Qwen2-Audio: V2T

1. Qwen2-Audio

Qwen2-Audio-7B 🤖 | 🤗
Qwen-Audio-7B-Instruct 🤖 | 🤗
Demo 🤖 | 🤗
📑 Paper | 📑 Blog | 💬 WeChat (微信) | Discord

1.1 Introduction

Qwen2-Audio, a Large Audio-Language Model (LALM) designed to process and interpret both audio and text inputs. Key novel elements include:

1. **Scaled-Up Training & Simplified Pre-training**
   - Significantly expands the dataset compared to earlier models.
   - Reduces complexity by directly using natural language prompts for various data types and tasks, narrowing the gap between pre-training and post-training.
2. **Enhanced Instruction-Following**
   - Employs instruction tuning and direct preference optimization to align outputs with human preferences.
3. **Dual Interaction Modes Without Explicit Switching**
   - **Audio Analysis Mode**: Handles a wide range of audio types (speech, sound, music, mixed audio) and autonomously detects command segments within the audio or text.
   - **Voice Chat Mode**: Functions as a conversational agent, allowing unrestricted dialogue via audio or text.
4. **State-of-the-Art Performance**
   - Outperforms previous LALMs on Aishell2, FLUERS-zh, VocalSound, and the AIR-Bench chat benchmark—demonstrating robust, multi-domain audio understanding and interaction.

Overall, **Qwen2-Audio** achieves stronger instruction-following capabilities and more flexible audio interaction modes, pushing the boundaries of audio-language modeling.

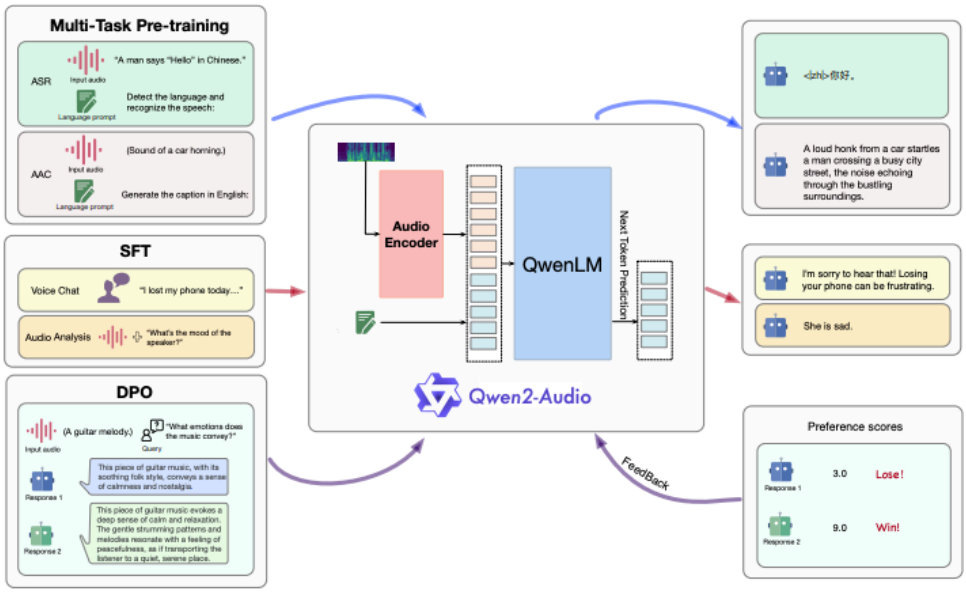

Figure 2: The overview of three-stage training process of Qwen2-Audio.

