

LLaMA-Omni 2: LLM-based Real-time Spoken Chatbot with Autoregressive Streaming Speech Synthesis

Qingkai Fang^{1,3}, Yan Zhou^{1,3}, Shoutao Guo^{1,3}, Shaolei Zhang^{1,3}, Yang Feng^{1,2,3*}

¹Key Laboratory of Intelligent Information Processing

Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)

²Key Laboratory of AI Safety, Chinese Academy of Sciences

³University of Chinese Academy of Sciences, Beijing, China

{fangqingkai21b, fengyang}@ict.ac.cn

Abstract

Real-time, intelligent, and natural speech interaction is an essential part of the next-generation human-computer interaction. Recent advancements have showcased the potential of building intelligent spoken chatbots based on large language models (LLMs). In this paper, we introduce LLaMA-Omni 2, a series of speech language models (SpeechLMs) ranging from 0.5B to 14B parameters, capable of achieving high-quality real-time speech interaction. LLaMA-Omni 2 is built upon the Qwen2.5 series models, integrating a speech encoder and an autoregressive streaming speech decoder. Despite being trained on only 200K multi-turn speech dialogue samples, LLaMA-Omni 2 demonstrates strong performance on several spoken question answering and speech instruction following benchmarks, surpassing previous state-of-the-art SpeechLMs like GLM-4-Voice, which was trained on millions of hours of speech data.¹

1 Introduction

Speech, as a critical interface for human-computer interaction, can significantly enhance both interaction efficiency and user experience (Clark et al., 2019). In recent years, as large language models (LLMs) like ChatGPT (OpenAI, 2022) have demonstrated outstanding performance across various fields, speech interactions with LLMs have attracted widespread attention from both academia and industry. For instance, GPT-4o (OpenAI, 2024) enables *real-time*, *intelligent*, and *natural* speech interaction between users and LLMs, heralding the advent of a new generation of human-computer interaction paradigms.

To develop a spoken chatbot similar to GPT-4o, the traditional approach typically employs a cascaded pipeline comprising an automatic speech

recognition (ASR) model, an LLM, and a text-to-speech (TTS) model. While this method is relatively straightforward to implement, it suffers from several notable limitations. First, errors can accumulate across the different stages of the pipeline. Second, the overall response latency tends to be high due to the sequential processing of multiple models. Third, the system struggles to capture paralinguistic information present in the input speech. To address these limitations, end-to-end speech language models (SpeechLMs) have gradually gained more attention, using a single unified model to handle the entire process from speech input to output. Overall, end-to-end SpeechLMs can be categorized into two types: *native* and *modular*. Native SpeechLMs typically discretize speech into tokens and employ a GPT-style decoder-only Transformer (Radford, 2018) to model both speech and text within a unified language model (Zhang et al., 2023; Rubenstein et al., 2023; Hassid et al., 2024a). A key advantage of this architecture is its ability to leverage vast amounts of unsupervised speech data for pretraining, making it easier to scale up in terms of model parameters and data size. This can potentially result in emergent capabilities, such as more human-like speech expressiveness (Zeng et al., 2024a; Open-Moss, 2025). However, native SpeechLMs typically require large-scale speech datasets (e.g., millions of hours) for pretraining (Zeng et al., 2024b; Défossez et al., 2024), which presents challenges in data collection and training costs, and may also lead to catastrophic forgetting of the model’s text capabilities. In contrast, modular SpeechLMs incorporate a speech encoder and a speech decoder around the LLM to handle speech understanding and generation (Fang et al., 2025; Wang et al., 2024). The advantage of this approach is its ability to leverage the inherent capabilities of each module, requiring only small-scale fine-tuning (e.g., a few hundred or thousand hours of speech data) to align the mod-

*Corresponding author: Yang Feng.

¹Code: <https://github.com/ictnlp/LLaMA-Omni2>
Audio Samples: <https://llama-omni2.github.io/>

ules. This enables the model to acquire speech interaction capabilities at a relatively low cost, while retaining most of its original capability. Moreover, modular SpeechLMs can typically generate speech guided by textual output, ensuring the intelligence of the generated speech.

