i and the corresponding discrete semantic audio tokens as a_i^d. The flow-matching model's forward step will mix m_i with Gaussian noise and the backward step will remove noise to obtain clean m_i with condition a_i^d and prompt c_j, where $j < i$, and c_j contains both m_j and a_j^d. With this design, during inference, when the LLM generates a chunk, we employ the flow-matching model to detokenize it to obtain the mel-spectrograms. Finally, we apply a BigVGAN [38] vocoder to generate wavforms for each chunk.

Look-Ahead Mechanism. With a preliminary study, we find that the generated audio in the boundaries of chunks still has an intermittent issue. Although a long range of history context has been seen during the diffusion denoising process, the future context of the boundary position cannot be seen due to the nature of block-wise causal attention, which causes the degradation of quality. Thus, we propose a look-ahead mechanism. In detail, for chunk c_i, we take the future n (e.g. 4) semantic tokens from chunk c_{i+1} and concatenate them to the end of c_i to form $\hat{c}_i$. Then we detokenize $\hat{c}_i$ to generate the mel-spectrograms, but only retain the mel-spectrograms corresponding to c_i. This mechanism is training-free and will only delay the generation of the first chunk by n tokens.

为了使模型能够同时生成音频语义标记和相应的文本响应，我们对标准的大型语言模型（LLM）架构进行了调整，将其结构化为具有共享和专用功能的组件。原始变换器底层的很大一部分，即前几层，被用作共享层。这些层处理输入序列并学习跨模态表示，整合输入或上下文中存在的文本和音频模态的信息。在这些共享层的基础上，架构分化为两个平行的头部，包含变换器层。第一个头部是文本头，专门负责自回归地预测文本标记，形成模型的文本输出。第二个头部是音频头，用于预测离散的音频语义标记。这些预测的音频标记随后传递给音频去标记模块，以合成最终的输出音频波形。

为了充分利用预训练文本大模型[76, 24, 13]强大的语言能力，共享变换器层和文本头的参数直接从预训练文本大模型的权重初始化。音频头层则随机初始化。这种初始化策略确保模型在学习有效处理和生成音频信息的同时，保持强大的文本理解和生成能力。

2.4 音频去标记器

音频解标器旨在生成高质量且富有表现力的语音，条件是离散的语义音频标记。我们采用与MoonCast [32]相同的解标器架构，包括两个部分：1) 一个流匹配模块，将12.5Hz的语义标记转换为50Hz的梅尔频谱图；2) 一个声码器，从梅尔频谱图生成波形。为了减少语音生成的延迟，我们设计了逐块流式解标器。直观上，我们可以将语义标记分成块并分别解码，但在我们的初步实验中，这在块边界处会遇到间歇性的问题。因此，我们提出了一种带有前瞻机制的逐块自回归流式框架。

块级自回归流式框架。我们将音频分成块（例如，每块1秒）：$\{c_1, c_2, ..., c_i, ..., c_N\}$，其中$N$是块的数量。首先，为了匹配语义标记（12.5Hz）和梅尔频谱图（50Hz）之间的序列长度，我们将语义标记上采样4倍。其次，在训练和推理过程中，我们应用块级因果掩码，即对于块c_i，所有之前的块c_j，其中$j < i$作为提示。我们将块c_i的梅尔频谱图表示为m_i，对应的离散语义音频标记为a_i^d。流匹配模型的前向步骤会将m_i与高斯噪声混合，反向步骤则会去除噪声，获得带有条件a_i^d和提示c_j的干净m_i，其中$j < i$，c_j包含m_j和a_j^d。通过这种设计，在推理过程中，当LLM生成一个块时，我们使用流匹配模型将其解码为梅尔频谱图。最后，我们应用BigVGAN [38]声码器为每个块生成波形。

前瞻机制。通过初步研究，我们发现生成的音频在块的边界处仍然存在间歇性问题。虽然在扩散去噪过程中已经观察到较长的历史上下文，但由于块级因果注意力的本质，边界位置的未来上下文无法被看到，这导致了质量的下降。因此，我们提出了一种前瞻机制。具体来说，对于块 c_i，我们从块 c_{i+1} 中取出未来的 n (例如 4) 个语义标记，并将它们连接到 c_i 的末尾，形成 $\hat{c}_i$。然后我们对 $\hat{c}_i$ 进行去标记，生成梅尔频谱图，但只保留对应于 c_i 的梅尔频谱图。该机制无需训练，只会将第一个块的生成延迟 n 个标记。

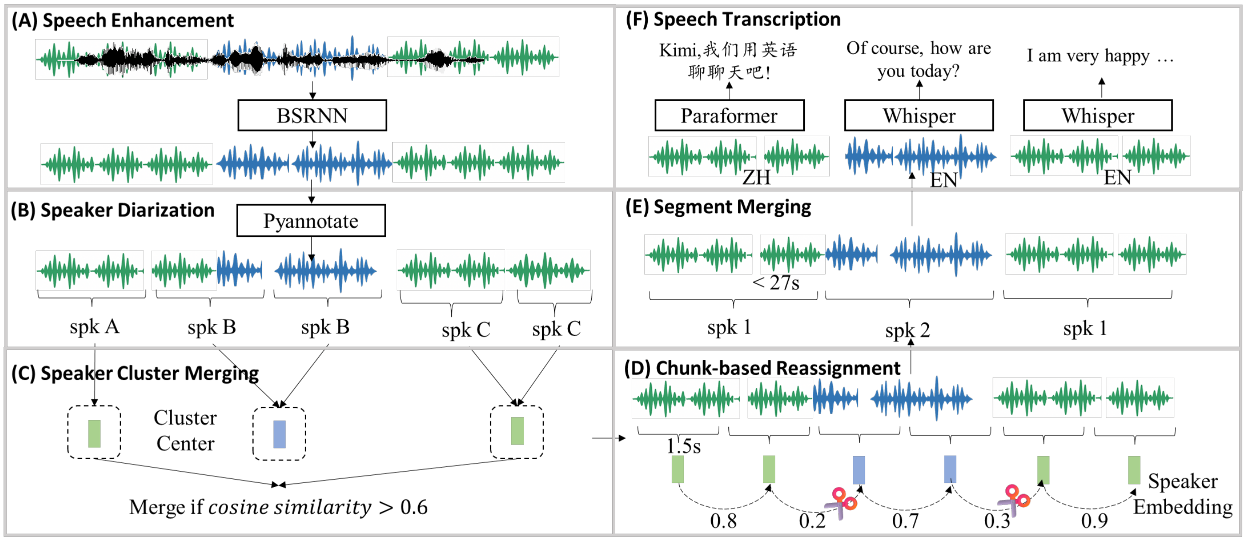

Figure 3: Processing pipeline for the audio pre-training data.

3 Data

3.1 Pre-Training Data

Our pre-training corpus comprises both unimodal (text-only, audio-only) and multimodal (text-audio) data. The audio-only pre-training data covers a wide range of real-world scenarios, including audiobooks, podcasts, and interviews, and consists of approximately 13 million hours of raw audio containing rich acoustic events, music, environmental sound, human vocalization, and multilingual information. The details of the text-only pre-training data can be found in [65].

Most audio corpus contains only raw audio without corresponding transcriptions, language types, speaker annotations, and segmentation boundaries. In addition, the raw audio often contains undesired artifacts such as background noise, reverberation, and speaker overlap.

Inspired by previous work [32, 81, 26], we develop an efficient automatic audio data processing pipeline to generate high-quality annotations, resulting in our multimodal (audio-text) data.

Compared to previous data processing pipelines that primarily focus on generating high-quality short audio segments without contextual information, our pipeline is designed to provide long-form audio annotations with consistent long-range context. The pipeline includes the following key components in a step-by-step manner, as shown in Figure 3 and described as follows.

Speech Enhancement. To suppress undesired background noise and reverberation, we develop a speech enhancement model based on the Band-Split RNN (BSRNN) architecture [49], as shown in Figure 3(A). Following the same hyper-parameter configuration as in [82], the model is applied to perform 48kHz speech enhancement. Empirically, we find that speech enhancement will remove the environmental sound and music, which can be harmful to audio understanding. Thus, we randomly choose original or enhanced audio with a ratio of $1:1$ in the pre-training stage.

Segmentation by Diarization. We employ a diarization-driven approach to segment long-form audio. We utilize the PyAnnote toolkit[1] for speaker diarization (Figure 3(B)), which segments the

[1] https://github.com/pyannote/pyannote-audio

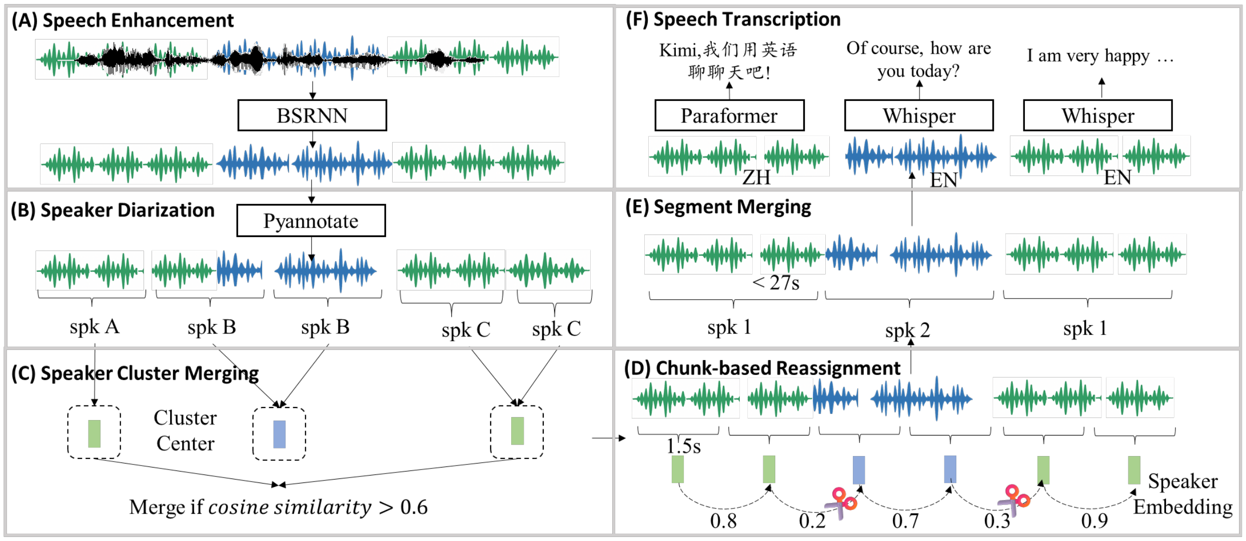

图 3：音频预训练数据的处理流程。

3 数据

3.1 预训练数据

我们的预训练语料库包括单模态（仅文本、仅音频）和多模态（文本-音频）数据。仅音频预训练数据涵盖了各种真实场景，包括有声书、播客和访谈，约包含1300万小时的原始音频，包含丰富的声学事件、音乐、环境声音、人类发声和多语言信息。仅文本预训练数据的详细信息可参见[65]。

大多数音频语料库仅包含原始音频，没有对应的转录、语言类型、说话人注释和分段边界。此外，原始音频常常包含不需要的杂音，例如背景噪声、混响和说话人重叠。

受到之前工作的启发 [32, 81, 26]，我们开发了一种高效的自动音频数据处理流程，以生成高质量的标注，从而获得我们的多模态（音频-文本）数据。

与之前主要专注于生成没有上下文信息的高质量短音频片段的数据处理流程相比，我们的流程旨在提供具有一致长程上下文的长格式音频注释。该流程按步骤包括以下关键组成部分，如图3所示，并描述如下。

语音增强。为了抑制不需要的背景噪声和混响，我们开发了一种基于带分离RNN（BSRNN）架构[49]的语音增强模型，如图3(A)所示。遵循[82]中的相同超参数配置，该模型用于进行48kHz的语音增强。经验表明，语音增强会去除环境声音和音乐，这可能对音频理解产生不利影响。因此，在预训练阶段，我们以1：1的比例随机选择原始或增强后的音频。

按说话人分段。我们采用基于说话人分离的方法对长音频进行分段。我们使用PyAnnote工具包[1]进行说话人分离（图3(B)），它将音频进行分段。

[1] https://github.com/pyannote/pyannote-audio

audio and assigns speaker labels. However, the raw output is sub-optimal, and thus we develop a post-processing pipeline to address the issues in the previous segmentation results:

- **Speaker Cluster Merging.** We observe that PyAnnote sometimes assigns multiple speaker labels to the same actual speaker, which results in speaker fragmentation. We compute representative speaker embeddings for each initial cluster and merge pairs of clusters whose embeddings have a cosine similarity greater than 0.6, as shown in Figure 3(C).
- **Chunk-based Reassignment.** The initial diarization occasionally produced segments containing multiple speakers. To purify the segments, 1) we first divide all segments into 1.5-second chunks, and then 2) for each pair of adjacent chunks, if their cosine similarity is below 0.5, we treat them as belonging to different speakers and reassign each chunk to the speaker cluster with the highest similarity, as shown in Figure 3(D).
- **Segment Merging.** The initial diarization could result in segments of highly variable and sometimes impractical lengths (shorter than 1s or longer than 100s). So we iteratively merge adjacent segments labeled with the same speaker (after the reassignment step). The merging process terminates if the accumulated segment length exceeds 27 seconds or the silence gap between two segments is greater than 2 seconds, as shown in Figure 3(E).

The resulting segmentation from this refined diarization process provides more accurate and consistently sized speaker turns compared to the baseline diarization output.

Speech Transcription. To obtain the language type and text transcription for each speech segment, we first apply the Whisper-large-v3 model [58][2] to detect the spoken language type. In this work, we retain only English and Mandarin segments for further transcription. For English segments, we directly use Whisper-large-v3 to generate both transcriptions and punctuation annotations. For Mandarin segments, we utilize the Paraformer-Zh [20] model from the FunASR toolkit[3] to generate transcriptions along with character-level timestamps. Since Paraformer-Zh cannot output punctuation annotations, we add punctuation annotations with the following strategy: if the time gap between two consecutive characters is greater than 0.5 seconds but less than 1.0 second, we insert a "comma"; if the gap exceeds 1.0 second, we insert a "period".

Implementation. The data processing pipeline is deployed on a cluster of 30 cloud instances. Each instance is equipped with 128 virtual CPUs (vCores), 1 TB of RAM, and 8 NVIDIA L20 GPUs, powered by Intel Xeon Platinum 8575C processors that support vectorized acceleration instructions, including Advanced Matrix Extensions (AMX). In total, the cluster provides $3,840$ vCores, 30 TB of memory, and 240 NVIDIA L20 GPUs. Following extensive optimization, the pipeline achieves a daily processing throughput of approximately $200,000$ hours of raw audio data.

3.2 SFT Data

After the pre-training stage, we perform supervised fine-tuning (SFT) to enhance the performance of Kimi-Audio on instruction following and audio processing. The SFT data can be mainly categorized into three parts: audio understanding, speech conversation, and audio-to-text chat.