1.2 Methodology

1. **Model Architecture**
   - **Audio Encoder**: Initialized from the Whisper-large-v3 model, audio is resampled to 16kHz, and converted into 128-channel mel-spectrograms. Incorporates a pooling layer to reduce sequence length, so each output frame corresponds to ~40ms of the original audio.
   - **Language Model**: Uses Qwen-7B as the core LLM.
   - **Overall**: Qwen2-Audio combines the audio encoder (~1.2B parameters) and Qwen-7B for a total of 8.2B parameters. The training objective maximizes the next text token probability conditioned on both audio representations and preceding text tokens.
2. **Pre-training**
   - Natural Language Prompts replace hierarchical tagging (Chu et al., 2023).
   - This approach improves generalization and aligns pre-training more closely with post-training usage, boosting instruction-following abilities.
3. **Supervised Fine-tuning (SFT)**
   - Builds on the thoroughly pre-trained base, further aligning the model with human intent using high-quality, carefully curated data.
   - Two Interaction Modes—both included in a single, unified training pipeline:
     i. Audio Analysis: Offline analysis of diverse audio inputs, with commands provided via audio or text.
     ii. Voice Chat: Conversational agent for real-time, open-ended voice queries.
   - Users do not need to switch modes manually; the model automatically handles both.
4. [Direct Preference Optimization (DPO)]
   - Used after supervised fine-tuning to refine the model's responses according to human preference data.
   - Aims to boost factuality and desirable behavior by comparing "good" vs. "bad" answers in a triplet dataset.

Direct Preference Optimization We employ DPO (Rafailov et al., 2024) to further optimize models to follow human preferences. By obtaining the dataset $\mathcal{D}$ with the triplet data $(\boldsymbol{x}, \boldsymbol{y_w}, \boldsymbol{y_l})$, where $\boldsymbol{x}$ is the input sequence with input audio, and $\boldsymbol{y_w}$ and $\boldsymbol{y_l}$ are the human-annotated good and bad responses respectively, we optimize the model $\mathcal{P}_\theta$ as follows:

$$\mathcal{L}_{\text{DPO}}(\mathcal{P}_\theta; \mathcal{P}_{\text{ref}}) = -\mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y_w}, \boldsymbol{y_l}) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\mathcal{P}_\theta(\boldsymbol{y_w} \mid \boldsymbol{x})}{\mathcal{P}_{\text{ref}}(\boldsymbol{y_w} \mid \boldsymbol{x})} - \beta \log \frac{\mathcal{P}_\theta(\boldsymbol{y_l} \mid \boldsymbol{x})}{\mathcal{P}_{\text{ref}}(\boldsymbol{y_l} \mid \boldsymbol{x})} \right) \right], \tag{2}$$

where $\mathcal{P}_{\text{ref}}$ denotes the reference model initialized with $\mathcal{P}_\theta$, σ represents sigmoid function and β is a hyperparameter. Figure 2 illustrates the three-stage training process of Qwen2-Audio.

1.3 Evaluation

- Testing on AIR-Bench, which better reflects actual user experiences compared to traditional SLU/SER datasets.
- They also conduct a comprehensive assessment across 13 datasets covering multiple tasks—ASR, Speech-to-Text Translation (S2TT), Speech Emotion Recognition (SER), and Vocal Sound Classification (VSC). All evaluation data is carefully excluded from the training set.

Table 1: Summary of Evaluation Benchmarks for Qwen2-Audio.

Task	Description	Dataset	Split	Metric
ASR	Automatic Speech Recognition	Fleurs (Conneau et al., 2022) Aishell2 (Du et al., 2018) Librispeech (Panayotov et al., 2015) Common Voice (Ardila et al., 2020)	dev | test test dev | test dev | test	WER
S2TT	Speech-to-Text Translation	CoVoST2 (Wang et al., 2020)	test	BLEU [1] (Papineni et al., 2002)
SER	Speech Emotion Recognition	Meld (Poria et al., 2019)	test	ACC
VSC	Vocal Sound Classification	VocalSound (Gong et al., 2022)	test	ACC
AIR-Bench (Yang et al., 2024)	Chat-Benchmark-Speech	Fisher (Cieri et al., 2004) SpokenWOZ (Si et al., 2023) IEMOCAP (Si et al., 2023) Common voice (Ardila et al., 2020)	dev | test	GPT-4 Eval
	Chat-Benchmark-Sound	Clotho (Drossos et al., 2020)	dev | test	GPT-4 Eval
	Chat-Benchmark-Music	MusicCaps (Agostinelli et al., 2023)	dev | test	GPT-4 Eval
	Chat-Benchmark-Mixed-Audio	Common voice (Ardila et al., 2020) AudioCaps (Kim et al., 2019) MusicCaps (Agostinelli et al., 2023)	dev | test	GPT-4 Eval