In addition to the intelligence of speech, real-time responsiveness and naturalness are also crucial characteristics of spoken chatbots. LLaMA-Omni (Fang et al., 2025) uses a non-autoregressive (NAR) streaming speech decoder to enable synchronized generation of speech and text, ensuring extremely low response latency. However, due to the limitations of non-autoregressive models in modeling capacity, the generated speech is often less natural and fluent. Freeze-Omni (Wang et al., 2024) combines both NAR and autoregressive (AR) models for speech generation, resulting in higher naturalness of the generated speech. However, it can only achieve sentence-level streaming speech generation through a simple sentence-split strategy, which prevents it from achieving very low response latency. To address these challenges, in this paper, we introduce LLaMA-Omni 2, a series of modular SpeechLMs ranging from 0.5B to 14B. LLaMA-Omni 2 adopts Qwen2.5-0.5B/1.5B/3B/7B/14B-Instruct models (Team, 2024) as the base LLM, and uses Whisper’s encoder (Radford et al., 2023) as the speech encoder. For the speech decoder, inspired by the state-of-the-art streaming speech synthesis model CosyVoice 2 (Du et al., 2024), it first includes an autoregressive text-to-speech language model initialized with Qwen2.5-0.5B, which generates speech tokens from the LLM output and achieves streaming generation through alternating read and write operations. The speech tokens are then passed through a chunk-aware causal flow matching model (Lipman et al., 2023) to generate the mel spectrogram in a streaming manner. To train the model, we synthesize 200K multi-turn speech-to-speech dialogue samples with diverse input voices and a uniform output voice. Experimental results show that LLaMA-Omni 2 achieves outstanding performance on spoken question answering and speech instruction following tasks in both speech-to-text and speech-to-speech settings, outperforming both LLaMA-Omni and the native SpeechLM GLM-4-Voice (Zeng et al., 2024a), which was trained on millions of hours of speech data. We also conducted detailed ablation studies on factors such as LLM parameter size, training data scale, speech decoder pretraining, and

read-write strategy, to better understand the impact of these factors on the overall system performance.

2 Model: LLaMA-Omni 2

In this section, we introduce the model architecture of LLaMA-Omni 2. As shown in Figure 1, the core of LLaMA-Omni 2 is an LLM, for which we use the Qwen2.5 series models (Team, 2024) due to their strong performance across various benchmarks. Next, we will describe how we equip the LLM with speech understanding and streaming speech generation capabilities. In the following, we use \mathcal{M}_{LLM} to denote the LLM. For a single-turn instruction-response pair, we denote the speech instruction as X , and the text and speech responses as Y^T and Y^S , respectively.

2.1 Speech Understanding

To enable speech understanding, we incorporate a speech encoder and a speech adapter before the LLM, similar to LLaMA-Omni (Fang et al., 2025). Specifically, we use the encoder of Whisper-large-v3 (Radford et al., 2023) as the speech encoder, which converts the input speech into a sequence of representations. The encoded representations are then passed into the speech adapter, which consists of a downsampling module and a feed-forward network (FFN). The downsampling module concatenates every k consecutive frames along the feature dimension, and the concatenated representations are further encoded by the FFN. The final output representation is then input into the LLM.

2.2 Streaming Speech Generation

To equip the model with streaming speech generation capabilities, we adopt a paradigm similar to CosyVoice 2 (Du et al., 2024). First, the speech response is converted into discrete tokens using a supervised semantic speech tokenizer. Then, an autoregressive text-to-speech language model is employed to model the streaming generation from the LLM output to speech tokens. Finally, a causal flow matching model converts speech tokens into the mel spectrogram in a streaming manner.

Speech Tokenizer The speech tokenizer is implemented by inserting a finite scalar quantization (FSQ) module (Mentzer et al., 2024) into the encoder of SenseVoice-Large ASR model (An et al., 2024). This module first projects the intermediate representations to a low-rank space and discretizes them through a rounding operation. Ultimately,