3.2.1 Audio Understanding

[2] https://huggingface.co/openai/whisper-large-v3

[3] https://github.com/modelscope/FunASR

[1] https://github.com/fighting41love/zhvoice

音频并分配说话人标签。然而，原始输出效果不佳，因此我们开发了一个后处理流程，以解决之前分段结果中的问题：

- 说话人簇合并。我们观察到PyAnnote有时会为同一实际说话人分配多个说话人标签，导致说话人碎片化。我们为每个初始簇计算代表性说话人嵌入，并合并余弦相似度大于0.6的簇对，如图3(C)所示。
- 基于块的重新分配。最初的说话人识别有时会产生包含多个说话人的片段。为了纯化这些片段，1）我们首先将所有片段划分为1.5秒的块，然后2）对于每对相邻的块，如果它们的余弦相似度低于0.5，我们将它们视为属于不同的说话人，并将每个块重新分配到具有最高相似度的说话人簇，如图3（D）所示。
- 片段合并。最初的说话人识别可能会产生长度变化很大且有时不太实用的片段（短于1秒或长于100秒）。因此，我们在重新分配步骤后，迭代合并标记为同一说话人的相邻片段。合并过程在以下任一条件满足时终止：累计片段长度超过27秒，或两个片段之间的静音间隙大于2秒，如图3(E)所示。

经过此改进的说话人识别过程所得的分割，比基线说话人识别输出提供了更准确且尺寸更一致的说话人轮次。

语音转录。为了获得每个语音片段的语言类型和文本转录，我们首先应用 Whisper-large-v3 模型 [58][2] 来检测所说的语言类型。在本工作中，我们仅保留英语和普通话片段以供进一步转录。对于英语片段，我们直接使用 Whisper-large-v3 生成转录文本和标点注释。对于普通话片段，我们利用 FunASR 工具包中的 Paraformer-Zh [20][3] 模型，生成带有字符级时间戳的转录文本。由于 Paraformer-Zh 不能输出标点注释，我们采用以下策略添加标点：如果两个连续字符之间的时间间隔大于 0.5 秒但少于 1.0 秒，则插入“逗号”；如果间隔超过 1.0 秒，则插入“句号”。

实现。数据处理管道部署在一个由30个云实例组成的集群上。每个实例配备128个虚拟CPU（vCores）、1 TB内存和8个NVIDIA L20 GPU，由支持向量化加速指令的Intel Xeon Platinum 8575C处理器驱动，包括高级矩阵扩展（AMX）。整个集群提供3, 840个vCores、30 TB内存和240个NVIDIA L20 GPU。经过广泛优化后，该管道实现了每日大约200, 000小时的原始音频数据处理吞吐量。

3.2 SFT 数据

在预训练阶段之后，我们进行有监督微调（SFT），以提升Kimi-Audio在指令执行和音频处理方面的性能。SFT数据主要可以分为三部分：音频理解、语音对话和音频转文本聊天。

3.2.1 音频理解

[2] https://huggingface.co/openai/whisper-large-v3

[3] https://github.com/modelscope/FunASR

[1] https://github.com/fighting41love/zhvoice

Table 1: List of datasets used for audio understanding and their training epoch in SFT stage.

Dataset	Audio Length (#hours)	Task Type	SFT Epochs
WenetSpeech [85]	10,518	ASR	2.0
WenetSpeech4TTS [50]	12,085	ASR	2.0
AISHELL-1 [4]	155	ASR	2.0
AISHELL-2 [17]	1,036	ASR	2.0
AISHELL-3 [62]	65	ASR	2.0
Emilla [25]	98,305	ASR	2.0
Fleurs [12]	17	ASR	2.0
CommonVoice [1]	43	ASR	2.0
KeSpeech [64]	1,428	ASR	2.0
Magicdata [79]	747	ASR	2.0
zhvoice[1]	901	ASR	2.0
Libriheavy [33]	51,448	ASR	2.0
MLS [57]	45,042	ASR	2.0
Gigaspeech [5]	10,288	ASR	2.0
LibriSpeech [54]	960	ASR	2.0
CommonVoice [1]	1,854	ASR	2.0
Voxpopuli [69]	529	ASR	2.0
LibriTTS [83]	568	ASR	2.0
CompA-R [22]	159	AQA	2.0
ClothoAQA [43]	7.4	AQA	4.0
AudioCaps [34]	137	AAC	2.0
Clotho-v2 [16]	24.0	AAC	2.0
MACS [51]	10.9	AAC	2.0
FSD50k [19]	80.8	SEC	2.0
CochlScene [31]	169.0	ASC	2.0
Nonspeech7k [59]	6.2	SEC	4.0
MusicAVQA$_{\text{audio-only}}$ [39]	77.1	AQA	2.0
WavCaps [52]	3,793.3	AAC	2.0
AVQA$_{\text{audio-only}}$ [78]	112	AQA	2.0
IEMOCAP [68]	10	SER	2.0
MELD [56]	9	SER	2.0
RAVDESS [47]	3	SER	2.0
SAVEE [30]	0.1	SER	2.0
ESD [89]	29	SER	2.0
TUT2016 [53]	10	ASC	2.0
TUT2017 [53]	13	ASC	4.0
TAU2022 [27]	67	ASC	2.0
ESC50 [55]	1	SEC	2.0
VocalSound [23]	19	SEC	4.0
VGGSound [6]	513	SEC	2.0
UrbanSound8K [61]	9	SEC	2.0
FSD50K [19]	74	SEC	2.0
Kimi Inhouse ASR Data	55,000	ASR	2.0
Kimi Inhouse Audio Data	5,200	AAC/AQA	2.0

We mainly leverage open-source datasets for audio understanding. The collected datasets include 6 tasks: Automatic Speech Recognition (ASR), Audio Question Answer (AQA), Automated Audio Caption (AAC), Speech Emotion Recognition (SER), Sound Event Classification (SEC), and Audio Scene Classification (ASC). The details of the datasets and the corresponding training epochs in the SFT stage are shown in Table 1. Besides the open-source datasets, we also utilize 55,000 hours in-house ASR data and 5,200 hours in-house audio data covering the AAC/AQA tasks.

表1：用于音频理解的数据集列表及其在SFT阶段的训练轮数。

Dataset	Audio Length (#hours)	Task Type	SFT Epochs
WenetSpeech [85]	10, 518	ASR	2.0
WenetSpeech4TTS [50]	12, 085	ASR	2.0
AISHELL-1 [4]	155	ASR	2.0
AISHELL-2 [17]	1, 036	ASR	2.0
AISHELL-3 [62]	65	ASR	2.0
Emilla [25]	98, 305	ASR	2.0
Fleurs [12]	17	ASR	2.0
CommonVoice [1]	43	ASR	2.0
KeSpeech [64]	1, 428	ASR	2.0
Magicdata [79]	747	ASR	2.0
zhvoice[1]	901	ASR	2.0
Libriheavy [33]	51, 448	ASR	2.0
MLS [57]	45, 042	ASR	2.0
Gigaspeech [5]	10, 288	ASR	2.0
LibriSpeech [54]	960	ASR	2.0
CommonVoice [1]	1, 854	ASR	2.0
Voxpopuli [69]	529	ASR	2.0
LibriTTS [83]	568	ASR	2.0
CompA-R [22]	159	AQA	2.0
ClothoAQA [43]	7.4	AQA	4.0
AudioCaps [34]	137	AAC	2.0
Clotho-v2 [16]	24.0	AAC	2.0
MACS [51]	10.9	AAC	2.0
FSD50k [19]	80.8	SEC	2.0
CochlScene [31]	169.0	ASC	2.0
Nonspeech7k [59]	6.2	SEC	4.0
MusicAVQA$_{\text{audio-only}}$ [39]	77.1	AQA	2.0
WavCaps [52]	3, 793.3	AAC	2.0
AVQA$_{\text{audio-only}}$ [78]	112	AQA	2.0
IEMOCAP [68]	10	SER	2.0
MELD [56]	9	SER	2.0
RAVDESS [47]	3	SER	2.0
SAVEE [30]	0.1	SER	2.0
ESD [89]	29	SER	2.0
TUT2016 [53]	10	ASC	2.0
TUT2017 [53]	13	ASC	4.0
TAU2022 [27]	67	ASC	2.0
ESC50 [55]	1	SEC	2.0
VocalSound [23]	19	SEC	4.0
VGGSound [6]	513	SEC	2.0
UrbanSound8K [61]	9	SEC	2.0
FSD50K [19]	74	SEC	2.0
Kimi Inhouse ASR Data	55, 000	ASR	2.0
Kimi Inhouse Audio Data	5, 200	AAC/AQA	2.0

我们主要利用开源数据集进行音频理解。收集的数据集包括6个任务：自动语音识别（ASR）、音频问答（AQA）、自动音频字幕（AAC）、语音情感识别（SER）、声音事件分类（SEC）和音频场景分类（ASC）。数据集的详细信息及在SFT阶段对应的训练轮数如表1所示。除了开源数据集外，我们还利用了55, 000小时的内部ASR数据和5, 200小时的内部音频数据，涵盖AAC/AQA任务。

3.2.2 Speech Conversation

To activate the Kimi-Audio model's ability to generate speech with diverse styles and high expressiveness in different conversation scenarios, we construct a large volume of speech conversation data, which consists of multi-turn conversations made up with a series of user queries and assistant responses. For user queries, we instruct LLMs to write the text of user queries and then convert them into speech with our Kimi-TTS system, where the prompt speech is randomly selected from a large timbre set containing more than 125K timbres. For the assistant responses, we first select a voice actor as our Kimi-Audio speaker and synthesize the assistant responses with appropriate style and emotion with this single timbre. In the following, we introduce the data recording process for the Kimi-Audio speaker, as well as the Kimi-TTS and Kimi-VC systems used to synthesize assistant responses with diverse styles and expressiveness.

Data Recording for Kimi-Audio Speaker. To achieve diverse and highly expressive styles and emotions in the generated speech, we select a voice actor as the Kimi-Audio speaker and meticulously record a dataset of this speaker in a professional recording studio. We pre-define over 20 styles and emotions for recording, with each emotion further divided into 5 levels to represent varying emotional intensities. For each style and emotional level, we record an audio as the reference to maintain the consistency of the emotion and style among different text sentences. The whole recording process is guided by a professional recording director.

Kimi-TTS. We develop a zero-shot text-to-speech synthesis (TTS) system, called Kimi-TTS, to generate speech with only a 3-second prompt, while preserving the timbre, emotion, and style of the prompt speech. With the help of Kimi-TTS, we can synthesize speech for 1) the query text in diverse speakers/timbres with a large timbre set; 2) the response text with the styles and emotions recorded by the Kimi-Audio speaker, a voice actor selected by Kimi. Similar to the architecture of MoonCast [32], Kimi-TTS employs an LLM to generate speech tokens given the prompt speech and input text. Then a flow-matching-based speech detokenizer is used to generate high-quality speech waveforms. We train Kimi-TTS on about 1M hours generated by the automatic data pipeline (Section 3.1) and apply reinforcement learning to further enhance the robustness and quality of the generated speech.

Kimi-VC. Since it is difficult for the voice actor to record speech in any styles, emotions, and accents, we develop a voice conversion (VC) system, called Kimi-VC, to convert diverse and in-the-wild speech in different speakers/timbres into the timbre of Kimi-Audio speaker while preserving the styles, emotions, and accents. Built on the Seed-VC framework [46], Kimi-VC incorporates source timbre perturbation via a timbre-shifting model during training, which mitigates information leakage and ensures alignment between training and inference phases. To ensure high quality of voice conversion, we fine-tune the Kimi-VC model using speech data recorded by the Kimi-Audio speaker, a voice actor selected by Kimi.

3.2.3 Audio-to-Text Chat

To help Kimi-Audio with the basic ability on chat, we collect open-source supervised fine-tuning data from text domain, as listed in Table 2, and then convert the user queries to speech with a variety of timbres, resulting in the audio-to-text chat data whose user query is speech while the assistant response is text. Considering that some text cannot be easily converted to speech, we perform several preprocessing steps on text by 1) filtering out text containing complex math, code, table, complex multilingual content, or too long content, 2) making colloquial rewriting, and 3) convert

3.2.2 语音对话

为了激活 Kimi-Audio 模型在不同对话场景中生成具有多样风格和高度表现力的语音的能力，我们构建了大量的语音对话数据，该数据由多轮对话组成，包括一系列用户查询和助手响应。对于用户查询，我们指导大型语言模型（LLMs）撰写用户查询文本，然后使用我们的 Kimi-TTS 系统将其转换为语音，其中提示语音从包含超过 125K 种声色的大型声色集随机选择。对于助手响应，我们首先选择一位配音演员作为我们的 Kimi-Audio 说话人，并用具有适当风格和情感的单一声色合成助手响应。接下来，我们介绍用于 Kimi-Audio 说话人数据录制的过程，以及用于合成具有多样风格和表现力的助手响应的 Kimi-TTS 和 Kimi-VC 系统。

Kimi-Audio扬声器的数据录制。为了在生成的语音中实现多样且高度富有表现力的风格和情感，我们选择一位配音演员作为Kimi-Audio扬声器，并在专业录音棚中为该扬声器仔细录制一个数据集。我们预定义了超过20种风格和情感进行录制，每种情感进一步划分为5个等级，以表现不同的情感强度。对于每种风格和情感等级，我们都会录制一段音频作为参考，以保持不同文本句子之间情感和风格的一致性。整个录制过程由专业的录音导演指导。

Kimi-TTS。我们开发了一种零样本文本到语音合成（TTS）系统，称为Kimi-TTS，能够仅凭3秒的提示生成语音，同时保持提示语的音色、情感和风格。在Kimi-TTS的帮助下，我们可以合成以下内容的语音：1）多样的说话人/音色的查询文本，具有大量的音色集；2）由Kimi-Audio说话人（由Kimi选择的配音演员）录制的风格和情感的应答文本。类似于MoonCast [32]的架构，Kimi-TTS采用大规模语言模型（LLM）在给定提示语和输入文本的情况下生成语音标记。然后，使用基于流匹配的语音解码器将这些标记转换为高质量的语音波形。我们在大约1百万小时由自动数据管道（第3.1节）生成的数据上训练Kimi-TTS，并应用强化学习以进一步提升生成语音的鲁棒性和质量。

Kimi-VC。由于配音演员在录制各种风格、情感和口音的语音方面存在困难，我们开发了一种叫做 Kimi-VC 的语音转换（VC）系统，用于将不同说话人/音色的多样化和自然语音转换为 Kimi-Audio 说话人的音色，同时保持风格、情感和口音。基于 Seed-VC 框架 [46]，Kimi-VC 在训练过程中引入了通过音色迁移模型进行的源音色扰动，这减轻了信息泄露问题，并确保训练和推理阶段的一致性。为了确保语音转换的高质量，我们使用由 Kimi 选择的配音演员 Kimi-Audio 说话人录制的语音数据对 Kimi-VC 模型进行了微调。

3.2.3 音频转文本聊天

为了帮助 Kimi-Audio 具备基本的聊天能力，我们从文本领域收集了开源的有监督微调数据，如表 2 所示，然后将用户查询转换为具有多种音色的语音，从而生成了音频到文本的聊天数据，其中用户查询为语音，而助手的回复为文本。考虑到一些文本难以直接转换为语音，我们对文本进行了若干预处理步骤：1）过滤掉包含复杂数学、代码、表格、复杂多语言内容或过长内容的文本；2）进行口语化改写；3）转换为

Table 2: List of text dataset used in audio-to-text chat with their training epochs in SFT stage.

Dataset	#Samples	SFT Epochs
Magpie-Pro [75]	300K	2.0
Magpie-MT [75]	300K	2.0
Evol-Instruct [15]	143K	2.0
Evol-Instruct-Code [15]	80K	2.0
Infinity-Instruct [2]	7M	2.0
Synthia [67]	119K	2.0
NuminaMath [40]	860K	2.0
Tulu3 [36]	900K	2.0
OpenHermes-2.5 [66]	1M	2.0
OpenOrca [42]	2M	2.0

a single-turn question-answer data with complex instruction into multi-turn data with easy and concise instructions.

4 Training

4.1 Pre-training

The pre-training stage of Kimi-Audio aims to learn the knowledge from both the real-world audio and text domains and align them in the model's latent space, thereby facilitating complex tasks such as audio understanding, audio-to-text chat, and speech conversation. To this end, we design several pre-training tasks with the following aspects: 1) pre-training in the unimodality (i.e., audio and text) to learn the knowledge from each domain individually in Section 4.1.1; 2) learning audio-text mapping in Section 4.1.2; 3) three audio-text interleaving tasks to further bridge two modalities in Section 4.1.3.

Formally, given a raw audio A, the data pre-processing pipeline (described in Section 3.1) splits it into a series of segments $\{S_1, S_2, ..., S_N\}$, and each segment $S_i, i \in [1, N]$ consists of an audio a_i and the corresponding transcription t_i. Furthermore, as illustrated in Section 2.2, for an audio segment a_i, we extract both the continuous acoustic vectors a_i^c and the discrete semantic tokens a_i^d. To comply with the design of our model architecture in Section 2 which takes discrete semantic audio tokens as the main representation of the input and output, while adding continuous acoustic audio tokens in the input and discrete text tokens in the output, we denote the training sequence as $\{a_1^c/a_1^d/t_1, a_2^c/a_2^d/t_2, ..., a_N^c/a_N^d/t_N\}$, where $a_i^c/a_i^d/t_i$ denotes the semantic audio, acoustic audio, and text sequence for segment i. We make sure that the audio and text sequences have the same lengths by appending blank tokens to the shorter sequence. The actual pre-training segments can be either one or two of $a_i^c/a_i^d/t_i$, such as a_i^d, t_i, a_i^c/a_i^d, or a_i^d/t_i. For a_i^c/a_i^d, we add the continuous vectors a_i^c and the semantic tokens a_i^d (the semantic tokens will be converted into embedding using a lookup table) to obtain the final audio feature a_i. Thus, we use a_i to represent a_i^c/a_i^d for short. For a_i^d/t_i, we add the lookup embedding of the semantic tokens and text tokens as input and generate each token with its respective head, as described in Section 2.