Table 2: The results of Automatic Speech Recognition (ASR), Speech-to-Text Translation (S2TT), Speech Emotion Recognition (SER), Vocal Sound Classification (VSC), and AIR-Bench chat benchmark. Note that for Qwen2-Audio, the results for Fleurs are zero-shot, whereas the results for Common Voice are not zero-shot.

Task	Dataset	Model	Performance	
			Metrics	Results
ASR	**Librispeech** *dev-clean | dev-other | test-clean | test-other*	SpeechT5 (Ao et al., 2021) SpeechNet (Chen et al., 2021) SLM-FT (Wang et al., 2023b) SALMONN (Tang et al., 2024) SpeechVerse (Das et al., 2024) Qwen-Audio (Chu et al., 2023) Qwen2-Audio	WER ↓	2.1 | 5.5 | 2.4 | 5.8 - | - | 30.7 | - - | - | 2.6 | 5.0 - | - | 2.1 | 4.9 - | - | 2.1 | 4.4 1.8 | 4.0 | 2.0 | 4.2 **1.3** | **3.4** | **1.6** | **3.6**
	Common Voice 15 *en | zh | yue | fr*	Whisper-large-v3 (Radford et al., 2023) Qwen2-Audio	WER ↓	9.3 | 12.8 | 10.9 | 10.8 **8.6** | **6.9** | **5.9** | **9.6**
	Fleurs *zh*	Whisper-large-v3 (Radford et al., 2023) Qwen2-Audio	WER ↓	7.7 **7.5**
	Aishell2 *Mic | iOS | Android*	MMSpeech-base (Zhou et al., 2022) Paraformer-large (Gao et al., 2023) Qwen-Audio (Chu et al., 2023) Qwen2-Audio	WER ↓	4.5 | 3.9 | 4.0 - | **2.9** | - 3.3 | 3.1 | 3.3 **3.0** | 3.0 | **2.9**
S2TT	**CoVoST2** *en-de | de-en | en-zh | zh-en*	SALMONN (Tang et al., 2024) SpeechLLaMA (Wu et al., 2023a) BLSP (Wang et al., 2023a) Qwen-Audio (Chu et al., 2023) Qwen2-Audio	BLEU ↑	18.6 | - | 33.1 | - - | 27.1 | - | 12.3 14.1 | - | - | - 25.1 | 33.9 | 41.5 | 15.7 **29.9** | **35.2** | **45.2** | **24.4**
	CoVoST2 *es-en | fr-en | it-en |*	SpeechLLaMA (Wu et al., 2023a) Qwen-Audio (Chu et al., 2023) Qwen2-Audio	BLEU ↑	27.9 | 25.2 | 25.9 39.7 | **38.5** | 36.0 **40.0** | **38.5** | **36.3**
SER	**Meld**	WavLM-large (Chen et al., 2022) Qwen-Audio (Chu et al., 2023) Qwen2-Audio	ACC ↑	0.542 **0.557** 0.553
VSC	**VocalSound**	CLAP (Elizalde et al., 2022) Pengi (Deshmukh et al., 2023) Qwen-Audio (Chu et al., 2023) Qwen2-Audio	ACC ↑	0.4945 0.6035 0.9289 **0.9392**
AIR-Bench (Yang et al., 2024)	**Chat Benchmark** *Speech | Sound | Music | Mixed-Audio*	SALMONN (Tang et al., 2024) BLSP (Wang et al., 2023a) Pandagpt (Su et al., 2023) Macaw-LLM (Lyu et al., 2023) SpeechGPT (Zhang et al., 2023) Next-gpt (Wu et al., 2023b) Qwen-Audio (Chu et al., 2023) Gemini-1.5-pro (Reid et al., 2024) Qwen2-Audio	GPT-4 ↑	6.16 | 6.28 | 5.95 | 6.08 6.17 | 5.55 | 5.08 | 5.33 3.58 | 5.46 | 5.06 | 4.25 0.97 | 1.01 | 0.91 | 1.01 1.57 | 0.95 | 0.95 | 4.13 3.86 | 4.76 | 4.18 | 4.13 6.47 | 6.95 | 5.52 | 6.08 6.97 | 5.49 | 5.06 | 5.27 **7.18** | **6.99** | **6.79** | **6.77**