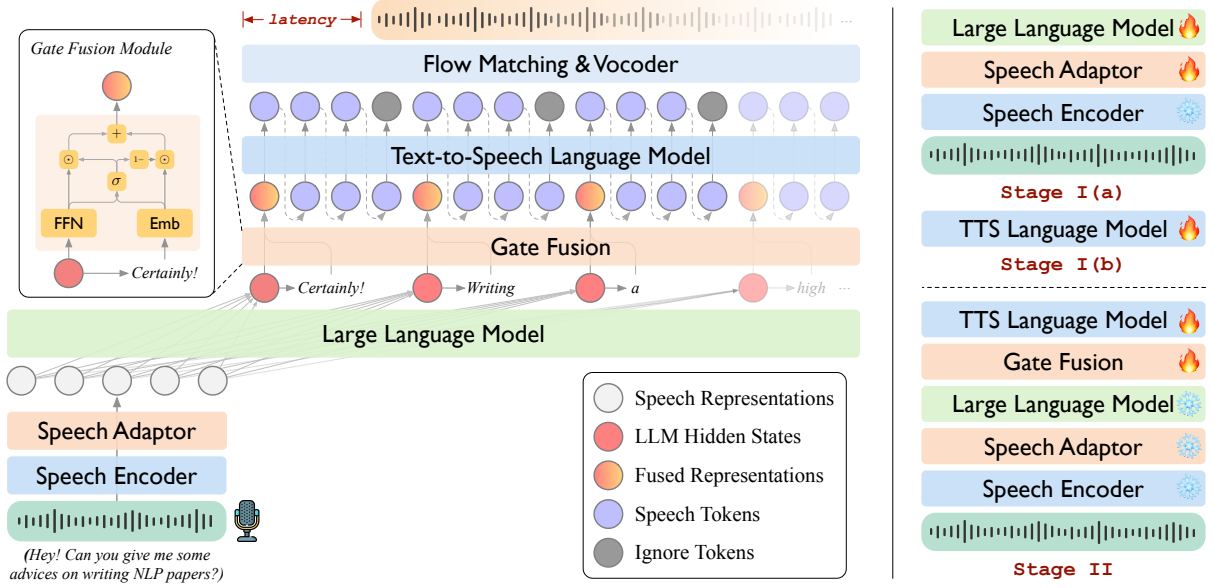


Figure 1: **Left:** Model architecture of LLaMA-Omni 2. **Right:** Illustration of the two-stage training strategy.

the speech response Y^S is converted into a token sequence $Y^U = [y_1^U, \dots, y_M^U]$, with 25 tokens per second, where each token $y_i^U \in \{K \in \mathbb{N} \mid 0 \leq K < 6561\}$. We use the pretrained speech tokenizer in CosyVoice 2.

Text-to-Speech Language Model After converting the speech response into discrete tokens, we use a decoder-only Transformer (Vaswani, 2017) to model the conditional language model from the LLM output to the speech tokens, denoted as \mathcal{M}_{TTS} . It is initialized with Qwen2.5-0.5B, and its vocabulary is extended as $\mathbb{V}' = \mathbb{V} \cup \{< i > \mid i \in \mathbb{N}, 0 \leq i < 6561\}$, where \mathbb{V} is the original vocabulary. This extension enables the model to generate speech tokens.

The input to \mathcal{M}_{TTS} comes from the output of the LLM. Specifically, the LLM output consists of two parts: *continuous hidden states* and *text tokens sampled from the hidden states*. The former contains contextual information, while the latter provides precise textual content. We aim to use both as inputs to the text-to-speech language model. This allows the model to both consider the current context and ensure better alignment with the text response when generating speech tokens. During training, the LLM is trained with teacher forcing, so its output hidden states are denoted as $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_N]$, where $\mathbf{h}_i = \mathcal{M}_{\text{LLM}}(X, Y_{<i}^T)$. The corresponding text is the ground truth $Y^T = [y_1^T, \dots, y_N^T]$. We first use a 2-layer feed-forward network (FFN) to map the hidden states to the embedding dimension of

\mathcal{M}_{TTS} , while also obtaining the text embeddings:

$$\mathbf{e}_i^{\text{hidden}} = \text{FFN}(\mathbf{h}_i), \quad (1)$$

$$\mathbf{e}_i^{\text{emb}} = \text{Emb}(y_i^T), \quad (2)$$

where $\text{Emb}(\cdot)$ is the embedding layer of \mathcal{M}_{TTS} . Afterward, we use an element-wise gate fusion mechanism to combine both representations. Specifically, we compute the gate \mathbf{g}_i as follows:

$$\mathbf{g}_i = \sigma(\mathbf{W}_g [\mathbf{e}_i^{\text{hidden}} \parallel \mathbf{e}_i^{\text{emb}}] + \mathbf{b}_g), \quad (3)$$