With this notation, we formulate the following pre-training tasks in Table 3 and introduce them as follows.

表2：在SFT阶段中用于音频转文本聊天的文本数据集列表及其训练轮数。

Dataset	#Samples	SFT Epochs
Magpie-Pro [75]	300K	2.0
Magpie-MT [75]	300K	2.0
Evol-Instruct [15]	143K	2.0
Evol-Instruct-Code [15]	80K	2.0
Infinity-Instruct [2]	7M	2.0
Synthia [67]	119K	2.0
NuminaMath [40]	860K	2.0
Tulu3 [36]	900K	2.0
OpenHermes-2.5 [66]	1M	2.0
OpenOrca [42]	2M	2.0

将具有复杂指令的单轮问答数据转换为具有简洁易懂指令的多轮数据。

4 训练

4.1 预训练

Kimi-Audio的预训练阶段旨在从真实世界的音频和文本领域学习知识，并在模型的潜在空间中对它们进行对齐，从而促进音频理解、音频转文本聊天和语音对话等复杂任务。为此，我们设计了几个预训练任务，涉及以下方面：1）在单模态（即音频和文本）中进行预训练，以在第4.1.1节中单独学习每个领域的知识；2）学习音频-文本映射，在第4.1.2节中；3）三个音频-文本交错任务，以在第4.1.3节中进一步桥接两种模态。

正式地，给定一个原始音频 A，数据预处理流程（在第3.1节中描述）将其拆分为一系列片段 $\{S_1, S_2, ..., S_N\}$，每个片段 $S_i, i \in [1, N]$ 由一个音频 a_i 和相应的转录 t_i 组成。此外，如第2.2节所示，对于一个音频片段 a_i，我们提取连续的声学向量 a_i^c 和离散的语义标记 a_i^d。为了符合第2节中我们模型架构的设计，该架构以离散的语义音频标记作为输入和输出的主要表示，同时在输入中加入连续的声学音频标记，在输出中加入离散的文本标记，我们用 $\{a_1^c/a_1^d/t_1, a_2^c/a_2^d/t_2, ..., a_N^c/a_N^d/t_N\}$ 表示训练序列，其中 $a_i^c/a_i^d/t_i$ 表示片段 i 的语义音频、声学音频和文本序列。我们确保音频和文本序列具有相同的长度，通过在较短的序列后添加空白标记。实际的预训练片段可以是 $a_i^c/a_i^d/t_i$ 中的一个或两个，例如 a_i^d、t_i、a_i^c/a_i^d 或 a_i^d/t_i。对于 a_i^c/a_i^d，我们添加连续向量 a_i^c 和语义标记 a_i^d (，语义标记将通过查找表) 转换为嵌入，得到最终的音频特征 a_i。因此，我们用 a_i 简写表示 a_i^c/a_i^d。对于 a_i^d/t_i，我们将语义标记和文本标记的查找嵌入作为输入，并用各自的头生成每个标记，如第2节所述。

使用此符号，我们在表3中制定了以下预训练任务，并如下介绍。

Table 3: List of pre-training tasks. We design three categories pre-training tasks, including: 1) audio/text unimodal pre-training; 2) audio-text mapping pre-training; 3) audio-text interleaving pre-training. Notation: a_i^d and a_i^c denotes the discrete semantic tokens and continuous acoustic vectors for audio segment i respectively, a_i denotes the combination of a_i^d and a_i^c for audio segment i, underline means it will receive loss during training.

Category	Pre-training Task	Task Formulation	Task Weight
Audio/Text Unimodal	Text Only	$\underline{t_1}, \underline{t_2}, ..., \underline{t_N}$	7
	Audio Only	$\underline{a_1^d}, \underline{a_2^d}, ..., \underline{a_N^d}$	1
Audio-Text Mapping	Audio to Text	$a_1, \underline{t_1}, a_2, \underline{t_2}, ..., a_N, \underline{t_N}$	1
	Text to Audio	$t_1, \underline{a_1^d}, t_2, \underline{a_2^d}, ..., t_N, \underline{a_N^d}$	1
Audio-Text Interleaving	Audio to Semantic	$a_1, \underline{a_2^d}, a_3, \underline{a_4^d}, ..., a_{N-1}, \underline{a_N^d}$	1
	Audio to Text	$a_1, \underline{t_2}, a_3, \underline{t_4}, ..., a_{N-1}, \underline{t_N}$	1
	Audio to Semantic and Text	$a_1, \underline{a_2^d}/\underline{t_2}, a_3, \underline{a_4^d}/\underline{t_4}, ..., a_{N-1}, \underline{a_N^d}/\underline{t_N}$	2

4.1.1 Audio/Text Unimodal Pre-training

We first learn the knowledge of text and audio separately. For text pre-training, we directly utilize the text data in MoonLight [44], which is high-quality and comprehensive for training large language models. We apply next-token prediction on text tokens solely. For audio pre-training, for each segment S_i, we apply next-token prediction on its discrete semantic token sequence a_i^d.

4.1.2 Audio-Text Mapping Pre-training

Intuitively, in order to align audio and text in a unified space, it is helpful to learn a mapping between two modalities. Thus, we design the automatic speech recognition (ASR) and text-to-speech synthesis (TTS) pre-training tasks. For ASR, we formulate the training sequence as $\{a_1, t_1, a_2, t_2, ..., a_N, t_N\}$. For TTS, we formulate the training sequence as $\{t_1, a_1^d, t_2, a_2^d, ..., t_N, a_N^d\}$. We only calculate the loss on text tokens for ASR and on audio semantic tokens for TTS.

4.1.3 Audio-Text Interleaving Pre-training

To further bridge the gap between audio and text modalities, we design three audio-text interleaving pre-training tasks.

- **Audio to semantic token interleaving.** We formulate the training sequence as $\{a_1, a_2^d, a_3, a_4^d, ..., a_{N-1}, a_N^d\}$[4]. Then we only calculate the loss on the semantic audio tokens a_i^d, but not on a_{i-1}.
- **Audio to text interleaving.** We formulate the training sequence as $\{a_1, t_2, a_3, t_4, ..., a_{N-1}, t_N\}$. We only calculate the loss on text tokens t_i.
- **Audio to semantic token + text interleaving.** We formulate the training sequence as $\{a_1, a_2^d/t_2, a_3, a_4^d/t_4, ..., a_{N-1}, a_N^d/t_N\}$. For a_i^d/t_i, as the semantic audio token sequence is always longer than the text token sequence, the prediction of the semantic token is like a streaming text-to-speech task as in Section 4.1.2. Empirically, we find that the prediction of the first few semantic tokens is hard because the model needs to concurrently predict the next text token and its semantic audio token. We address this issue by delaying the prediction of the first several semantic audio tokens by prepending 6 special blank tokens (6 is determined by trading off the generation quality and latency according to preliminary experiments) to the semantic audio tokens.

[4] It is also possible that the first segment is a_1^d, or the last segment is a_N.

表3：预训练任务列表。我们设计了三类预训练任务，包括：1）音频/文本单模态预训练；2）音频-文本映射预训练；3）音频-文本交错预训练。符号说明：a_i^d 和 a_i^c 分别表示音频片段 i 的离散语义标记和连续声学向量，a_i 表示音频片段 i 的 a_i^d 和 a_i^c 的组合，带下划线的表示在训练过程中会受到损失。

Category	Pre-training Task	Task Formulation	Task Weight
Audio/Text Unimodal	Text Only	$\underline{t_1}, \underline{t_2}, ..., \underline{t_N}$	7
	Audio Only	$\underline{a_1^d}, \underline{a_2^d}, ..., \underline{a_N^d}$	1
Audio-Text Mapping	Audio to Text	$a_1, \underline{t_1}, a_2, \underline{t_2}, ..., a_N, \underline{t_N}$	1
	Text to Audio	$t_1, \underline{a_1^d}, t_2, \underline{a_2^d}, ..., t_N, \underline{a_N^d}$	1
Audio-Text Interleaving	Audio to Semantic	$a_1, \underline{a_2^d}, a_3, \underline{a_4^d}, ..., a_{N-1}, \underline{a_N^d}$	1
	Audio to Text	$a_1, \underline{t_2}, a_3, \underline{t_4}, ..., a_{N-1}, \underline{t_N}$	1
	Audio to Semantic and Text	$a_1, \underline{a_2^d}/\underline{t_2}, a_3, \underline{a_4^d}/\underline{t_4}, ..., a_{N-1}, \underline{a_N^d}/\underline{t_N}$	2

4.1.1 音频/文本单模预训练

我们首先分别学习文本和音频的知识。对于文本预训练，我们直接利用MoonLight [44]中的文本数据，该数据具有高质量和全面性，适用于训练大型语言模型。我们仅对文本标记应用下一词预测。对于音频预训练，对于每个片段S_i，我们对其离散语义标记序列a_i^d应用下一词预测。

4.1.2 音频-文本映射预训练

直观地，为了在统一空间中对齐音频和文本，学习两种模态之间的映射是有帮助的。因此，我们设计了自动语音识别（ASR）和文本到语音合成（TTS）的预训练任务。对于ASR，我们将训练序列表述为$\{a_1, t_1, a_2, t_2, ..., a_N, t_N\}$。对于TTS，我们将训练序列表述为$\{t_1, a_1^d, t_2, a_2^d, ..., t_N, a_N^d\}$。我们仅在ASR中对文本标记计算损失，在TTS中对音频语义标记计算损失。

4.1.3 音频-文本交错预训练

为了进一步弥合音频和文本模态之间的差距，我们设计了三种音频-文本交错预训练任务。

- 音频与语义标记交错。我们将训练序列表述为 $\{a_1, a_2^d, a_3, a_4^d, ..., a_{N-1}, a_N^d\}$[4]。然后我们只在语义音频标记 a_i^d 上计算损失，而不在 a_{i-1} 上计算。

- 音频与文本交错。我们将训练序列表述为 $\{a_1, t_2, a_3, t_4, ..., a_{N-1}, t_N\}$。我们只在文本标记 t_i 上计算损失。

- 音频到语义标记 + 的文本交错。我们将训练序列表述为 $\{a_1, a_2^d/t_2, a_3, a_4^d/t_4, ..., a_{N-1}, a_N^d/t_N\}$。对于 a_i^d/t_i，由于语义音频标记序列总是比文本标记序列长，语义标记的预测类似于第 4.1.2 节中的流式文本转语音任务。经验上，我们发现前几个语义标记的预测较为困难，因为模型需要同时预测下一个文本标记及其对应的语义音频标记。我们通过在语义音频标记前添加 6 个特殊空白标记（6 个空白标记的数量是通过权衡生成质量和延迟在初步实验中确定的）来推迟前几个语义音频标记的预测，从而解决了这个问题。

[4] It is also possible that the first segment is a_1^d, or the last segment is a_N.

4.1.4 Pre-training Recipe

We initialize the audio LLM of Kimi-Audio from the pre-trained Qwen2.5 7B model [76] and extend its vocabulary with semantic audio tokens and special tokens. We perform pre-training on the above pre-training tasks with the corresponding task weight $1:7:1:1:1:1:2$, as shown in Table 3. We pre-train Kimi-Audio using 585B audio tokens and 585B text tokens with 1 epoch. We use AdamW [48] optimizer with a learning rate schedule from $2e^{-5}$ to $2e^{-6}$ in a cosine decay. We use 1% tokens for learning rate warmup.

The continuous acoustic feature extraction module in audio tokenizer is initialized from Whisper large-v3 [58], which can capture the fine-grained acoustic characteristics inherent in the input audio signal. During the initial phases of model pretraining (about 20% tokens in pretraining), the parameters of this whisper-based feature extractor are kept frozen. Subsequently, the feature extractor is unfrozen, enabling its parameters to be fine-tuned jointly with the rest of the model, allowing it to adapt more specifically to the nuances of the training data and the requirements of the target tasks.

4.2 Supervised Fine-tuning

4.2.1 Formulation

After pre-training Kimi-Audio with massive real-world audio and text data, we perform supervised fine-tuning to equip it with the ability of instruction-following. We have the following design choices: 1) considering the downstream tasks are diverse, we do not set special task switching operations, but use natural language as instructions for each task; 2) for instruction, we construct both the audio and text versions (i.e., the audio is generated by Kimi-TTS in a zero-shot way given the text) and randomly choose one during training; 3) to enhance the robustness of instruction-following capability, we construct 200 instructions for ASR task and 30 instructions for other tasks by LLM and randomly choose one for each training sample. As described in Section 3.2, we build about 300K hours of data for supervised fine-tuning.

4.2.2 Fine-tuning Recipe

As shown in Table 1 and Table 2, we fine-tune Kimi-Audio on each data source with 2-4 epochs based on comprehensive ablation experiments. We use AdamW [48] optimizer with a learning rate schedule from $1e^{-5}$ to $1e^{-6}$ in a cosine decay. We use 10% tokens for learning rate warmup.

4.3 Training of Audio Detokenizer

We train the audio detokenizer in three stages. Firstly, we use about 1M hours of audio from the pre-training data described in Section 3.1 and pre-train both the flow-matching model and the vocoder to learn audio with diverse timbre, prosody, and quality. Secondly, we adopt the chunk-wise fine-tuning strategy with a dynamic chunk size from 0.5 seconds to 3 seconds on the same pre-training data [32]. Finally, we fine-tune on the high-quality single-speaker recording data from the Kimi-Audio speaker.

5 Inference and Deployment

Kimi-Audio is designed to handle various audio-related tasks, such as speech recognition, audio understanding, audio-to-text chat, and speech-to-speech conversation. We take real-time speech-

4.1.4 预训练方案

我们将 Kimi-Audio 的音频大模型初始化为预训练的 Qwen2.5 7B 模型 [76]，并用语义音频标记和特殊标记扩展其词汇表。我们在上述预训练任务上进行预训练，任务权重比为 1 : 7 : 1 : 1 : 1 : 1 : 2，如表 3 所示。我们使用 585B 音频标记和 585B 文本标记进行 1 个 epoch 的预训练。我们采用 AdamW [48] 优化器，学习率从 $2e^{-5}$ 逐步衰减到 $2e^{-6}$，采用余弦衰减。我们使用 1% 的标记进行学习率预热。

音频分词器中的连续声学特征提取模块是从Whisper large-v3 [58]初始化的，能够捕捉输入音频信号中固有的细粒度声学特性。在模型预训练的初始阶段（大约20%的预训练标记），该基于Whisper的特征提取器的参数保持冻结状态。随后，特征提取器被解冻，使其参数能够与模型的其他部分共同微调，从而使其更具体地适应训练数据的细微差别和目标任务的需求。

4.2 监督微调

4.2.1 公式化

在用大量真实世界的音频和文本数据对Kimi-Audio进行预训练后，我们进行有监督的微调，以赋予其指令跟随能力。我们有以下设计选择：1）考虑到下游任务的多样性，我们不设置特殊的任务切换操作，而是使用自然语言作为每个任务的指令；2）对于指令，我们构建了音频和文本两个版本（即音频由Kimi-TTS在零样本方式下根据文本生成），在训练过程中随机选择其中一个；3）为了增强指令跟随能力的鲁棒性，我们通过LLM为ASR任务构建了200个指令，为其他任务构建了30个指令，并在每个训练样本中随机选择一个。如第3.2节所述，我们为有监督微调构建了约300K小时的数据。

4.2.2 微调方案

如表1和表2所示，我们在每个数据源上对Kimi-Audio进行了2-4轮的微调，基于全面的消融实验。我们使用AdamW [48]优化器，学习率在$1e^{-5}$到$1e^{-6}$之间采用余弦衰减的调度。我们使用10%的tokens进行学习率预热。

4.3 音频去标记器的训练

我们将音频解码器的训练分为三个阶段。首先，我们使用来自第3.1节所述预训练数据的约1百万小时的音频，预训练流匹配模型和声码器，以学习具有多样音色、韵味和质量的音频。其次，我们采用逐块微调策略，在相同的预训练数据上以动态块大小从0.5秒到3秒进行微调 [32]。最后，我们在Kimi-Audio说话人提供的高质量单一说话人录音数据上进行微调。

5 推理与部署

Kimi-Audio旨在处理各种与音频相关的任务，例如语音识别、音频理解、音频转文本聊天以及语音对话。我们支持实时语音-

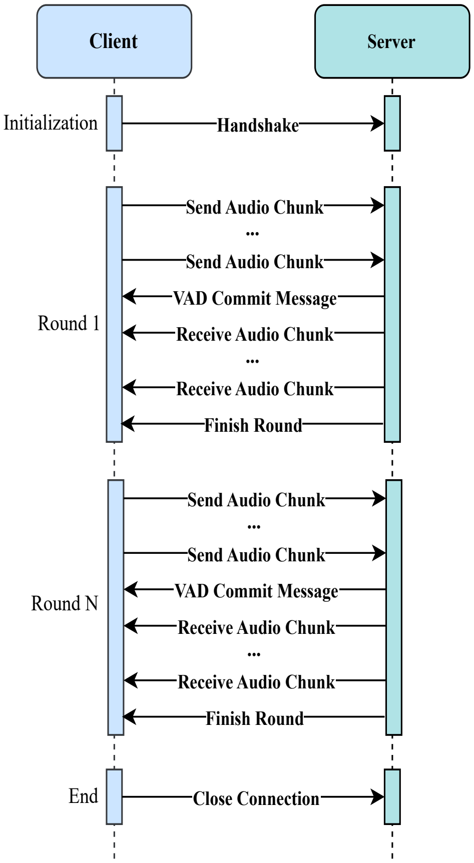

Figure 4: The client-server communication for real-time speech-to-speech conversation in Kimi-Audio.

to-speech conversation as an example to illustrate the practices in Kimi-Audio deployment, since this task is more complicated than the rest of audio tasks in terms of infrastructure and engineering efforts. We first introduce the workflow of real-time speech conversation between the client (e.g., Kimi APP or web browser) and the server (Kimi-Audio service) and then describe the practices of product deployment.