2. Qwen-Audio [WIP]

Qwen-Audio 🤖 | 🤗 | Qwen-Audio-Chat 🤖 | 🤗 | Demo 🤖 | 🤗

Homepage | Paper | WeChat | Discord

The authors introduce Qwen-Audio, a large-scale audio–language model designed to handle diverse tasks, multiple languages, and a variety of audio types (e.g., human speech, natural sounds, music).

- Qwen-Audio is trained on over 30 tasks spanning at least eight languages.
- A core innovation is a multi-task training framework that leverages a sequence of hierarchical tags in the decoder, mitigating the “one-to-many” problem posed by differing textual labels (e.g., annotation granularity, structured vs. unstructured labels) across datasets.
- A significant challenge of multi-task and multi-dataset co-training arises from the considerable variation in textual labels associated with different datasets. This variation stems from differences in task objectives, languages, annotation granularity, and text structure (structured or unstructured). To address this one-to-many challenge, we have carefully designed a multi-task training framework that conditions the decoder on a sequence of hierarchical tags. This design encourages knowledge sharing and helps mitigate interference through shared and specified tags, respectively
- Additionally, the model incorporates speech recognition with word-level time-stamp prediction (SRWT)—a rarely addressed component in previous work—to enhance both ASR accuracy and grounding-based tasks (e.g., question answering for music or natural sounds).