where \parallel denotes concatenation, σ is the sigmoid function, and $\mathbf{W}_g \in \mathbb{R}^{2d \times d}$ and $\mathbf{b}_g \in \mathbb{R}^d$ are the weight and bias parameters of the gate, and d is the embedding size of \mathcal{M}_{TTS} . Finally, the fused representation is computed as:

$$\mathbf{c}_i = \mathbf{g}_i \odot \mathbf{e}_i^{\text{hidden}} + (1 - \mathbf{g}_i) \odot \mathbf{e}_i^{\text{emb}}, \quad (4)$$

where \odot denotes element-wise multiplication. This fused representations $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_N]$ are then passed to \mathcal{M}_{TTS} for generating speech tokens.

To achieve streaming generation, i.e., to generate speech tokens simultaneously during the LLM’s output process, we adopt a “Read- \mathcal{R} -Write- \mathcal{W} ” strategy, similar to CosyVoice 2. Specifically, we mix the fused representation \mathbf{C} and the speech tokens Y^U at a predefined ratio $\mathcal{R} : \mathcal{W}$. For every \mathcal{R} fused representations read in, the model generates \mathcal{W} speech tokens. Once all fused representations are read, the model continues to generate the remaining speech tokens until completion. During

training, cross-entropy loss is computed only for the generated speech tokens as follows:

$$\mathcal{L}_{\text{TTS}} = - \sum_{i=1}^M \log P(y_i^U | \mathbf{C}_{\leq \min(\lfloor \frac{i-1}{W} \rfloor + 1) \cdot \mathcal{R}, N}, Y_{<i}^U), \quad (5)$$

where $\mathbf{C}_{\leq \min(\lfloor \frac{i-1}{W} \rfloor + 1) \cdot \mathcal{R}, N}$ denotes the fused representations that have already been read.

Flow Matching Model The speech tokens generated by \mathcal{M}_{TTS} are further processed by a chunk-aware causal flow matching model (Lipman et al., 2023) to synthesize the mel spectrogram in a streaming manner. Every time \mathcal{W} speech tokens are generated, they are treated as a chunk for mel spectrogram synthesis. The synthesized mel spectrogram is then passed through a HiFi-GAN vocoder (Kong et al., 2020) to generate the final waveform. We use the pretrained flow matching model and vocoder in CosyVoice 2.

2.3 Training

The training of LLaMA-Omni 2 relies solely on 200K multi-turn speech-to-speech dialogue data (we will describe how this is synthesized in Section 3) and does not use any ASR or TTS data. We find that it is sufficient to achieve excellent performance while minimizing training costs. Specifically, the training process consists of two stages, as shown in Figure 1.

Stage I In Stage I training, we train the speech-to-text and text-to-speech components separately. The training data consists of <speech instruction, text response> pairs and <text response, speech response> pairs from the multi-turn speech-to-speech dialogue data. Specifically, for the speech-to-text part (Stage I(a)), we freeze the speech encoder and train the speech adapter and LLM with cross-entropy loss. For the text-to-speech part (Stage I(b)), we train the text-to-speech language model with cross-entropy loss. Note that during this stage, the gate fusion module is not trained, and only text embeddings are input into \mathcal{M}_{TTS} .

Stage II In Stage II, we train the model’s speech-to-speech generation capability with speech-to-speech dialogue data. During this stage, we freeze the speech encoder, speech adapter, and LLM, and only train the gate fusion module and \mathcal{M}_{TTS} .

2.4 Inference

During inference, the LLM autoregressively generates the text response based on the speech instruction. After generating \mathcal{R} text tokens, its hidden states and the corresponding decoded text are fed into the gate fusion module and \mathcal{M}_{TTS} to generate \mathcal{W} speech tokens, which are then passed through the flow matching model and the vocoder to synthesize a speech chunk. In this way, text and speech responses can be generated simultaneously. The response latency for the first synthesized speech chunk can be calculated as:

$$\mathcal{T}_{\text{total}} = \mathcal{T}_{\text{LLM}}(\mathcal{R}) + \mathcal{T}_{\text{TTS}}(\mathcal{W}) + \mathcal{T}_{\text{FM}}(\mathcal{W}) + \mathcal{T}_{\text{Voc}}(2\mathcal{W}), \quad (6)$$

where $\mathcal{T}_{\text{LLM}}(\mathcal{R})$ and $\mathcal{T}_{\text{TTS}}(\mathcal{W})$ represent the time required by the \mathcal{M}_{LLM} and \mathcal{M}_{TTS} models to generate \mathcal{R} and \mathcal{W} tokens, respectively. $\mathcal{T}_{\text{FM}}(\mathcal{W})$ and $\mathcal{T}_{\text{Voc}}(2\mathcal{W})$ represent the decoding times of the flow matching model and vocoder when the inputs are \mathcal{W} and $2\mathcal{W}$ tokens², respectively.