5.1 Workflow of Real-Time Speech Conversation

The workflow of a speech-to-speech conversation between the user client (e.g., Kimi APP) and the server (Kimi-Audio service) is illustrated in Figure 4. The workflow proceeds in the following manner for each conversation round:

- The user speaks to the client (e.g., Kimi APP or web browser), and the audio data is collected and streamed to the server.
- On the server side, a voice activity detection (VAD) module determines if the user has finished speaking.
- Once the user stops speaking, the server sends a commit signal and initiates the inference process of the Kimi-Audio model.
- During inference, the client receives audio chunks in real-time as they are generated and starts playing them for the user.

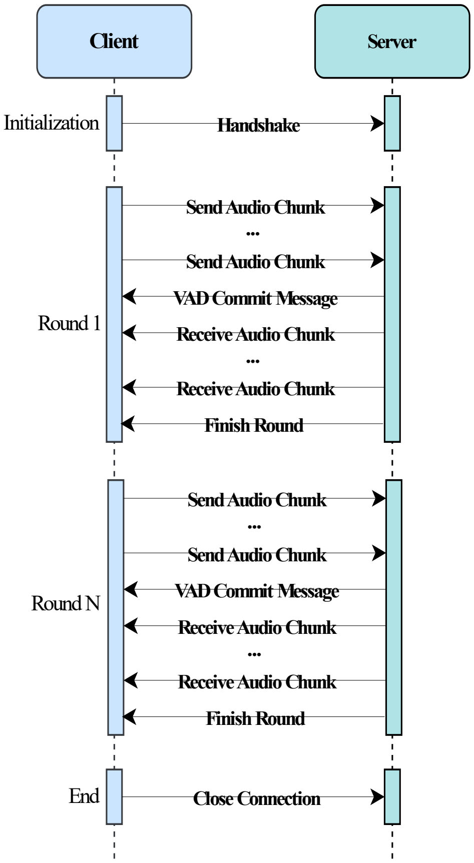

图4：Kimi-Audio中实时语音到语音对话的客户端-服务器通信。

以语音对话为例，说明Kimi-Audio部署中的实践，因为这个任务在基础设施和工程方面比其他音频任务更为复杂。我们首先介绍客户端（例如Kimi APP或网页浏览器）与服务器（Kimi-Audio服务）之间的实时语音对话的工作流程，然后描述产品部署的实践。

5.1 实时语音对话的工作流程

用户客户端（例如 Kimi APP）与服务器（Kimi-Audio 服务）之间的语音到语音对话的工作流程如图 4 所示。每轮对话的工作流程如下进行:

- 用户与客户端（例如，Kimi APP 或网页浏览器）进行对话，音频数据被采集并流式传输到服务器。
- 在服务器端，语音活动检测（VAD）模块判断用户是否已完成发言。
- 一旦用户停止说话，服务器会发送提交信号并启动Kimi-Audio模型的推理过程。
- 在推理过程中，客户端会实时接收生成的音频片段，并开始为用户播放。

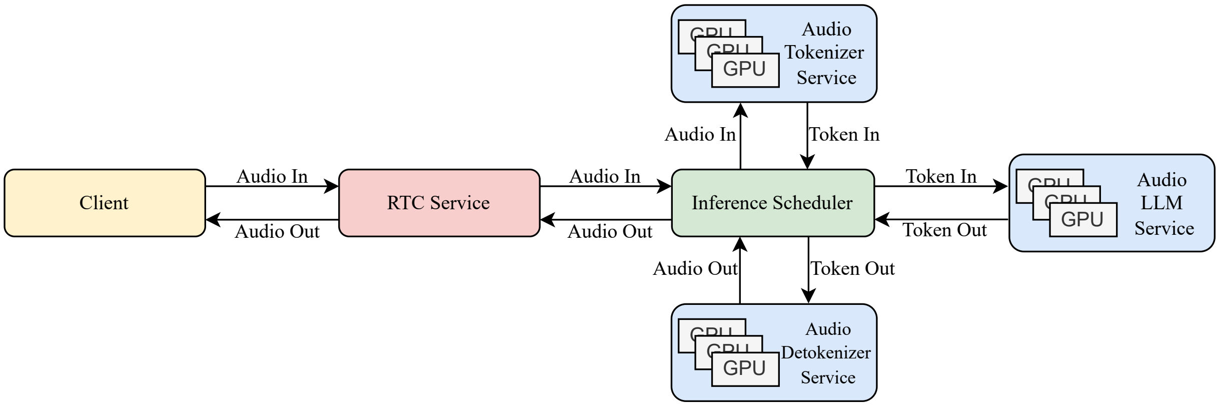

Figure 5: The workflow of production deployment for real-time speech-to-speech conversation in Kimi-Audio.

- The client (mobile phone or web browser) plays the received audio chunks back to the user.

The inference process of Kimi-Audio on the server side for each round follows these steps. First, the input audio is converted to discrete semantic tokens and continuous acoustic vectors using the audio tokenizer. Next, the input to the Audio LLM is assembled by concatenating the system prompt tokens, audio tokens, and conversation history tokens. The token sequence is then passed to the Audio LLM, which generates output tokens. Finally, the output tokens are converted back into an audio waveform using the detokenizer.

5.2 Production Deployment

As shown in Figure 5, in a production environment, all core components: Audio Tokenizer, Audio LLM, and Audio Detokenizer, are computationally intensive, requiring a scalable and efficient infrastructure. To address this, we designed the production deployment architecture as follows.

Kimi-Audio RTC Service. This service interfaces with the client, receiving audio from the user, forwarding it to the Inference Scheduler, and returning the generated audio chunks to the client. We use the WebRTC protocol to ensure a stable and low-latency connection.

Inference Scheduler. The Inference Scheduler manages the conversation flow by maintaining conversation history as tokens in a storage backend. For each interaction round, it performs the following steps:

- Call the Tokenizer Service to convert the user's audio into tokens.
- Construct the model input by combining the new tokens with the conversation history.
- Send the input to the LLM Service to generate response tokens.
- Call the Detokenizer Service to convert the response tokens into audio output.

Additionally, it stores all output tokens as part of the ongoing conversation history to ensure continuity in the dialogue.

Tokenizer/Detokenizer/LLM Services: These services handle model inference and are equipped with a load balancer and multiple inference instances to handle requests in parallel, ensuring scalability.

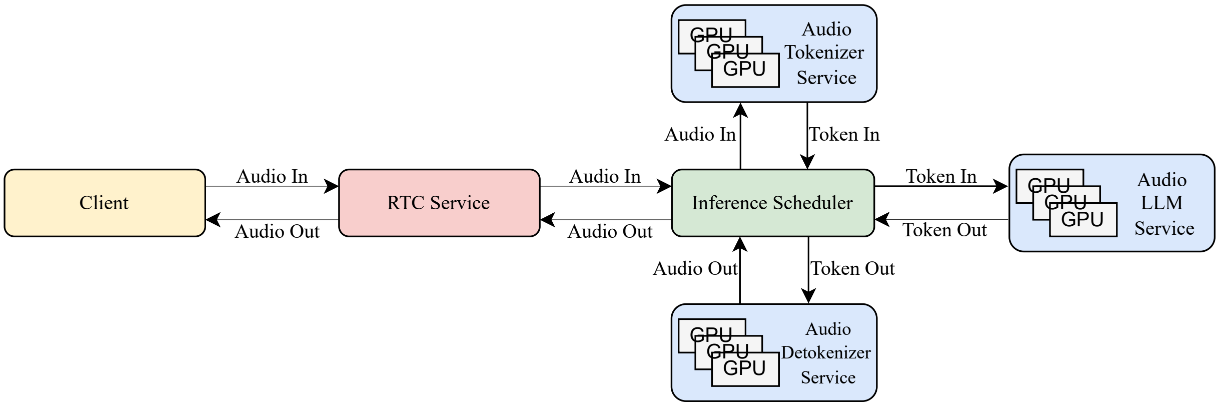

图5：Kimi-Audio中实时语音到语音对话的生产部署工作流程。

- 客户端（手机或网页浏览器）将接收到的音频片段播放给用户。

Kimi-Audio 在服务器端每一轮的推理过程遵循以下步骤。首先，使用音频标记器将输入音频转换为离散的语义标记和连续的声学向量。接下来，通过连接系统提示标记、音频标记和对话历史标记，组装成输入到 Audio LLM 的序列。然后，将该标记序列传递给 Audio LLM，生成输出标记。最后，使用解标记器将输出标记转换回音频波形。

5.2 生产部署

如图5所示，在生产环境中，所有核心组件：音频分词器、音频大模型和音频去分词器，都是计算密集型的，需配备可扩展且高效的基础设施。为此，我们将生产部署架构设计如下。

Kimi-Audio RTC 服务。该服务与客户端接口，接收用户的音频，将其转发给推理调度器，并将生成的音频片段返回给客户端。我们使用 WebRTC 协议以确保连接的稳定性和低延迟。

推理调度器。推理调度器通过将对话历史作为令牌存储在后端存储中来管理对话流程。对于每个交互轮次，它执行以下步骤：

- 调用 Tokenizer 服务将用户的音频转换为标记。
- 通过将新令牌与对话历史结合，构建模型输入。
- 将输入发送到LLM服务以生成响应标记。
- 调用 Detokenizer 服务将响应的标记转换为音频输出。

此外，它将所有输出标记作为持续对话历史的一部分存储，以确保对话的连续性。

分词器/去分词器/大模型服务：这些服务负责模型推理，配备负载均衡器和多个推理实例，以并行处理请求，确保可扩展性。

This modular architecture ensures that Kimi-Audio can scale effectively to meet the performance demands of real-time speech interactions while maintaining low latency and high availability in production.

6 Evaluation

Evaluating audio foundation models and comparing with previous state-of-the-art systems are challenging, due to some inherent issues in the audio community. Thus, we first develop a fair, reproducible, and comprehensive evaluation toolkit for audio foundation models in Section 6.1, and then evaluate Kimi-Audio on a variety of audio processing tasks including speech recognition, general audio understanding, audio-to-text chat, and speech conversation, and compare Kimi-Audio with previous systems to demonstrate the advantages in Section 6.2.

6.1 Evaluation Toolkit

Even if an audio foundation model is fully open-source, it is still troublesome to reproduce the same results as reported in its paper or technical report, let alone those closed-source models. We analyze the challenges in evaluating and comparing audio foundation models in various audio processing tasks as follows:

- **Limitations in Metrics.** Current practices suffer from inconsistent metric implementations (e.g., variations in Word Error Rate calculation due to different text normalizations) and inadequate assessment methods (e.g., relying solely on exact string matching for tasks like audio question answering fails to capture the semantic correctness of complex LLM responses).
- **Diverse Configurations.** Reproducibility is severely hampered by the high sensitivity of model performance to inference parameters such as decoding temperature, system prompts, and task prompts.
- **Lack of Generation Evaluation** While progress has been made in understanding tasks, assessing the quality and coherence of the generated audio response still lacks benchmarks.

To address these critical limitations, we develop an open-source evaluation toolkit for audio foundation models on audio understanding, generation, and conversation tasks. It currently integrates and supports Kimi-Audio and a series of recent audio LLMs [11, 74, 84, 41, 28], and can be leveraged to evaluate any other audio foundation models. The toolkit provides the following features and benefits:

- We implement a standardized WER calculation (based on Qwen-2-Audio [11]) and integrate GPT-4o-mini as an intelligent judge (following [8]) for tasks like audio question answering. This approach overcomes the limitations of inconsistent metrics and simplistic string matching, enabling fair comparison.
- Our toolkit offers a single unified platform supporting diverse models and versions, simplifying side-by-side comparisons. It provides a crucial structure for defining and sharing standardized inference parameters and prompting strategies ("recipes"), thereby directly addressing inconsistencies in evaluation setups and fostering greater reproducibility across different research works.

这种模块化架构确保了Kimi-Audio能够有效扩展，以满足实时语音交互的性能需求，同时在生产环境中保持低延迟和高可用性。

6 评估

评估音频基础模型并与之前的最先进系统进行比较具有挑战性，因为音频社区存在一些固有的问题。因此，我们首先在第6.1节开发了一个公平、可复现且全面的音频基础模型评估工具包，然后在第6.2节对Kimi-Audio在包括语音识别、通用音频理解、音频转文本聊天和语音对话在内的多种音频处理任务中进行了评估，并将Kimi-Audio与之前的系统进行了比较，以展示其优势。

6.1 评估工具包

即使一个音频基础模型完全开源，仍然很难复现其论文或技术报告中报道的相同结果，更不用说那些闭源模型了。我们分析了在各种音频处理任务中评估和比较音频基础模型所面临的挑战，具体如下：

- 指标的局限性。目前的做法存在指标实现不一致的问题（例如，由于不同的文本归一化导致的词错误率计算差异）以及评估方法不足（例如，在语音问答等任务中仅依赖精确字符串匹配，无法捕捉复杂大型语言模型响应的语义正确性）。

- 多样的配置。模型性能对推理参数（如解码温度、系统提示和任务提示）的高度敏感性严重影响了可复现性。

- 缺乏生成评估 目前虽然在理解任务方面取得了一定的进展，但在评估生成的音频响应的质量和连贯性方面仍然缺乏基准。

为了解决这些关键限制，我们开发了一个开源的评估工具包，用于音频基础模型在音频理解、生成和对话任务中的评估。它目前集成并支持Kimi-Audio以及一系列最新的音频大模型[11, 74, 84, 41, 28]，并可用于评估任何其他音频基础模型。该工具包提供了以下功能和优势：

- 我们实现了标准化的WER计算（基于Qwen-2-Audio [11]），并集成了GPT-4o-mini作为智能判定（遵循[8]）用于音频问答等任务。这种方法克服了指标不一致和简单字符串匹配的局限性，实现了公平的比较。

- 我们的工具包提供了一个统一的平台，支持多种模型和版本，简化了并排比较。它为定义和共享标准化推理参数和提示策略（“配方”）提供了关键结构，从而直接解决评估设置中的不一致问题，并促进不同研究工作之间的更高可复现性。

Table 4: Performance of Kimi-Audio and baseline models on ASR task. Best results are in bold.