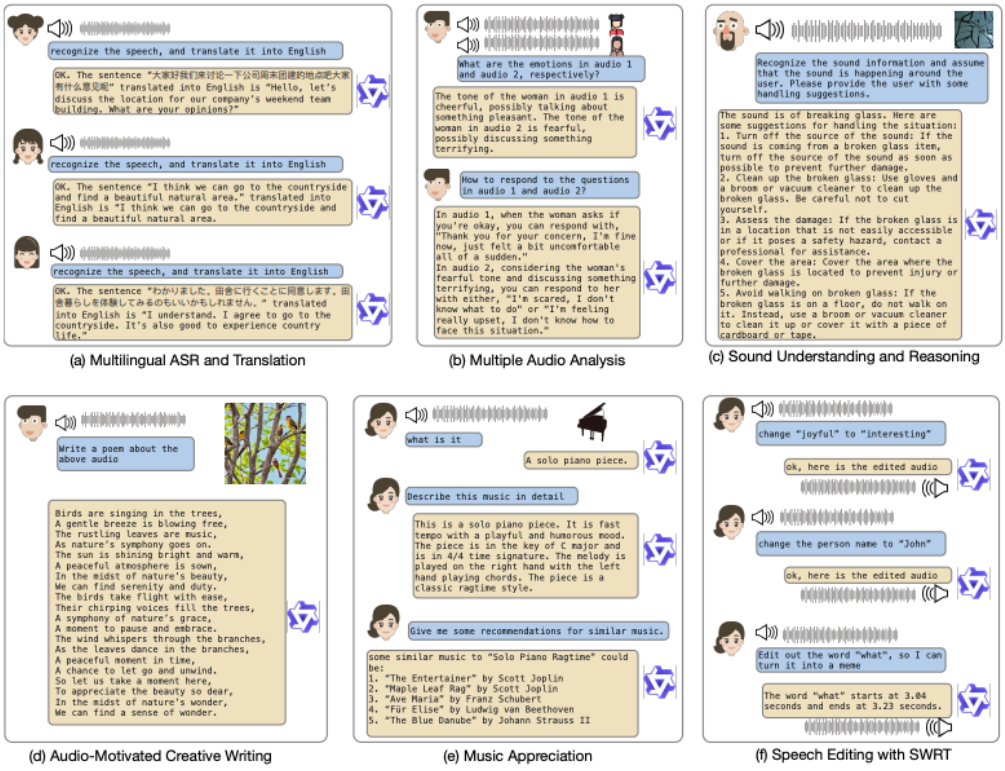

Figure 2: Examples of Qwen-Audio showcasing its proficiency in perceiving and comprehending various types of audio. Qwen-Audio supports multiple-audio analysis, sound understanding and reasoning, music appreciation, and tool usage for speech editing. Demos are available at https://qwen-audio.github.io/Qwen-Audio/.

2 Related Work

Prior efforts in multi-task audio–text learning have largely targeted **specific audio types** (e.g., human speech) and **narrow sets of tasks** (e.g., recognition, translation), often overlooking natural sounds or music. Works such as SpeechNet, SpeechT5, and Whisper have unified certain human speech tasks by sharing encoder–decoder frameworks or adopting decoder-only

Transformers. However, they still fall short in handling non-speech audio and in integrating diverse task requirements (e.g., different output granularities).

In parallel, **LLM-based multimodal approaches** (e.g., AudioGPT, HuggingGPT) often rely on **external audio tools**, limiting direct, end-to-end learning of audio features such as prosody or non-verbal cues. Recent models like SpeechGPT, BLSP, and LTU have begun bridging speech inputs with LLMs more directly but remain focused on single types of audio or narrower instruction datasets.

Qwen-Audio addresses these gaps by using a single encoder for various audio signals—spanning human speech, natural sounds, music, and songs—and performing large-scale end-to-end training. This unified approach supports diverse tasks (e.g., speech recognition, grounding, audio captioning) without specialized architectural tweaks, thereby expanding the range of audio–text interactions that an LLM can handle and improving performance across a variety of benchmarks.

3 Methodology

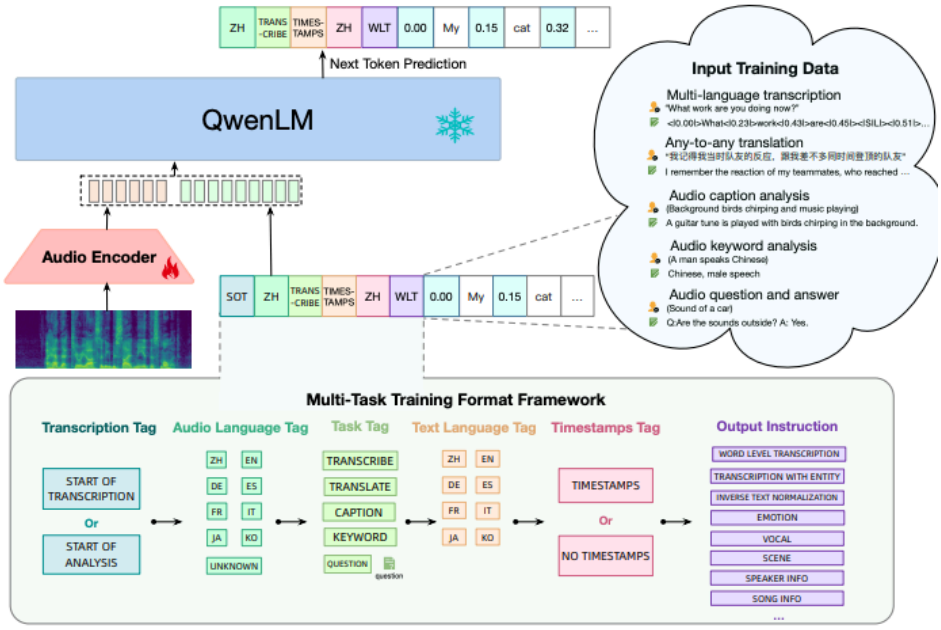

Figure 3: The overview of Qwen-Audio architecture and multitask-pretraining.