3 Data Construction

In this section, we introduce the process of constructing multi-turn speech-to-speech dialogue data. Our data is an extension of the InstructS2S-200K dataset introduced in Fang et al. (2025), which contains 200K single-turn instruction-following samples designed for speech interaction scenarios. These samples are derived from the Alpaca (Taori et al., 2023) and UltraChat (Ding et al., 2023) datasets through rewriting using LLMs. Specifically, for each sample, we first sample the number of turns from a Poisson distribution: $N \sim \text{Poisson}(\lambda = 2)$, then clip N to the range of 1 to 5. Next, we use the Llama-3.3-70B-Instruct³ (Dubey et al., 2024) model to iteratively generate the dialog. For the i -th turn, the instruction and response are generated based on the dialogue history of previous $i - 1$ turns. In this way, we obtain 200K multi-turn text dialog samples.

Next, we need to convert the text dialogue into speech. To simulate real-world applications, we aim to have varied voices for the instruction, while maintaining a consistent voice for the response. For each multi-turn dialogue, we first use the fish-speech-1.5⁴ model (Liao et al., 2024) to synthesize

²The length of the mel spectrogram is twice that of the speech tokens (50 Hz vs. 25 Hz).

³<https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>

⁴<https://huggingface.co/fishaudio/fish-speech-1.5>

a short prompt (e.g., "This is a randomly generated voice") with a random voice. Then, we use the synthesized speech as the prompt for the CosyVoice2-0.5B⁵ model, which synthesizes the instruction into speech while simultaneously cloning the voice. This ensures consistency in the voice across different turns of the dialogue, while maintaining diversity across dialogues. For all responses, we use a uniform voice as the prompt and then synthesize the speech using the CosyVoice2-0.5B model.

4 Experiments

4.1 Experimental Setups

Model Configuration We use the encoder of Whisper-large-v3 as the speech encoder. The speech adapter first performs a $5\times$ downsampling, followed by a FFN with an intermediate dimension of 2048. For the LLM, we select the Qwen2.5 series models, including Qwen2.5-0.5B/1.5B/3B/7B/14B-Instruct models. We refer to the corresponding models as LLaMA-Omni2-0.5B/1.5B/3B/7B/14B in the following sections. For the text-to-speech language model, we initialize it with the Qwen2.5-0.5B model and set the read-write strategy with $\mathcal{R} = 3$ and $\mathcal{W} = 10$. We will discuss the impact of these hyperparameters on speech quality and response latency later. The speech tokenizer, flow matching model, and vocoder are directly taken from CosyVoice 2.

Training Details We use the 200K multi-turn speech-to-speech dialogue data from Section 3 for two-stage training. In Stage I(a), we freeze the speech encoder and train all parameters of the speech adaptor and LLM. The batch size is 32, and we train for 3 epochs with a peak learning rate of $5e-5$. In Stage I(b), we train the text-to-speech language model with a batch size of 32 for 5 epochs and a peak learning rate of $5e-4$. In Stage II, we freeze the speech encoder, speech adaptor, and LLM, and train the remaining components with a batch size of 32 for 1 epoch and a peak learning rate of $1e-3$. For all stages, we use a warmup strategy for the first 3% of steps and a cosine annealing learning rate scheduler. The LLaMA-Omni2-14B model is trained on 4 NVIDIA H800 GPUs, while other models are trained on 4 NVIDIA L40 GPUs.

⁵<https://www.modelscope.cn/studios/iic/CosyVoice2-0.5B>

4.2 Evaluation

Our evaluation includes two tasks: spoken question answering and speech instruction following. For both tasks, we evaluate the model’s speech-to-text and speech-to-speech capabilities. The speech-to-speech evaluation is done by transcribing the speech response into text using the Whisper-large-v3 model, and then applying the same evaluation method as used for speech-to-text evaluation. In all experiments, we use greedy search for the LLM to ensure stable results. For the text-to-speech language model, we use sampling with temperature set to 1.0, as we find that using greedy search causes the model to fall into repetition.