Datasets	Model	Performance (WER↓)
LibriSpeech [54] test-clean | test-other	Qwen2-Audio-base	1.74 | 4.04
	Baichuan-Audio-base	3.02 | 6.04
	Step-Audio-chat	3.19 | 10.67
	Qwen2.5-Omni	2.37 | 4.21
	Kimi-Audio	**1.28** | **2.42**
Fleurs [12] zh | en	Qwen2-Audio-base	3.63 | 5.20
	Baichuan-Audio-base	4.15 | 8.07
	Step-Audio-chat	4.26 | 8.56
	Qwen2.5-Omni	2.92 | **4.17**
	Kimi-Audio	**2.69** | 4.44
AISHELL-1 [4]	Qwen2-Audio-base	1.52
	Baichuan-Audio-base	1.93
	Step-Audio-chat	2.14
	Qwen2.5-Omni	1.13
	Kimi-Audio	**0.60**
AISHELL-2 [17] ios	Qwen2-Audio-base	3.08
	Baichuan-Audio-base	3.87
	Step-Audio-chat	3.89
	Qwen2.5-Omni	**2.56**
	Kimi-Audio	**2.56**
WenetSpeech [85] test-meeting | test-net	Qwen2-Audio-base	8.40 | 7.64
	Baichuan-Audio-base	13.28 | 10.13
	Step-Audio-chat	10.83 | 9.47
	Qwen2.5-Omni	7.71 | 6.04
	Kimi-Audio	**6.28** | **5.37**
Kimi-ASR Internal Testset subset1 | subset2	Qwen2-Audio-base	2.31 | 3.24
	Baichuan-Audio-base	3.41 | 5.60
	Step-Audio-chat	2.82 | 4.74
	Qwen2.5-Omni	1.53 | 2.68
	Kimi-Audio	**1.42** | **2.44**

- We record and release an evaluation benchmark to test the ability of audio LLMs on speech conversation from the perspective of 1) speech control on emotion, speed, and accent; 2) empathy conversation; and 3) diverse styles such as storytelling and tongue twister.

We open-source this toolkit to the community `https://github.com/MoonshotAI/Kimi-Audio-Evalkit`. We believe this toolkit can serve as a valuable asset to advance the field by promoting more reliable and comparable benchmarking. We actively encourage researchers and developers to utilize it, contribute by adding new models and datasets, and help refine standardized evaluation protocols and inference recipes. In this way, we can build a better ecosystem for the audio community.

6.2 Evaluation Results

In this section, based on our evaluation toolkit, we detail the evaluation of Kimi-Audio across a comprehensive suite of audio processing tasks, including Automatic Speech Recognition (ASR),

表4：Kimi-Audio 和基线模型在语音识别任务中的性能。最佳结果以加粗显示。

Datasets	Model	Performance (WER↓)
LibriSpeech [54] test-clean | test-other	Qwen2-Audio-base	1.74 | 4.04
	Baichuan-Audio-base	3.02 | 6.04
	Step-Audio-chat	3.19 | 10.67
	Qwen2.5-Omni	2.37 | 4.21
	Kimi-Audio	**1.28 | 2.42**
Fleurs [12] zh | en	Qwen2-Audio-base	3.63 | 5.20
	Baichuan-Audio-base	4.15 | 8.07
	Step-Audio-chat	4.26 | 8.56
	Qwen2.5-Omni	2.92 | **4.17**
	Kimi-Audio	**2.69** | 4.44
AISHELL-1 [4]	Qwen2-Audio-base	1.52
	Baichuan-Audio-base	1.93
	Step-Audio-chat	2.14
	Qwen2.5-Omni	1.13
	Kimi-Audio	**0.60**
AISHELL-2 [17] ios	Qwen2-Audio-base	3.08
	Baichuan-Audio-base	3.87
	Step-Audio-chat	3.89
	Qwen2.5-Omni	**2.56**
	Kimi-Audio	**2.56**
WenetSpeech [85] test-meeting | test-net	Qwen2-Audio-base	8.40 | 7.64
	Baichuan-Audio-base	13.28 | 10.13
	Step-Audio-chat	10.83 | 9.47
	Qwen2.5-Omni	7.71 | 6.04
	Kimi-Audio	**6.28 | 5.37**
Kimi-ASR Internal Testset subset1 | subset2	Qwen2-Audio-base	2.31 | 3.24
	Baichuan-Audio-base	3.41 | 5.60
	Step-Audio-chat	2.82 | 4.74
	Qwen2.5-Omni	1.53 | 2.68
	Kimi-Audio	**1.42 | 2.44**

- 我们记录并发布了一个评估基准，用于测试音频大模型在语音对话方面的能力，评估内容包括：1）情感、语速和口音的语音控制；2）共情对话；以及3）多样的风格，如讲故事和绕口令。

我们将这个工具包开源给社区 https://github.com/MoonshotAI/ Kimi-Audio-Evalkit。我们相信这个工具包可以作为推动该领域发展的宝贵资产，通过促进更可靠和可比的基准测试。我们积极鼓励研究人员和开发者使用它，贡献新的模型和数据集，并帮助完善标准化的评估协议和推理方案。通过这种方式，我们可以为音频社区建立一个更好的生态系统。

6.2 评估结果

在本节中，基于我们的评估工具包，我们详细介绍了Kimi-Audio在一系列全面的音频处理任务中的评估，包括自动语音识别（ASR），

Table 5: Performance of Kimi-Audio and baseline models on audio understanding task. Best results are in bold.

Datasets	Model	Performance↑
MMAU [60] music | sound | speech	Qwen2-Audio-base	58.98 | 69.07 | 52.55
	Baichuan-chat	49.10 | 59.46 | 42.47
	GLM-4-Voice	38.92 | 43.54 | 32.43
	Step-Audio-chat	49.40 | 53.75 | 47.75
	Qwen2.5-Omni	**62.16** | 67.57 | 53.92
	Kimi-Audio	61.68 | **73.27** | **60.66**
ClothoAQA [43] test | dev	Qwen2-Audio-base	71.73 | 72.63
	Baichuan-chat	48.02 | 48.16
	Step-Audio-chat	45.84 | 44.98
	Qwen2.5-Omni	**72.86** | 73.12
	Kimi-Audio	71.24 | **73.18**
VocalSound [23]	Qwen2-Audio-base	93.82
	Baichuan-Audio-base	58.17
	Step-Audio-chat	28.58
	Qwen2.5-Omni	93.73
	Kimi-Audio	**94.85**
Nonspeech7k [59]	Qwen2-Audio-base	87.17
	Baichuan-chat	59.03
	Step-Audio-chat	21.38
	Qwen2.5-Omni	69.89
	Kimi-Audio	**93.93**
MELD [56]	Qwen2-Audio-base	51.23
	Baichuan-chat	23.59
	Step-Audio-chat	33.54
	Qwen2.5-Omni	49.83
	Kimi-Audio	**59.13**
TUT2017 [53]	Qwen2-Audio-base	33.83
	Baichuan-Audio-base	27.9
	Step-Audio-chat	7.41
	Qwen2.5-Omni	43.27
	Kimi-Audio	**65.25**
CochlScene [31] test | dev	Qwen2-Audio-base	52.69 | 50.96
	Baichuan-Audio-base	34.93 | 34.56
	Step-Audio-chat	10.06 | 10.42
	Qwen2.5-Omni	63.82 | 63.82
	Kimi-Audio	**79.84 | 80.99**

audio understanding, audio-to-text chat, and speech conversation. We compare Kimi-Audio against other audio foundation models (Qwen2-Audio [11], Baichuan-Audio [41], Step-Audio [28], GLM-4-Voice [84], and Qwen2.5-Omini [73]) using established benchmarks and internal test sets.

6.2.1 Automatic Speech Recognition

The ASR capabilities of Kimi-Audio were evaluated on diverse datasets spanning multiple languages and acoustic conditions. As presented in Table 4, Kimi-Audio consistently demonstrates superior performance compared to previous models. We report Word Error Rate (WER) on these datasets, where lower values indicate better performance.

表5：Kimi-Audio 和基线模型在音频理解任务中的性能。最佳结果以加粗显示。

Datasets	Model	Performance↑
MMAU [60] music | sound | speech	Qwen2-Audio-base	58.98 | 69.07 | 52.55
	Baichuan-chat	49.10 | 59.46 | 42.47
	GLM-4-Voice	38.92 | 43.54 | 32.43
	Step-Audio-chat	49.40 | 53.75 | 47.75
	Qwen2.5-Omni	**62.16** | 67.57 | 53.92
	Kimi-Audio	61.68 | **73.27** | **60.66**
ClothoAQA [43] test | dev	Qwen2-Audio-base	71.73 | 72.63
	Baichuan-chat	48.02 | 48.16
	Step-Audio-chat	45.84 | 44.98
	Qwen2.5-Omni	**72.86** | 73.12
	Kimi-Audio	71.24 | **73.18**
VocalSound [23]	Qwen2-Audio-base	93.82
	Baichuan-Audio-base	58.17
	Step-Audio-chat	28.58
	Qwen2.5-Omni	93.73
	Kimi-Audio	**94.85**
Nonspeech7k [59]	Qwen2-Audio-base	87.17
	Baichuan-chat	59.03
	Step-Audio-chat	21.38
	Qwen2.5-Omni	69.89
	Kimi-Audio	**93.93**
MELD [56]	Qwen2-Audio-base	51.23
	Baichuan-chat	23.59
	Step-Audio-chat	33.54
	Qwen2.5-Omni	49.83
	Kimi-Audio	**59.13**
TUT2017 [53]	Qwen2-Audio-base	33.83
	Baichuan-Audio-base	27.9
	Step-Audio-chat	7.41
	Qwen2.5-Omni	43.27
	Kimi-Audio	**65.25**
CochlScene [31] test | dev	Qwen2-Audio-base	52.69 | 50.96
	Baichuan-Audio-base	34.93 | 34.56
	Step-Audio-chat	10.06 | 10.42
	Qwen2.5-Omni	63.82 | 63.82
	Kimi-Audio	**79.84** | **80.99**

音频理解、音频转文本聊天和语音对话。我们使用既定的基准和内部测试集，将Kimi-Audio与其他音频基础模型（Qwen2-Audio [11]、Baichuan-Audio [41]、Step-Audio [28]、GLM-4-Voice [84] 和 Qwen2.5-Omini [73]）进行比较。

6.2.1 自动语音识别

Kimi-Audio 的语音识别（ASR）能力在涵盖多种语言和声学条件的多样化数据集上进行了评估。如表4所示，Kimi-Audio 一贯表现出优越的性能，优于之前的模型。我们在这些数据集上报告了词错误率（WER），其中较低的值表示更好的性能。

Table 6: Performance of Kimi-Audio and baseline models on the tasks of audio-to-text chat. Best results are in bold.

Datasets	Model	Performance↑
OpenAudioBench [41] AlpacaEval | Llama Questions | Reasoning QA | TriviaQA | Web Questions	Qwen2-Audio-chat	57.19 | 69.67 | 42.77 | 40.30 | 45.20
	Baichuan-chat	59.65 | 74.33 | 46.73 | 55.40 | 58.70
	GLM-4-Voice	57.89 | 76.00 | 47.43 | 51.80 | 55.40
	Step-Audio-chat	56.53 | 72.33 | 60.00 | 56.80 | **73.00**
	Qwen2.5-Omni	72.76 | 75.33 | **63.76** | 57.06 | 62.80
	Kimi-Audio	**75.73** | **79.33** | 58.02 | **62.10** | 70.20
VoiceBench [8] AlpacaEval | CommonEval | SD-QA | MMSU	Qwen2-Audio-chat	3.69 | 3.40 | 35.35 | 35.43
	Baichuan-chat	4.00 | 3.39 | 49.64 | 48.80
	GLM-4-Voice	4.06 | 3.48 | 43.31 | 40.11
	Step-Audio-chat	3.99 | 2.99 | 46.84 | 28.72
	Qwen2.5-Omni	4.33 | 3.84 | 57.41 | 56.38
	Kimi-Audio	**4.46** | **3.97** | **63.12** | **62.17**
VoiceBench [8] OpenBookQA | IFEval | AdvBench | Avg	Qwen2-Audio-chat	49.01 | 22.57 | 98.85 | 54.72
	Baichuan-chat	63.30 | 41.32 | 86.73 | 62.51
	GLM-4-Voice	52.97 | 24.91 | 88.08 | 57.17
	Step-Audio-chat	31.87 | 29.19 | 65.77 | 48.86
	Qwen2.5-Omni	79.12 | 53.88 | 99.62 | 72.83
	Kimi-Audio	**83.52** | **61.10** | **100.00** | **76.93**

Notably, Kimi-Audio achieves the best results on the widely-used LibriSpeech [54] benchmark, attaining error rates of 1.28 on test-clean and 2.42 on test-other, significantly outperforming models like Qwen2-Audio-base and Qwen2.5-Omni. For Mandarin ASR benchmarks, Kimi-Audio sets SOTA results on AISHELL-1 [4] (0.60) and AISHELL-2 ios [17] (2.56). Furthermore, it excels on the challenging WenetSpeech [85] dataset, achieving the lowest error rates on both test-meeting and test-net. Finally, evaluation on our internal Kimi-ASR test set confirms the model robustness. These results demonstrate the strong ASR capabilities of Kimi-Audio across various domains and languages.

6.2.2 Audio Understanding

Beyond speech recognition, we assess Kimi-Audio's ability to comprehend diverse audio signals, including music, sound events, and speech. Table 5 summarizes the performance on various audio understanding benchmarks, where higher scores generally indicate better performance.

On the MMAU benchmark [60], Kimi-Audio demonstrates superior understanding across sound category (73.27), and speech category (60.66). Similarly, it outperforms other models on the MELD [56] speech emotion understanding task, scoring 59.13. Kimi-Audio also leads on tasks involving non-speech sound classification (VocalSound [23] and Nonspeech7k [59]) and acoustic scene classification (TUT2017 [53] and CochlScene [31]). These results highlight Kimi-Audio's advanced capabilities in interpreting complex acoustic information beyond simple speech recognition.

6.2.3 Audio-to-Text Chat

We evaluate the ability of Kimi-Audio to engage in text conversations based on audio input using the OpenAudioBench [41] and VoiceBench benchmarks [8]. These benchmarks assess various aspects like instruction following, question answering, and reasoning. Performance metrics are

表6：Kimi-Audio和基线模型在音频转文本聊天任务中的性能。最佳结果用粗体显示。

Datasets	Model	Performance↑
OpenAudioBench [41] AlpacaEval | Llama Questions | Reasoning QA | TriviaQA | Web Questions	Qwen2-Audio-chat	57.19 | 69.67 | 42.77 | 40.30 | 45.20
	Baichuan-chat	59.65 | 74.33 | 46.73 | 55.40 | 58.70
	GLM-4-Voice	57.89 | 76.00 | 47.43 | 51.80 | 55.40
	Step-Audio-chat	56.53 | 72.33 | 60.00 | 56.80 | **73.00**
	Qwen2.5-Omni	72.76 | 75.33 | **63.76** | 57.06 | 62.80
	Kimi-Audio	**75.73** | **79.33** | 58.02 | **62.10** | 70.20
VoiceBench [8] AlpacaEval | CommonEval | SD-QA | MMSU	Qwen2-Audio-chat	3.69 | 3.40 | 35.35 | 35.43
	Baichuan-chat	4.00 | 3.39 | 49.64 | 48.80
	GLM-4-Voice	4.06 | 3.48 | 43.31 | 40.11
	Step-Audio-chat	3.99 | 2.99 | 46.84 | 28.72
	Qwen2.5-Omni	4.33 | 3.84 | 57.41 | 56.38
	Kimi-Audio	**4.46 | 3.97 | 63.12 | 62.17**
VoiceBench [8] OpenBookQA | IFEval | AdvBench | Avg	Qwen2-Audio-chat	49.01 | 22.57 | 98.85 | 54.72
	Baichuan-chat	63.30 | 41.32 | 86.73 | 62.51
	GLM-4-Voice	52.97 | 24.91 | 88.08 | 57.17
	Step-Audio-chat	31.87 | 29.19 | 65.77 | 48.86
	Qwen2.5-Omni	79.12 | 53.88 | 99.62 | 72.83
	Kimi-Audio	**83.52 | 61.10 | 100.00 | 76.93**

值得注意的是，Kimi-Audio在广泛使用的LibriSpeech [54]基准测试中取得了最佳结果，在test-clean上达到1.28的错误率，在test-other上达到2.42，显著优于Qwen2-Audio-base和Qwen2.5-Omni等模型。在普通话语音识别基准测试中，Kimi-Audio在AISHELL-1 [4]（0.60）和AISHELL-2 ios [17]（2.56）上设定了SOTA（最先进技术）结果。此外，它在具有挑战性的WenetSpeech [85]数据集上表现出色，在test-meeting和test-net上都实现了最低的错误率。最后，在我们内部的Kimi-ASR测试集上的评估证实了模型的鲁棒性。这些结果展示了Kimi-Audio在各个领域和语言中的强大语音识别能力。

6.2.2 音频理解

除了语音识别之外，我们还评估了Kimi-Audio理解各种音频信号的能力，包括音乐、声音事件和语音。表5总结了在各种音频理解基准上的表现，得分越高通常表示性能越好。

在MMAU基准测试[60]中，Kimi-Audio在声音类别（73.27）和语音类别（60.66）方面表现出色。同样，它在MELD[56]语音情感理解任务中也优于其他模型，得分为59.13。Kimi-Audio还在涉及非语音声音分类（VocalSound[23]和Nonspeech7k[59]）以及声学场景分类（TUT2017[53]和CochlScene[31]）的任务中领先。这些结果突显了Kimi-Audio在解释超越简单语音识别的复杂声学信息方面的先进能力。

6.2.3 音频转文本聊天

我们使用OpenAudioBench [41]和VoiceBench基准测试 [8]评估Kimi-Audio基于音频输入进行文本对话的能力。这些基准测试评估了指令遵循、问答和推理等各个方面。性能指标为
{v*}

Table 7: Performance of Kimi-Audio and baseline models on speech conversation. Best results are in bold and second-best results are underlined.