3.1 Model Architecture

Qwen-Audio contains an audio encoder and a large language model.

Given the paired data (a, x), where the a and x denote the audio sequences and text sequences, the training objective is to maximize the next text token probability as:

$$\mathcal{P}_\theta(x_t|\boldsymbol{x}_{<t}, \text{Encoder}_\phi(\boldsymbol{a})),$$

Audio Encoder

- Initialized from Whisper-large-v2 model (640M)
- Takes 16kHz audio and converts the raw waveform into 80-channel melspectrogram using a window size of 25ms and a hop size of 10ms.
- Additionally, a pooling layer with a stride of two is incorporated to reduce the length of the audio representation. As a result, each frame of the encoder output approximately corresponds to a 40ms segment of the original audio signal.
- SpecAugment (Park et al.) is applied at the training time as data augmentation.

Large Language Model: The model is initialized using pre-trained weights derived from Qwen-7B.

3.2 Multitask Pretraining

Qwen-Audio unifies diverse audio datasets and tasks (e.g., speech recognition, translation, captioning, and QA) under a single co-training framework. This approach aims to share knowledge across related tasks—where lower-level perceptual tasks can boost higher-level reasoning tasks—while avoiding interference from mismatched text labels, languages, and annotation structures.

To address the “one-to-many” challenge of combining heterogeneous datasets, the authors adopt a multi-task training format using hierarchical tags:

1. **Transcription Tag**
   - Indicates whether the task is speech transcription (<|startoftranscripts|>) vs. analysis (<|startofanalysis|>).
2. **Audio Language Tag**
   - Specifies the spoken language (from eight possible languages) or <|unknown|> if the audio contains no speech.
3. **Task Tag**
   - Defines the task category: <|transcribe|>, <|translate|>, <|caption|>, <|analysis|>, or <|question-answer|>.
   - For QA tasks, the question is appended directly after this tag.
4. **Text Language Tag**
   - Informs the model of the desired output language.
5. **Timestamps Tag**
   - Determines if **word-level time stamps** (<|timestamps|>) must be predicted for speech recognition (the SRWT task), which significantly improves the model’s audio grounding and QA performance.
6. **Output Instruction**
   - Provides more detailed guidance on the desired output format or structure.

By leveraging shared tags for similar tasks and using clearly distinct tags for differing tasks, Qwen-Audio maximizes knowledge sharing while preventing label overlap.

This design distinguishes it from previous models (e.g., Whisper), which focus mainly on speech recognition and translation. Crucially, word-level timestamps (SRWT) go beyond sentence-level timestamps, enhancing both ASR accuracy and grounding-based QA. The end result is a unified model that does not require separate architectures or fine-tuned checkpoints for different audio tasks.

Table 1: Multi-task pre-training dataset.

Types	Task	Description	Hours
Speech	ASR	Automatic speech recognition (multiple languages)	30k
	S2TT	Speech-to-text translation	3.7k
	OSR	Overlapped speech recognition	<1k
	Dialect ASR	Automatic dialect speech recognition	2k
	SRWT	English speech recognition with word-level timestamps	10k
		Mandarin speech recognition with word-level timestamps	11k
	DID	Dialect identification	2k
	LID	Spoken language identification	11.7k
	SGC	Speaker gender recognition (biologically)	4.8k
	ER	Emotion recognition	<1k
	SV	Speaker verification	1.2k
	SD	Speaker diarization	<1k
	SER	Speech entity recognition	<1k
	KS	Keyword spotting	<1k
	IC	Intent classification	<1k
	SF	Slot filling	<1k
	SAP	Speaker age prediction	4.8k
	VSC	Vocal sound classification	<1k
Sound	AAC	Automatic audio caption	8.4k
	SEC	Sound event classification	5.4k
	ASC	Acoustic scene classification	<1k
	SED	Sound event detection with timestamps	<1k
	AQA	Audio question answering	<1k
Music&Song	SID	Singer identification	<1k
	SMER	Singer and music emotion recognition	<1k
	MC	Music caption	25k
	MIC	Music instruments classification	<1k
	MNA	Music note analysis such as pitch, velocity	<1k
	MGR	Music genre recognition	9.5k
	MR	Music recognition	<1k
	MQA	Music question answering	<1k