Spoken Question Answering The speech question answering (SpokenQA) task involves asking the model spoken questions, then checking whether the reference answer appears in the model’s response, and calculating the accuracy. We evaluate our model on two benchmarks: Llama Questions⁶ (Nachmani et al., 2024) and Web Questions⁷ (Berant et al., 2013). Since the questions in the Web Questions dataset are in text form, we use CosyVoice2-0.5B to synthesize them into speech.

Speech Instruction Following For the speech instruction following task, we follow the settings in Fang et al. (2025), selecting the *helpful_base* and *vicuna* subsets from the Alpaca-Eval⁸ (Li et al., 2023) dataset, excluding math and code-related instructions. The remaining 199 instructions are then synthesized into speech for evaluation. Following Fang et al. (2025), we evaluate the model using the following metrics:

ChatGPT Score: To evaluate the model’s ability to follow instructions, we use GPT-4o (OpenAI, 2024) to score the model’s responses. It considers factors such as helpfulness, relevance, fluency, and suitability for speech interaction scenarios, and assigns a single score between 1 and 5. The detailed prompt can be found in Appendix A.

ASR-WER: To assess the consistency between model’s text and speech responses, we use Whisper-large-v3 to transcribe the speech response into text, and calculate the word error rate (WER) between the transcribed text and text response. We perform

⁶<https://github.com/google-research-datasets/LLAMA1-Test-Set>

⁷https://huggingface.co/datasets/Stanford/web_questions

⁸https://github.com/tatsu-lab/alpaca_eval

Model	SpokenQA (Accuracy \uparrow)				Speech Instruction Following				
	Llama Questions		Web Questions		ChatGPT Score \uparrow		ASR-WER \downarrow	UTMOS \uparrow	Latency (ms) \downarrow
	S2T	S2S	S2T	S2S	S2T	S2S			
TWIST	-	4.0	-	1.5	-	-	-	-	-
SpeechGPT	21.6	-	6.5	-	2.98	2.17	40.01	3.51	5587.94
Spectron	21.9	-	6.1	-	-	-	-	-	-
Moshi (7B)	62.3	21.0	26.6	9.2	-	-	-	-	-
GLM-4-Voice (9B)	64.7	50.7	32.2	15.9	4.16	4.09	9.02	3.48	1562.81
LLaMA-Omni (8B)	67.7	49.0	33.4	23.7	3.99	3.52	5.95	3.67	346.73
LLaMA-Omni2-0.5B	45.7	38.7	17.7	16.8	3.24	3.20	2.64	4.21	542.71
LLaMA-Omni2-1.5B	62.0	52.7	28.2	26.6	4.01	3.91	3.06	4.22	552.76
LLaMA-Omni2-3B	64.3	55.7	30.5	28.0	4.24	4.14	3.37	4.22	567.84
LLaMA-Omni2-7B	70.3	60.7	34.5	31.3	4.28	4.15	3.26	4.19	582.91
LLaMA-Omni2-14B	73.0	62.7	40.4	37.1	4.56	4.35	3.89	4.20	663.32

Table 1: Results on speech question answering and speech instruction following benchmarks. S2T and S2S represent speech-to-text and speech-to-speech, respectively. We set $\mathcal{R} = 3$ and $\mathcal{W} = 10$ for all LLaMA-Omni2 series models.

text normalization⁹ before calculating the WER.

UTMOS: To evaluate the naturalness of the generated speech, we use the UTMOS model¹⁰ (Saeki et al., 2022) to predict the mean opinion score (MOS) of the generated speech.

Latency: We measure the time from receiving the speech instruction to generating the first speech chunk on a single NVIDIA L40 GPU.

4.3 Baseline Systems

We primarily compare LLaMA-Omni 2 with the following baseline systems:

LLaMA-Omni (Fang et al., 2025): One of the earliest SpeechLMs that achieves real-time speech interaction, by using a CTC-based (Graves et al., 2006) streaming speech decoder to simultaneously generate text and speech units. The generated units are fed into the vocoder for streaming synthesis in fixed-size chunks. We set the chunk size $\Omega = 40$.