Model	Speed Control	Accent Control	Emotion Control	Empathy	Style Control	Avg
GPT-4o	4.21	**3.65**	4.05	**3.87**	**4.54**	**4.06**
Step-Audio-chat	3.25	2.87	3.33	3.05	4.14	3.33
GLM-4-Voice	3.83	3.51	3.77	3.07	4.04	3.65
GPT-4o-mini	3.15	2.71	4.24	3.16	4.01	3.45
Kimi-Audio	**4.30**	3.45	**4.27**	3.39	4.09	3.90

benchmark-specific, with higher scores indicating better conversational ability. The results are presented in Table 6.

On OpenAudioBench, Kimi-Audio achieves state-of-the-art performance on several sub-tasks, including AlpacaEval, Llama Questions, and TriviaQA, and achieves highly competitive performance on Reasoning QA and Web Questions.

The VoiceBench evaluation further confirms Kimi-Audio's strengths. It consistently outperforms all compared models on AlpacaEval (4.46), CommonEval (3.97), SD-QA (63.12), MMSU (62.17), OpenBookQA (83.52), Advbench (100.00), and IFEval (61.10). Kimi-Audio's overall performance across these comprehensive benchmarks demonstrates its superior ability in audio-based conversation and complex reasoning tasks.

6.2.4 Speech Conversation

Finally, we assess the end-to-end speech conversation capabilities of Kimi-Audio based on subjective evaluations across multiple dimensions. As shown in Table 7, Kimi-Audio was compared against models like GPT-4o and GLM-4-Voice based on human ratings (on a 1-5 scale, higher is better).

Excluding GPT-4o, Kimi-Audio achieves the highest scores for emotion control, empathy, and speed control. While GLM-4-Voice shows slightly better accent control, Kimi-Audio achieves a strong overall average score of 3.90. This score is higher than Step-Audio-chat (3.33), GPT-4o-mini (3.45), and GLM-4-Voice (3.65), and remains a small margin with GPT-4o (4.06). Overall, the evaluation results demonstrate Kimi-Audio's proficiency in generating expressive and controllable speech.

7 Related Work

The application of large language models (LLMs) to audio tasks has led to remarkable progress across a wide range of domains, including automatic speech recognition (ASR), audio understanding, text-to-speech synthesis (TTS), general audio generation, and speech-based human-computer interaction. These efforts explore how to bridge the gap between raw acoustic signals and linguistic reasoning by treating audio as a tokenizable sequence, enabling LLMs to process or generate audio in a language-like manner.

ASR and Audio Understanding A number of LLM-based systems have been developed to improve automatic speech recognition (ASR) and broader audio understanding tasks. Whisper [58] serves as a powerful audio encoder, and when combined with large language models (LLMs), it significantly enhances the performance of speech understanding systems. This approach has been successfully utilized in models such as Qwen-Audio [10], Qwen2-Audio [11], SALMONN [63], and OSUM [21]

表7：Kimi-Audio 和基线模型在语音对话中的性能。最佳结果用粗体显示，第二佳结果用下划线标出。

Model	Speed Control	Accent Control	Emotion Control	Empathy	Style Control	Avg
GPT-4o	<u>4.21</u>	**3.65**	4.05	**3.87**	**4.54**	**4.06**
Step-Audio-chat	3.25	2.87	3.33	3.05	<u>4.14</u>	3.33
GLM-4-Voice	3.83	3.51	3.77	3.07	4.04	3.65
GPT-4o-mini	3.15	2.71	<u>4.24</u>	3.16	4.01	3.45
Kimi-Audio	**4.30**	<u>3.45</u>	**4.27**	<u>3.39</u>	4.09	<u>3.90</u>

基准特定，得分越高表示对话能力越强。结果显示在表6中。

在OpenAudioBench上，Kimi-Audio在多个子任务中实现了最先进的性能，包括AlpacaEval、Llama Questions和TriviaQA，并在Reasoning QA和Web Questions上表现出极具竞争力的性能。

VoiceBench 评估进一步确认了 Kimi-Audio 的优势。它在 AlpacaEval (4.46)、CommonEval (3.97)、SD-QA (63.12)、MMSU (62.17)、OpenBookQA (83.52)、Advbench (100.00) 和 IFEval (61.10) 上始终优于所有对比模型。Kimi-Audio 在这些全面基准测试中的整体表现展示了其在基于音频的对话和复杂推理任务中的卓越能力。

6.2.4 语音对话

最后，我们基于多个维度的主观评估，评估了Kimi-Audio的端到端语音对话能力。如表7所示，Kimi-Audio与GPT-4o和GLM-4-Voice等模型进行了比较，评估采用人类评分（1-5分，分数越高越好）。

不包括 GPT-4o，Kimi-Audio 在情感控制、共情和速度控制方面都取得了最高分。虽然 GLM-4-Voice 在口音控制方面略胜一筹，但 Kimi-Audio 的整体平均分达到了 3.90。这一分数高于 Step-Audio-chat（3.33）、GPT-4o-mini（3.45）和 GLM-4-Voice（3.65），与 GPT-4o（4.06）之间仍有一定差距。总体而言，评估结果显示 Kimi-Audio 在生成富有表现力且可控的语音方面具有较强的能力。

7 相关工作

大规模语言模型（LLMs）在音频任务中的应用在包括自动语音识别（ASR）、音频理解、文本到语音合成（TTS）、通用音频生成以及基于语音的人机交互等多个领域取得了显著的进展。这些努力探索了如何通过将音频作为可标记的序列来弥合原始声学信号与语言推理之间的差距，从而使LLMs能够以类似语言的方式处理或生成音频。

语音识别（ASR）与音频理解 许多基于大型语言模型（LLM）的系统已被开发出来，以改善自动语音识别（ASR）和更广泛的音频理解任务。Whisper [58] 作为一个强大的音频编码器，当与大型语言模型（LLMs）结合时，显著提升了语音理解系统的性能。这一方法已在如 Qwen-Audio [10]、Qwen2-Audio [11]、SALMONN [63] 和 OSUM [21] 等模型中成功应用。

These systems, however, are mostly limited to understanding tasks and do not natively support audio output.

TTS and Audio Generation For speech synthesis and general audio generation, models such as AudioLM [3], VALL-E [70], and LLASA [80] tokenize audio via neural codecs and use decoder-only language models for autoregressive generation. Other efforts like UniAudio [77] and VoiceBox [37] extend these methods with hybrid tokenization or flow matching to improve quality and control. While these models can produce high-fidelity audio, they typically focus on generation only and lack understanding and conversation capabilities or instruction-following speech interaction.

Speech Conversation and Real-Time Dialogue Recent models have moved toward enabling real-time, end-to-end speech interaction. Moshi [14], GLM-4-Voice [84], and Mini-Omni [72] adopt interleaved or parallel decoding to support simultaneous generation of text and audio tokens, facilitating low-latency dialogue systems. OmniFlatten [86] introduces a progressive training pipeline to adapt a frozen LLM for full-duplex conversation. LLaMA-Omni [18] and Freeze-Omni [71] further refine duplex speech interaction through streaming decoders or multi-task alignment strategies. However, these systems often rely heavily on speech-only datasets and compromise language modeling quality or generality due to limited pre-training.

Toward Universal Audio-Language Foundation Models A small number of recent works aim to unify understanding and generation within a single multimodal model. Baichuan-Audio [41] uses a multi-codebook discretization to capture both semantic and acoustic features, enabling real-time interaction and strong question-answering capabilities. However, its focus on speech domain limits its broader applicability, especially for non-speech audio tasks like music or environmental sound. Step-Audio [28], on the other hand, provides a powerful solution for real-time speech interaction with a 130B-parameter unified speech-text multimodal model. While Step-Audio demonstrates strong performance, its dependency on synthetic voice data generation and the high computational costs associated with its 130B parameters pose significant barriers to accessibility and cost-effectiveness for a broader user base. Qwen2.5-Omni [74] introduces a Thinker-Talker architecture for simultaneous text and speech decoding and achieves strong benchmark results, but its design primarily emphasizes streaming inference, and it lacks an extensive pre-training phase on raw audio.

Kimi-Audio Kimi-Audio advances beyond these limitations by introducing a truly universal and open-source audio foundation model that supports speech recognition, audio understanding, audio generation, and speech conversation in a single framework. It adopts a hybrid input representation combining Whisper-derived continuous acoustic features and discrete semantic tokens (12.5Hz), ensuring rich perception and efficient modeling. The audio LLM is initialized from a text LLM, with dual-generation heads for text and audio, and a chunk-wise flow-matching detokenizer paired with BigVGAN [38] to produce expressive and low-latency speech.

Most critically, Kimi-Audio features extensive multimodal pretraining on 13 million hours of curated audio data across speech, music, and environmental sound—a scale far exceeding prior works. The pretraining tasks include audio-only, text-only, audio-to-text, and interleaved modalities, which enable the model to learn generalizable audio reasoning and maintain strong language abilities. This is followed by instruction-based fine-tuning across diverse tasks, leading to state-of-the-art results on ASR, general audio understanding, audio-text chat, and speech conversation benchmarks.

这些系统主要局限于理解任务，原生不支持音频输出。

TTS 和音频生成　对于语音合成和一般音频生成，模型如AudioLM [3]、VALL-E [70] 和 LLASA [80] 通过神经编解码器对音频进行标记，并使用仅解码器的语言模型进行自回归生成。其他方法如 UniAudio [77] 和 VoiceBox [37] 通过混合标记或流匹配扩展了这些方法，以提高质量和控制。虽然这些模型可以生成高保真度的音频，但它们通常只专注于生成，缺乏理解和对话能力或指令跟随的语音交互。

语音对话与实时对话 近年来的模型已朝着实现实时端到端语音交互的方向发展。Moshi [14]、GLM-4-Voice [84] 和 Mini-Omni [72] 采用交错或并行解码方式，支持文本和音频标记的同时生成，从而促进低延迟的对话系统。OmniFlatten [86] 引入了一个渐进式训练流程，以适应冻结的大模型（LLM）进行全双工对话。LLaMA-Omni [18] 和 Freeze-Omni [71] 通过流式解码器或多任务对齐策略，进一步优化双工语音交互。然而，这些系统通常严重依赖仅含语音的数据集，并由于预训练有限，影响了语言模型的质量或通用性。

迈向通用音频-语言基础模型　少数近期工作旨在在单一的多模态模型中统一理解与生成。百川-音频 [41] 采用多码本离散化方法，捕捉语义和声学特征，实现实时交互和强大的问答能力。然而，它专注于语音领域，限制了其在更广泛应用中的适用性，尤其是非语音音频任务，如音乐或环境声音。Step-音频 [28] 提供了一种强大的实时语音交互解决方案，采用130B参数的统一语音-文本多模态模型。虽然Step-音频表现出色，但其对合成语音数据生成的依赖以及与130B参数相关的高计算成本，成为更广泛用户基础获取和成本效益的重大障碍。Qwen2.5-Omni [74] 引入了Thinker-Talker架构，用于同时进行文本和语音解码，并取得了优异的基准测试结果，但其设计主要强调流式推理，且缺乏对原始音频的广泛预训练阶段。

Kimi-Audio Kimi-Audio 超越了这些限制，推出了一款真正的通用开源音频基础模型，支持语音识别、音频理解、音频生成和语音对话，统一在一个框架中。它采用混合输入表示，将基于 Whisper 的连续声学特征与离散语义标记（12.5Hz）相结合，确保丰富的感知能力和高效的建模。音频大型语言模型（LLM）从文本LLM初始化，配备双重生成头，用于文本和音频，并配备块式流匹配的去标记器，结合 BigVGAN [38]，以生成富有表现力且低延迟的语音。

最关键的是，Kimi-Audio 在 1300 万小时的精选音频数据（涵盖语音、音乐和环境声音）上进行了广泛的多模态预训练，这一规模远超以往的工作。预训练任务包括纯音频、纯文本、音频到文本以及交错的模态，这使模型能够学习具有泛化能力的音频推理并保持强大的语言能力。随后，通过基于指令的微调，涵盖各种任务，取得了在自动语音识别（ASR）、通用音频理解、音频-文本聊天和语音对话基准测试中的最先进成果。

In contrast to existing models that are either limited in scope, lack pre-training, or are not publicly available, Kimi-Audio is a fully open-source, pre-trained, instruction-followable, and real-time capable model. Its comprehensive coverage, scalable architecture, and broad task alignment make it a significant step toward general-purpose audio intelligence.

8 Challenges and Future Trends

Although Kimi-Audio has achieved significant advancements in building universal audio foundation models, several challenges remain in the quest for more capable and intelligent audio processing systems. We describe the challenges and point out several exciting future directions as follows.

- **From Audio Transcription to Audio Description.** Current pre-training paradigms for audio foundation models typically leverage audio-text pre-training to bridge the gap between text and audio, where the text is obtained from audio (speech) by ASR transcription. However, text transcription focuses on the content of spoken words (what is said), neglecting important information in audio, such as paralanguage information (e.g., emotion, style, timbre, tone), acoustic scene, and non-linguistic sounds. Thus, it is important to introduce descriptive text (i.e., audio caption) to depict the audio in richer context. Incorporating both transcriptive and descriptive text of the audio enables models to better understand and generate not only spoken language but also complex acoustic environments, paving the way for more nuanced, multimodal audio processing systems, and thus more general and versatile audio intelligence.
- **Better Audio Representations.** Current audio leverages semantic tokens or acoustic tokens as its representations. Semantic tokens are typically obtained by ASR-based auxiliary loss, which focuses on transcription-oriented information and fails to capture rich acoustic details crucial for understanding and generation. Acoustic tokens are typically learned by audio reconstruction loss, which focuses on description-oriented acoustic details and fails to capture abstractive semantic information that is crucial to bridge to text intelligence. A valuable research direction is to develop representations that integrate both transcription-oriented semantic information and description-oriented acoustic features, encompassing nuances like speaker identity, emotion, and environmental sounds while maintaining high-level abstractive information, which is paramount for more sophisticated audio understanding and generation.
- **Throw Away ASR and TTS in Audio Modeling.** Current audio foundation models rely heavily on ASR and TTS to generate training data in both the pre-training and fine-tuning stages. The quality of training data is constrained by the text recognition accuracy of ASR and expressiveness/diversity/quality of synthesized speech in TTS. In this way, the audio models behave like a sophisticated distillation of existing ASR and TTS systems. As a result, they can hardly achieve performance far beyond the ceiling of ASR/TTS and cannot achieve truly autonomous audio intelligence. An important future direction is to train audio models without relying on ASR/TTS-based pseudo audio data but relying on native audio data, which can result in much higher performance upperbound.

References

[1] Rosana Ardila et al. “Common voice: A massively-multilingual speech corpus”. In: *arXiv preprint arXiv:1912.06670* (2019).

[2] Beijing Academy of Artificial Intelligence (BAAI). “Infinity Instruct”. In: *arXiv preprint arXiv:2406.XXXX* (2024).