3.3 Supervised Fine-tuning

After multi-task pretraining, the authors further fine-tune the model via instruction-based methods to ensure it aligns closely with human intent. This yields Qwen-Audio-Chat, an interactive model capable of handling audio and text inputs in multi-turn dialogues. Key points include:

1. **Instruction Data Creation**
   - **Manual demonstrations**: Raw text labels, Q&A pairs for each task.
   - **GPT-3.5 augmentation**: Additional questions and answers automatically generated from the raw labels.
   - **Audio-dialogue data**: Mixture of **manual annotations**, **model-generated content**, and **strategically concatenated** dialogues. These examples incorporate **reasoning**, **story generation**, and **multi-audio** comprehension.
2. **Dialogue Format**
   - Uses **ChatML** style: each turn is enclosed by `<im_start>` and `<im_end>` tokens.
   - Audio inputs labeled as **"Audio id:"**, enabling **multi-audio** dialogues (e.g., "Audio 1," "Audio 2," etc.).
   - Includes both **audio-centric** and **pure text** instructions, so the model can handle either modality in **multi-turn** interactions.
3. **Outcome**
   - The final **Qwen-Audio-Chat** model seamlessly accepts **diverse inputs**—multiple audios, text queries, or both—while **maintaining** robust conversation capabilities.
   - A **20k**–example dataset underpins this instruction fine-tuning, ensuring **broad coverage** of tasks and interaction scenarios.

The Data Format Example of Supervised Fine-Tuning.

<im_start>user
Audio 1: <audio>emov-db/141-168-0155.wav</audio>what does the speaker say?<im_end>
<im_start>assistant
The speaker says in English, "Won't you draw up, gentlemen.".<im_end>
<im_start>user
What's the mood of the person?<im_end>
<im_start>assistant
Based on the voice, the mood of the person is disgusted.<im_end>

In order to facilitate versatile input from both audio and pure text modalities within multi-turn dialogues, we use a combination of audio-centric instruction data mentioned above and pure text instruction data during this training process. This approach allows the model to handle diverse forms of input seamlessly. The total amount of instruction tuning data is 20k.

A Hyperparameters

We report the detailed training hyperparameter settings of Qwen-Audio in Table 6.

Table 6: Training hyperparameters of Qwen-Audio

Configuration	Multi-task Pre-training	Supervised Fine-tuning
Audio encoder init.	Whisper-large-v2	Qwen-audio 1st-stage
LLM init.	Qwen-7B	Qwen-7B
SpecAugment Policy	LibriSpeech Basic	LibriSpeech Basic
Optimizer	AdamW	AdamW
Optimizer hyperparameter	β_1=0.9, β_2=0.98, $eps = 1e^{-6}$	
Peak learning rate	$5e^{-5}$	$1e^{-5}$
Minimum learning rate	$1e^{-5}$	$1e^{-6}$
Audio encoder learning rate decay	0.95	0
Learning rate schedule	cosine decay	cosine decay
Weight decay	0.05	0.05
Gradient clip	1.0	1.0
Training steps	500k	8k
Warm-up steps	2000	3k
Global batch size	120	128
Gradient Acc.	1	8
Numerical precision	`bfloat16`	`bfloat16`
Optimizer sharding	✓	✓
Activation checkpointing	✗	✗
Model parallelism	✗	2
Pipeline parallelism	✗	✗