GLM-4-Voice (Zeng et al., 2024a): The current state-of-the-art native SpeechLM, pretrained on millions of hours of speech data. It enables real-time speech interaction by alternately generating text and speech tokens in a fixed ratio of 13:26. The generated speech tokens are input into a flow matching model with a fixed chunk size.

In addition, we also borrow some results from Zeng et al. (2024a), including results of TWIST (Hassid et al., 2024b), SpeechGPT (Zhang et al., 2023), Spectron (Nachmani et al., 2024), and Moshi (Défossez et al., 2024).

⁹<https://github.com/openai/whisper/blob/main/whisper/normalizers/english.py>

¹⁰<https://github.com/tarepan/SpeechMOS>

5 Results and Analysis

5.1 Main Results

Table 1 presents the main results on the speech question answering and speech instruction following benchmarks.

Spoken Question Answering For the SpokenQA task, we observe that: (1) For models with similar parameter sizes, LLaMA-Omni2-7B outperforms both GLM-4-Voice and LLaMA-Omni in both S2T and S2S settings. Notably, our model significantly reduces the gap between S2T and S2S performance. For example, on the Web Questions benchmark, GLM-4-Voice drops by 16.3 (32.2 \rightarrow 15.9), LLaMA-Omni drops by 9.7 (33.4 \rightarrow 23.7), while LLaMA-Omni2-7B only drops by 3.2 (34.5 \rightarrow 31.3), demonstrating that our approach largely improves speech generation capabilities. (2) For models with varying parameter sizes, we observe that accuracy increases as the LLM size grows, indicating that LLaMA-Omni 2 effectively leverages the LLM’s inherent capabilities. For smaller models, LLaMA-Omni2-1.5B/3B exceeds the accuracy of GLM-4-Voice and LLaMA-Omni in the S2S setting, making them suitable choices for edge devices. For larger models, we observe a significant accuracy improvement with LLaMA-Omni2-14B compared to LLaMA-Omni2-7B, highlighting the potential of our approach for scaling to larger models.

Speech Instruction Following For the speech instruction following task, we observe that: (1) LLaMA-Omni2-3B/7B/14B outperforms both GLM-4-Voice and LLaMA-Omni in the S2T and S2S settings, demonstrating the strong instruction-following capabilities of our models. (2) Similar

Model	Score (S2S)	ASR-WER
LLaMA-Omni2-7B	4.15	3.26
w/o Gate Fusion	4.02	4.89
w/o Text Embedding	3.88	6.83

Table 2: Ablation study on the gate fusion module with LLaMA-Omni2-7B.

Model	Score (S2S)	ASR-WER
Streaming TTS	4.15	3.26
Offline TTS	4.13	3.51
Text Pretrained	3.53	10.34
Scratch	1.08	80.65

Table 3: Ablation study on different TTS pretraining strategies with LLaMA-Omni2-7B.

to the results on SpokenQA benchmarks, we observe that model performance improves as the LLM size increases, with LLaMA-Omni2-14B achieving significantly better performance. (3) The models’ ASR-WER is generally low, significantly lower than previous models, proving that our models maintain strong consistency between the text and speech responses. (4) Regarding speech quality, thanks to the CosyVoice 2’s strong causal flow matching model, our models achieve good UTMOs scores under streaming synthesis, significantly outperforming the baseline models. (5) The latency of LLaMA-Omni 2 is around 600ms. Although it is slightly higher than LLaMA-Omni, it still meets the requirements for real-time interaction and is significantly lower than that of GLM-4-Voice.

5.2 Ablation Studies

To understand the impact of different factors on overall performance, we conduct a series of ablation studies on the LLaMA-Omni2-7B model.

Gate Fusion Module Table 2 shows the ablation study on the gate fusion module. Gate fusion module allows the model to adaptively fuse LLM hidden states and text embeddings, considering both contextual information and textual content. When the gate fusion module is removed and the two components are simply added together ($e_i^{\text{hidden}} + e_i^{\text{emb}}$) as input to the text-to-speech language model, we observe a decrease in performance. Further removing the text embedding and only inputting the hidden states (e_i^{hidden}) results in a further performance decline. This validates the effectiveness of adding text embeddings as input and adaptively

\mathcal{R}	\mathcal{W}	Score (S2S)	ASR-WER	UTMOS	Latency (ms)
1	5	4.09	3.48	3.98	457.29
2	10	4.15	4.00	4.19	557.79
3	10	4.15	3.26	4.19	582.91
3	15	4.12	4.37	4.27	663.32
4	15	4.10	3.77	4.27	683.42
5	20	4.15	3.62	4.32	798.99
Offline		4.14	3.40	4.46	-

Table 4: Ablation study on the read/write strategy with LLaMA-Omni2-7B. “Offline” means generating speech tokens only after receiving the complete input, and then synthesizing all speech tokens into waveform at once.

fusing them with the gate fusion module.