与现有模型相比，这些模型要么范围有限、缺乏预训练，要么不对公众开放，而Kimi-Audio是一个完全开源、预训练、可遵循指令且具备实时能力的模型。其全面的覆盖范围、可扩展的架构以及广泛的任务适应性，使其成为迈向通用音频智能的重要一步。

8 个挑战与未来趋势

虽然Kimi-Audio在构建通用音频基础模型方面取得了重大进展，但在追求更强大、更智能的音频处理系统的过程中仍然面临一些挑战。我们描述了这些挑战，并指出了几个令人振奋的未来方向，如下所示。

- 从音频转录到音频描述。目前的音频基础模型预训练范式通常利用音频-文本预训练来弥合文本与音频之间的差距，其中文本是通过ASR转录从音频（语音）中获得的。然而，文本转录主要关注口语内容（说了什么），忽略了音频中的重要信息，例如旁声信息（如情感、风格、音色、语调）、声学场景以及非语言声音。因此，引入描述性文本（即音频字幕）以在更丰富的背景下描述音频变得尤为重要。结合音频的转录文本和描述性文本，使模型能够更好地理解和生成不仅是口语，还包括复杂声学环境的内容，为更细腻的多模态音频处理系统铺平了道路，从而实现更通用、更灵活的音频智能。

- 更优的音频表示。目前的音频利用语义标记或声学标记作为其表示。语义标记通常通过基于ASR的辅助损失获得，侧重于转录导向的信息，未能捕捉对理解和生成至关重要的丰富声学细节。声学标记通常通过音频重建损失学习，侧重于描述导向的声学细节，未能捕捉对连接文本智能至关重要的抽象语义信息。一个有价值的研究方向是开发结合转录导向的语义信息和描述导向的声学特征的表示，涵盖说话人身份、情感和环境声音等细微差别，同时保持高层次的抽象信息，这对于更复杂的音频理解和生成至关重要。

- 在音频建模中抛弃ASR和TTS。目前的音频基础模型在预训练和微调阶段都严重依赖于ASR和TTS来生成训练数据。训练数据的质量受到ASR的文本识别准确率和TTS中合成语音的表现力/多样性/质量的限制。这样一来，音频模型就像是对现有ASR和TTS系统的复杂蒸馏。因此，它们几乎难以超越ASR/TTS的性能上限，也无法实现真正的自主音频智能。未来的一个重要方向是训练不依赖于基于ASR/TTS的伪音频数据，而是依赖于原生音频数据，这将带来更高的性能上限。

参考文献

[1] Rosana Ardila 等人。“Common voice：一个大规模多语种语音语料库”。发表于：*arXiv preprint arXiv:1912.06670* (2019)。 [2] 北京人工智能研究院（BAAI）。“Infinity Instruct”。发表于：*arXiv preprint arXiv:2406.XXXX* (2024)。

[3] Zalán Borsos et al. “Audiolm: a language modeling approach to audio generation”. In: *IEEE/ACM transactions on audio, speech, and language processing* 31 (2023), pp. 2523–2533.

[4] Hui Bu et al. “Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline”. In: *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*. IEEE. 2017, pp. 1–5.

[5] Guoguo Chen et al. “Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio”. In: *arXiv preprint arXiv:2106.06909* (2021).

[6] Honglie Chen et al. “Vggsound: A large-scale audio-visual dataset”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 721–725.

[7] Qian Chen et al. “Minmo: A multimodal large language model for seamless voice interaction”. In: *arXiv preprint arXiv:2501.06282* (2025).

[8] Yiming Chen et al. “VoiceBench: Benchmarking LLM-Based Voice Assistants”. In: *arXiv preprint arXiv:2410.17196* (2024).

[9] Zesen Cheng et al. “Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms”. In: *arXiv preprint arXiv:2406.07476* (2024).

[10] Yunfei Chu et al. “Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models”. In: *arXiv preprint arXiv:2311.07919* (2023).

[11] Yunfei Chu et al. “Qwen2-audio technical report”. In: *arXiv preprint arXiv:2407.10759* (2024).

[12] Alexis Conneau et al. “Fleurs: Few-shot learning evaluation of universal representations of speech”. In: *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2023, pp. 798–805.

[13] DeepSeek-AI. *DeepSeek-V3 Technical Report*. 2024. arXiv: 2412.19437 [cs.CL]. URL: https://arxiv.org/abs/2412.19437.

[14] Alexandre Défossez et al. “Moshi: a speech-text foundation model for real-time dialogue”. In: *arXiv preprint arXiv:2410.00037* (2024).

[15] Chandeepa Dissanayake et al. *OpenBezoar: Small, Cost-Effective and Open Models Trained on Mixes of Instruction Data*. 2024. arXiv: 2404.12195 [cs.CL].

[16] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. “Clotho: An audio captioning dataset”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 736–740.

[17] Jiayu Du et al. “Aishell-2: Transforming mandarin asr research into industrial scale”. In: *arXiv preprint arXiv:1808.10583* (2018).

[18] Qingkai Fang et al. “Llama-omni: Seamless speech interaction with large language models”. In: *arXiv preprint arXiv:2409.06666* (2024).

[19] Eduardo Fonseca et al. “Fsd50k: an open dataset of human-labeled sound events”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2021), pp. 829–852.

[20] Zhifu Gao et al. “Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition”. In: *arXiv preprint arXiv:2206.08317* (2022).

[21] Xuelong Geng et al. “OSUM: Advancing Open Speech Understanding Models with Limited Resources in Academia”. In: *arXiv preprint arXiv:2501.13306* (2025).

[22] Sreyan Ghosh et al. “Gama: A large audio-language model with advanced audio understanding and complex reasoning abilities”. In: *arXiv preprint arXiv:2406.11768* (2024).

[23] Yuan Gong, Jin Yu, and James Glass. “Vocalsound: A Dataset for Improving Human Vocal Sounds Recognition”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 151–155. DOI: 10.1109/ICASSP43922.2022.9746828.

[24] Aaron Grattafiori et al. “The llama 3 herd of models”. In: *arXiv preprint arXiv:2407.21783* (2024).

[25] Haorui He et al. “Emilia: A Large-Scale, Extensive, Multilingual, and Diverse Dataset for Speech Generation”. In: *arXiv preprint arXiv:2501.15907* (2025).

[26] Haorui He et al. “Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation”. In: *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2024, pp. 885–890.

[27] T. Heittola et al. *TAU Urban Acoustic Scenes 2022 Mobile, Development dataset*. Zenodo. Mar. 2022. DOI: 10.5281/zenodo.6337421.

[28] Ailin Huang et al. “Step-audio: Unified understanding and generation in intelligent speech interaction”. In: *arXiv preprint arXiv:2502.11946* (2025).

[29] Aaron Hurst et al. “Gpt-4o system card”. In: *arXiv preprint arXiv:2410.21276* (2024).

[3] Zalán Borsos 等人。“Audiolm：一种音频生成的语言建模方法”。发表于：*IEEE/ACM transactions on audio, speech, and language processing* 31 (2023)，第 2523–2533 页。 [4] Hui Bu 等人。“Aishell-1：一个开源的普通话语音语料库及其语音识别基线”。发表于：*2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*。IEEE。2017，第 1–5 页。 [5] Guoguo Chen 等人。“Gigaspeech：一个不断发展的多领域自动语音识别语料库，包含 10,000 小时的转录音频”。发表于：*arXiv preprint arXiv:2106.06909* (2021)。 [6] Honglie Chen 等人。“Vggsound：一个大规模的音频-视觉数据集”。发表于：*ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*。IEEE。2020，第 721–725 页。 [7] Qian Chen 等人。“Minmo：一个用于无缝语音交互的多模态大规模语言模型”。发表于：*arXiv preprint arXiv:2501.06282* (2025)。 [8] Yiming Chen 等人。“VoiceBench：基于大规模语言模型的语音助手基准测试”。发表于：*arXiv preprint arXiv:2410.17196* (2024)。 [9] Zesen Cheng 等人。“Videollama 2：在视频大模型中推进时空建模和音频理解”。发表于：*arXiv preprint arXiv:2406.07476* (2024)。 [10] Yunfei Chu 等人。“Qwen-audio：通过统一的大规模音频-语言模型推进通用音频理解”。发表于：*arXiv preprint arXiv:2311.07919* (2023)。 [11] Yunfei Chu 等人。“Qwen2-audio 技术报告”。发表于：*arXiv preprint arXiv:2407.10759* (2024)。 [12] Alexis Conneau 等人。“Fleurs：少样本学习评估通用语音表示”。发表于：*2022 IEEE Spoken Language Technology Workshop (SLT)*。IEEE。2023，第 798–805 页。 [13] DeepSeek-AI。*DeepSeek-V3 Technical Report*。2024。arXiv：2412.19437 [cs.CL]。网址：https://arxiv.org/abs/2412.19437。 [14] Alexandre Défossez 等人。“Moshi：用于实时对话的语音文本基础模型”。发表于：*arXiv preprint arXiv:2410.00037* (2024)。 [15] Chandeepa Dissanayake 等人。*OpenBezoar: Small, Cost-Effective and Open Models Trained on Mixes of Instruction Data*。2024。arXiv：2404.12195 [cs.CL]。 [16] Konstantinos Drossos、Samuel Lipping 和 Tuomas Virtanen。“Clotho：一个音频字幕数据集”。发表于：*ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*。IEEE。2020，第 736–740 页。 [17] Jiayu Du 等人。“Aishell-2：将普通话自动语音识别研究转化为工业规模”。发表于：*arXiv preprint arXiv:1808.10583* (2018)。 [18] Qingkai Fang 等人。“Llama-omni：与大规模语言模型的无缝语音交互”。发表于：*arXiv preprint arXiv:2409.06666* (2024)。 [19] Eduardo Fonseca 等人。“Fsd50k：一个由人类标注的声音事件开源数据集”。发表于：*IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2021)，第 829–852 页。 [20] Zhifu Gao 等人。“Paraformer：一种快速且准确的非自回归端到端语音识别的并行变换器”。发表于：*arXiv preprint arXiv:2206.08317* (2022)。 [21] Xuelong Geng 等人。“OSUM：在学术界利用有限资源推进开放语音理解模型”。发表于：*arXiv preprint arXiv:2501.13306* (2025)。 [22] Sreyan Ghosh 等人。“Gama：具有先进音频理解和复杂推理能力的大型音频-语言模型”。发表于：*arXiv preprint arXiv:2406.11768* (2024)。 [23] Yuan Gong、Jin Yu 和 James Glass。“Vocalsound：用于提升人类发声识别的数据集”。发表于：*ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*。2022，第 151–155 页。DOI：10.1109/ICASSP43922.2022.9746828。 [24] Aaron Grattafiori 等人。“Llama 3 模型群”。发表于：*arXiv preprint arXiv:2407.21783* (2024)。 [25] Haorui He 等人。“Emilia：一个大规模、广泛、多语种、多样化的语音生成数据集”。发表于：*arXiv preprint arXiv:2501.15907* (2025)。 [26] Haorui He 等人。“Emilia：一个广泛、多语种、多样化的语音数据集，用于大规模语音生成”。发表于：*2024 IEEE Spoken Language Technology Workshop (SLT)*。IEEE。2024，第 885–890 页。 [27] T. Heittola 等人。Zenodo。2022年3月。DOI：10.5281/zenodo.6337421。 [28] Ailin Huang 等人。“Step-audio：在智能语音交互中实现统一理解与生成”。发表于：*arXiv preprint arXiv:2502.11946* (2025)。 [29] Aaron Hurst 等人。“Gpt-4o 系统卡”。发表于：*arXiv preprint arXiv:2410.21276* (2024)。

[30] Philip Jackson and Sana ul haq. *Surrey Audio-Visual Expressed Emotion (SAVEE) database*. Apr. 2011.

[31] Il-Young Jeong and Jeongsoo Park. "Cochlscene: Acquisition of acoustic scene data using crowdsourcing". In: *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE. 2022, pp. 17–21.

[32] Zeqian Ju et al. "MoonCast: High-Quality Zero-Shot Podcast Generation". In: *arXiv preprint arXiv:2503.14345* (2025).

[33] Wei Kang et al. "Libriheavy: A 50,000 hours ASR corpus with punctuation casing and context". In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2024, pp. 10991–10995.

[34] Chris Dongjoo Kim et al. "Audiocaps: Generating captions for audios in the wild". 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)*. 2019, pp. 119–132.

[35] Zhifeng Kong et al. "Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities". In: *arXiv preprint arXiv:2402.01831* (2024).

[36] Nathan Lambert et al. "T\" ulu 3: Pushing frontiers in open language model post-training". In: *arXiv preprint arXiv:2411.15124* (2024).

[37] Matthew Le et al. "Voicebox: Text-guided multilingual universal speech generation at scale". In: *Advances in neural information processing systems* 36 (2023), pp. 14005–14034.

[38] Sang-gil Lee et al. "Bigvgan: A universal neural vocoder with large-scale training". In: *arXiv preprint arXiv:2206.04658* (2022).

[39] Guangyao Li et al. "Learning to answer questions in dynamic audio-visual scenarios". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19108–19118.

[40] Jia Li et al. "Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions". In: *Hugging Face repository* 13 (2024), p. 9.

[41] Tianpeng Li et al. "Baichuan-audio: A unified framework for end-to-end speech interaction". In: *arXiv preprint arXiv:2502.17239* (2025).

[42] Wing Lian et al. *OpenOrca: An Open Dataset of GPT Augmented FLAN Reasoning Traces*. https://https://huggingface.co/datasets/Open-Orca/OpenOrca. 2023.

[43] Samuel Lipping et al. "Clotho-aqa: A crowdsourced dataset for audio question answering". In: *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE. 2022, pp. 1140–1144.

[44] Jingyuan Liu et al. *Muon is Scalable for LLM Training*. 2025. arXiv: 2502.16982 [cs.LG]. URL: https://arxiv.org/abs/2502.16982.

[45] Rui-Bo Liu et al. "Convincing Audio Generation Based on LLM and Speech Tokenization". In: *2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE. 2024, pp. 591–595.

[46] Songting Liu. "Zero-shot Voice Conversion with Diffusion Transformers". In: *arXiv preprint arXiv:2411.09943* (2024).

[47] Steven R Livingstone and Frank A Russo. "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English". In: *PloS one* 13.5 (2018), e0196391.

[48] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[49] Yi Luo and Jianwei Yu. "Music Source Separation With Band-Split RNN". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), pp. 1893–1901. DOI: 10.1109/TASLP.2023.3271145.

[50] Linhan Ma et al. "Wenetspeech4tts: A 12,800-hour mandarin tts corpus for large speech generation model benchmark". In: *arXiv preprint arXiv:2406.05763* (2024).

[51] Irene Martín-Morató and Annamaria Mesaros. "What is the ground truth? reliability of multi-annotator data for audio tagging". In: *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE. 2021, pp. 76–80.

[52] Xinhao Mei et al. "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2024).

[53] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. "TUT Database for Acoustic Scene Classification and Sound Event Detection". In: *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*. Budapest, Hungary, 2016.

[54] Vassil Panayotov et al. "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.

[30] Philip Jackson 和 Sana ul haq. *Surrey Audio-Visual Expressed Emotion (SAVEE) database*。2011年4月。 [31] Il-Young Jeong 和 Jeongsoo Park. “Cochlscene：使用众包获取声景数据”。发表于：*2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*。IEEE。2022年，第17–21页。 [32] Zeqian Ju 等人。“MoonCast：高质量零样本播客生成”。发表于：*arXiv preprint arXiv:2503.14345* (2025)。 [33] Wei Kang 等人。“Libriheavy：带标点大小写和上下文的50,000小时语音识别语料库”。发表于：*ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*。IEEE。2024年，第10991–10995页。 [34] Chris Dongjoo Kim 等人。“Audiocaps：为野外音频生成字幕”。发表于：*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)*。2019年，第119–132页。 [35] Zhifeng Kong 等人。“Audio flamingo：具有少样本学习和对话能力的新型音频语言模型”。发表于：*arXiv preprint arXiv:2402.01831* (2024)。 [36] Nathan Lambert 等人。“T\” ulu 3：推动开放式语言模型后训练的前沿”。发表于：*arXiv preprint arXiv:2411.15124* (2024)。 [37] Matthew Le 等人。“Voicebox：大规模文本引导的多语言通用语音生成”。发表于：*Advances in neural information processing systems* 36 (2023)，第14005–14034页。 [38] Sang-gil Lee 等人。“Bigvgan：具有大规模训练的通用神经声码器”。发表于：*arXiv preprint arXiv:2206.04658* (2022)。 [39] Guangyao Li 等人。“在动态音频视觉场景中学习回答问题”。发表于：*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*。2022年，第19108–19118页。 [40] Jia Li 等人。“Numinamath：在ai4maths中最大的公共数据集，包含86万对竞赛数学题目和解答”。发表于：*Hugging Face repository* 13 (2024)，第9页。 [41] Tianpeng Li 等人。“Baichuan-audio：端到端语音交互的统一框架”。发表于：*arXiv preprint arXiv:2502.17239* (2025)。 [42] Wing Lian 等人。*OpenOrca: An Open Dataset of GPT Augmented FLAN Reasoning Traces*。https://https://huggingface.co/datasets/Open-Orca/OpenOrca。2023年。 [43] Samuel Lipping 等人。“Clotho-aqa：一个众包的音频问答数据集”。发表于：*2022 30th European Signal Processing Conference (EUSIPCO)*。IEEE。2022年，第1140–1144页。 [44] Jingyuan Liu 等人。*Muon is Scalable for LLM Training*。2025年。arXiv: 2502.16982 [cs.LG]。网址：https://arxiv.org/abs/2502.16982。 [45] Rui-Bo Liu 等人。“基于LLM和语音标记的令人信服的音频生成”。发表于：*2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP)*。IEEE。2024年，第591–595页。 [46] Songting Liu。“基于扩散变换器的零样本语音转换”。发表于：*arXiv preprint arXiv:2411.09943* (2024)。 [47] Steven R Livingstone 和 Frank A Russo。“Ryerson情感语音和歌曲的音视频数据库（RAVDESS）：一套动态、多模态的面部和声乐表达集，使用北美英语”。发表于：*PloS one* 13.5 (2018)，e0196391。 [48] Ilya Loshchilov 和 Frank Hutter。“解耦权重衰减正则化”。发表于：*arXiv preprint arXiv:1711.05101* (2017)。 [49] Yi Luo 和 Jianwei Yu。“带带分离RNN的音乐源分离”。发表于：*IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023)，第1893–1901页。 DOI: 10.1109/TASLP.2023.3271145。 [50] Linhan Ma 等人。“Wenetspeech4tts：用于大型语音生成模型基准的12,800小时普通话TTS语料库”。发表于：*arXiv preprint arXiv:2406.05763* (2024)。 [51] Irene Martín-Morató 和 Annamaria Mesaros。“什么是真实基础？多注释者数据在音频标记中的可靠性”。发表于：*2021 29th European Signal Processing Conference (EUSIPCO)*。IEEE。2021年，第76–80页。 [52] Xinhao Mei 等人。“Wavcaps：一个由ChatGPT辅助的弱标注音频字幕数据集，用于音频-语言多模态研究”。发表于：*IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2024)。 [53] Annamaria Mesaros、Toni Heittola 和 Tuomas Virtanen。“TUT声景分类和声音事件检测数据库”。发表于：*24th European Signal Processing Conference 2016 (EUSIPCO 2016)*。匈牙利布达佩斯，2016年。 [54] Vassil Panayotov 等人。“Librispeech：基于公共领域有声书的语音识别语料库”。发表于：*2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*。IEEE。2015年，第5206–5210页。

[55] Karol J. Piczak. “ESC: Dataset for Environmental Sound Classification”. In: *Proceedings of the 23rd Annual ACM Conference on Multimedia*. Brisbane, Australia: ACM Press, Oct. 13, 2015, pp. 1015–1018. ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806390. URL: http://dl.acm.org/citation.cfm?doid=2733373.2806390.

[56] Soujanya Poria et al. “Meld: A multimodal multi-party dataset for emotion recognition in conversations”. In: *arXiv preprint arXiv:1810.02508* (2018).

[57] Vineel Pratap et al. “MLS: A Large-Scale Multilingual Dataset for Speech Research”. In: *ArXiv* abs/2012.03411 (2020).

[58] Alec Radford et al. “Robust speech recognition via large-scale weak supervision”. In: *International conference on machine learning*. PMLR. 2023, pp. 28492–28518.

[59] Muhammad Mamunur Rashid, Guiqing Li, and Chengrui Du. “Nonspeech7k dataset: Classification and analysis of human non-speech sound”. In: *IET Signal Processing* 17.6 (2023), e12233.

[60] S Sakshi et al. “Mmau: A massive multi-task audio understanding and reasoning benchmark”. In: *arXiv preprint arXiv:2410.19168* (2024).

[61] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. “A dataset and taxonomy for urban sound research”. In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, pp. 1041–1044.

[62] Yao Shi et al. “Aishell-3: A multi-speaker mandarin tts corpus and the baselines”. In: *arXiv preprint arXiv:2010.11567* (2020).

[63] Changli Tang et al. “Salmonn: Towards generic hearing abilities for large language models”. In: *arXiv preprint arXiv:2310.13289* (2023).

[64] Zhiyuan Tang et al. “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.

[65] Kimi Team et al. *Kimi k1.5: Scaling Reinforcement Learning with LLMs*. 2025. arXiv: 2501.12599 [cs.AI]. URL: https://arxiv.org/abs/2501.12599.

[66] Teknium. *OpenHermes 2.5: An Open Dataset of Synthetic Data for Generalist LLM Assistants*. 2023. URL: https://huggingface.co/datasets/teknium/OpenHermes-2.5.

[67] Migel Tissera. *Synthia-70b-v1.2: Synthetic intelligent agent*. Hugging Face. 2023. URL: https://huggingface.co/migtissera/Synthia-13B.

[68] Samarth Tripathi, Sarthak Tripathi, and Homayoon Beigi. “Multi-modal emotion recognition on iemocap dataset using deep learning”. In: *arXiv preprint arXiv:1804.05788* (2018).

[69] Changhan Wang et al. “VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation”. In: *arXiv preprint arXiv:2101.00390* (2021).

[70] Chengyi Wang et al. “Neural codec language models are zero-shot text to speech synthesizers”. In: *arXiv preprint arXiv:2301.02111* (2023).

[71] Xiong Wang et al. “Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm”. In: *arXiv preprint arXiv:2411.00774* (2024).

[72] Zhifei Xie and Changqiao Wu. “Mini-omni: Language models can hear, talk while thinking in streaming”. In: *arXiv preprint arXiv:2408.16725* (2024).

[73] Jin Xu et al. “Qwen2. 5-omni technical report”. In: *arXiv preprint arXiv:2503.20215* (2025).

[74] Jin Xu et al. *Qwen2.5-Omni Technical Report*. 2025. arXiv: 2503.20215 [cs.CL]. URL: https://arxiv.org/abs/2503.20215.

[75] Zhangchen Xu et al. “Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing”. In: *arXiv preprint arXiv:2406.08464* (2024).

[76] An Yang et al. “Qwen2.5 Technical Report”. In: *arXiv preprint arXiv:2412.15115* (2024).

[77] Dongchao Yang et al. “Uniaudio: An audio foundation model toward universal audio generation”. In: *arXiv preprint arXiv:2310.00704* (2023).

[78] Pinci Yang et al. “Avqa: A dataset for audio-visual question answering on videos”. In: *Proceedings of the 30th ACM international conference on multimedia*. 2022, pp. 3480–3491.

[79] Zehui Yang et al. “Open source magicdata-ramc: A rich annotated mandarin conversational (ramc) speech dataset”. In: *arXiv preprint arXiv:2203.16844* (2022).

[80] Zhen Ye et al. “Llasa: Scaling Train-Time and Inference-Time Compute for Llama-based Speech Synthesis”. In: *arXiv preprint arXiv:2502.04128* (2025).

[55] Karol J. Piczak. “ESC：环境声音分类数据集”。发表于：*Proceedings of the 23rd Annual ACM Conference on Multimedia*。澳大利亚布里斯班：ACM出版社，2015年10月13日，第1015–1018页。ISBN：978-1-4503-3459-4。DOI：10.1145/2733373.2806390。网址：http://dl.acm.org/citation.cfm?doid= 2733373.2806390。[56] Soujanya Poria 等人。“Meld：用于对话中情感识别的多模态多方数据集”。发表于：*arXiv preprint arXiv:1810.02508* (2018)。[57] Vineel Pratap 等人。“MLS：用于语音研究的大规模多语种数据集”。发表于：*ArXiv* abs/2012.03411（2020年）。[58] Alec Radford 等人。“通过大规模弱监督实现稳健的语音识别”。发表于：*International conference on machine learning*。PMLR。2023年，第28492–28518页。[59] Muhammad Mamunur Rashid，Li Guiqing，Du Chengrui。“Nonspeech7k数据集：人类非语音声音的分类与分析”。发表于：*IET Signal Processing* 17.6（2023年），e12233。[60] S Sakshi 等人。“Mmau：一个庞大的多任务音频理解与推理基准”。发表于：*arXiv preprint arXiv:2410.19168* (2024)。[61] Justin Salamon，Christopher Jacoby，Juan Pablo Bello。“城市声音研究的数据集与分类体系”。发表于：*Proceedings of the 22nd ACM international conference on Multimedia*。2014年，第1041–1044页。[62] Yao Shi 等人。“Aishell-3：多说话人普通话TTS语料库及基线”。发表于：*arXiv preprint arXiv:2010.11567* (2020)。[63] Changli Tang 等人。“Salmonn：面向大型语言模型的通用听觉能力”。发表于：*arXiv preprint arXiv:2310.13289* (2023)。[64] Zhiyuan Tang 等人。“Kespeech：一个开源的普通话及其八个方言的语音数据集”。发表于：*Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*。2021年。[65] Kimi Team 等人。*Kimi k1.5: Scaling Reinforcement Learning with LLMs*。2025年。arXiv：2501.12599 [cs.AI]。网址：https://arxiv.org/abs/2501.12599。[66] Teknium。*OpenHermes 2.5: An Open Dataset of Synthetic Data for Generalist LLM Assistants*。2023年。网址：https://huggingface.co/datasets/teknium/OpenHermes-2.5。[67] Migel Tissera。*Synthia-70b-v1.2: Synthetic intelligent agent*。Hugging Face。2023年。网址：https://huggingface.co/migtissera/Synthia-13B。[68] Samarth Tripathi，Sarthak Tripathi，Homayoon Beigi。“基于深度学习的多模态情感识别在iemocap数据集上的应用”。发表于：*arXiv preprint arXiv:1804.05788* (2018)。[69] Changhan Wang 等人。“VoxPopuli：用于表征学习、半监督学习和解释的大规模多语种语音语料库”。发表于：*arXiv preprint arXiv:2101.00390* (2021)。[70] Chengyi Wang 等人。“神经编解码器语言模型是零样本文本到语音合成器”。发表于：*arXiv preprint arXiv:2301.02111* (2023)。[71] Xiong Wang 等人。“Freeze-omni：一种具有冻结大模型的智能低延迟语音对话模型”。发表于：*arXiv preprint arXiv:2411.00774* (2024)。[72] Zhifei Xie 和 Changqiao Wu。“Mini-omni：语言模型可以听、说，同时进行流式思考”。发表于：*arXiv preprint arXiv:2408.16725* (2024)。[73] Jin Xu 等人。“Qwen2.5-omni技术报告”。发表于：*arXiv preprint arXiv:2503.20215* (2025)。[74] Jin Xu 等人。*Qwen2.5-Omni Technical Report*。2025年。arXiv：2503.20215 [cs.CL]。网址：https://arxiv.org/abs/2503.20215。[75] Zhangchen Xu 等人。“Magpie：通过提示对齐的大模型从零开始合成对齐数据”。发表于：*arXiv preprint arXiv:2406.08464* (2024)。[76] An Yang 等人。“Qwen2.5技术报告”。发表于：*arXiv preprint arXiv:2412.15115* (2024)。[77] Dongchao Yang 等人。“Uniaudio：面向通用音频生成的音频基础模型”。发表于：*arXiv preprint arXiv:2310.00704* (2023)。[78] Pinci Yang 等人。“Avqa：用于视频中的音频视觉问答的数据集”。发表于：*Proceedings of the 30th ACM international conference on multimedia*。2022年，第3480–3491页。[79] Zehui Yang 等人。“开源魔术数据-ramc：一个丰富注释的普通话对话（ramc）语音数据集”。发表于：*arXiv preprint arXiv:2203.16844* (2022)。[80] Zhen Ye 等人。“Llasa：为Llama基础的语音合成扩展训练时间和推理时间的计算”。发表于：*arXiv preprint arXiv:2502.04128* (2025)。

[81] Jianwei Yu et al. “Autoprep: An automatic preprocessing framework for in-the-wild speech data”. In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2024, pp. 1136–1140.

[82] Jianwei Yu et al. “High fidelity speech enhancement with band-split rnn”. In: *arXiv preprint arXiv:2212.00406* (2022).

[83] Heiga Zen et al. “Libritts: A corpus derived from librispeech for text-to-speech”. In: *arXiv preprint arXiv:1904.02882* (2019).

[84] Aohan Zeng et al. “Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot”. In: *arXiv preprint arXiv:2412.02612* (2024).

[85] Binbin Zhang 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.

[86] Qinglin Zhang et al. “Omniflatten: An end-to-end gpt model for seamless voice conversation”. In: *arXiv preprint arXiv:2410.17799* (2024).

[87] Yu Zhang et al. “Google usm: Scaling automatic speech recognition beyond 100 languages”. In: *arXiv preprint arXiv:2303.01037* (2023).

[88] Hang Zhao et al. “MINT: Boosting Audio-Language Model via Multi-Target Pre-Training and Instruction Tuning”. In: *Interspeech 2024*. 2024, pp. 52–56. DOI: `10.21437/Interspeech.2024-1863`.

[89] Kun Zhou et al. “Emotional voice conversion: Theory, databases and ESD”. In: *Speech Communication* 137 (2022), pp. 1–18.

[81] 俞建伟等。“Autoprep：一种用于野外语音数据的自动预处理框架”。发表于：*ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*。IEEE。2024，页码1136–1140。 [82] 俞建伟等。“带带分离RNN的高保真语音增强”。发表于：*arXiv preprint arXiv:2212.00406* (2022)。 [83] 禅平 Heiga 等。“Libritts：一个源自LibriSpeech的文本转语音语料库”。发表于：*arXiv preprint arXiv:1904.02882* (2019)。 [84] 曾奥汉等。“Glm-4-voice：迈向智能且类人端到端语音聊天机器人”。发表于：*arXiv preprint arXiv:2412.02612* (2024)。 [85] 张彬彬等。“Wenetspeech：一个包含10000+小时多领域普通话语料库，用于语音识别”。发表于：*ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*。IEEE。2022，页码6182–6186。 [86] 张青林等。“Omniflatten：一种端到端的GPT模型，用于无缝语音对话”。发表于：*arXiv preprint arXiv:2410.17799* (2024)。 [87] 张宇等。“Google USM：将自动语音识别扩展到超过100种语言”。发表于：*arXiv preprint arXiv:2303.01037* (2023)。 [88] 赵航等。“MINT：通过多目标预训练和指令调优提升音频-语言模型”。发表于：*Interspeech 2024*。2024，页码52–56。DOI：10.21437/Interspeech.2024-1863。 [89] 周坤等。“情感语音转换：理论、数据库与ESD”。发表于：*Speech Communication* 137（2022），页码1–18。

Appendix

A Contributions

Core Contributors

Ding Ding
Zeqian Ju
Yichong Leng
Songxiang Liu
Tong Liu
Zeyu Shang
Kai Shen
Wei Song
Xu Tan[#]
Heyi Tang
Zhengtao Wang
Chu Wei
Yifei Xin
Xinran Xu
Jianwei Yu
Yutao Zhang
Xinyu Zhou[#]

Contributors

Y. Charles
Jun Chen
Yanru Chen
Yulun Du
Weiran He
Zhenxing Hu
Guokun Lai
Qingcheng Li
Yangyang Liu
Weidong Sun
Jianzhou Wang
Yuzhi Wang
Yuefeng Wu
Yuxin Wu
Dongchao Yang
Hao Yang
Ying Yang
Zhilin Yang
Aoxiong Yin
Ruibin Yuan
Yutong Zhang
Zaida Zhou

[#] Project leads.
The contributor list is in alphabetical order based on their last names.

附录

A 贡献

核心贡献者

丁丁泽谦聚一
冲冷松香刘通
刘泽宇尚凯申
伟宋旭谭#贺
毅唐正涛王楚
伟易飞新新然
许建伟于玉涛
张新宇周#

贡献者

Y. Charles Jun
Chen Yanru Ch
en Yulun Du W
eiran He Zhenxi
ng Hu Guokun
Lai Qingcheng
Li Yangyang Li
u Weidong Sun
Jianzhou Wang
Yuzhi Wang Yu
efeng Wu Yuxi
n Wu Dongchao
Yang Hao Yang
Ying Yang Zhil
in Yang Aoxion
g Yin Ruibin Y
uan Yutong Zha
ng Zaida Zhou

# Project leads.
The contributor list is in alphabetical order based on their last names.