TTS Pretraining Our text-to-speech language model is initialized with the Qwen2.5-0.5B model and undergoes streaming TTS pretraining using text-speech pairs from speech dialogue data in Stage I(b) ($\mathcal{R} = 3, \mathcal{W} = 10$). We also explore several other strategies, as shown in Table 3. “Offline TTS” refers to pretraining with the offline TTS task on top of Qwen2.5-0.5B, which shows a slight performance drop compared to the streaming TTS pretraining. “Text Pretrained” refers to directly initializing with Qwen2.5-0.5B (with the extended vocabulary including speech tokens), and we observe a significant performance decline. “Scratch” refers to a randomly initialized model, whose loss fails to converge within a short period. These experiments demonstrate the importance of pretraining for the TTS language model.

Read/Write Strategy The read/write strategies of the TTS language model is a key factor influencing performance, primarily affecting the speech quality and system response latency. As shown in Table 4, we explore different combinations of \mathcal{R} and \mathcal{W} . First, we observe that when $\mathcal{R} = 3$ and $\mathcal{W} = 10$, the ASR-WER is the lowest, indicating the best alignment between speech and text responses. As for the UTMOS score, we find that it is primarily determined by \mathcal{W} , as \mathcal{W} represents the chunk size of speech tokens input to the flow matching model, with larger chunk sizes leading to better speech quality. Regarding response latency, it is jointly determined by \mathcal{R} and \mathcal{W} , as shown in Equation 6. Without any engineering optimizations, LLaMA-Omni2-7B can achieve a latency below 500ms. We choose $\mathcal{R} = 3$ and $\mathcal{W} = 10$ in our main experiments because it provides a good trade-off across all aspects.

#Samples	Multiturn	SpokenQA (Accuracy)				Speech Instruction Following		
		Llama Questions		Web Questions		ChatGPT Score		ASR-WER
		S2T	S2S	S2T	S2S	S2T	S2S	
200K	✓	70.3	60.7	34.5	31.3	4.28	4.15	3.26
200K	×	70.0	59.0	33.7	30.5	4.11	3.98	3.28
150K	✓	70.7	58.7	34.7	31.7	4.23	4.10	3.71
100K	✓	67.7	55.3	34.1	29.9	4.19	4.07	4.45
50K	✓	50.0	37.0	16.6	13.9	3.02	2.84	5.42

Table 5: Results under different training data sizes with LLaMA-Omni2-7B.

5.3 Effects of the Training Data Sizes

We explore the impact of different training data sizes on performance. As shown in Table 5, we first observe that, with the same number of training samples, multi-turn dialogue data consistently achieves better results across all benchmarks compared to single-turn dialogue data, highlighting the effectiveness of multi-turn dialogue data for training. Additionally, for different training data sizes, we observe that as the data size increases, the model’s performance improves, gradually stabilizing at 200K training samples. This indicates that our 200K multi-turn dialogue data is generally sufficient while ensuring efficient training.

6 Related Work

With the rapid development of LLMs, SpeechLMs have gained widespread attention in recent years (Cui et al., 2024; Ji et al., 2024), aiming to endow LLMs with the ability to understand or generate speech. Generally speaking, SpeechLMs can be divided into two categories: native SpeechLMs and modular SpeechLMs. Native SpeechLMs refer to decoder-only Transformer models capable of directly inputting and outputting speech tokens. Some early works include SpeechGPT (Zhang et al., 2023, 2024a), AudioPaLM (Rubenstein et al., 2023), and TWIST (Hassid et al., 2024a). These models first convert speech into discrete tokens, then extend the vocabulary of pretrained LLMs to include these tokens, and finally train the LLMs using a large amount of speech or speech-text pair data. Spirit-LM (Nguyen et al., 2024) and GLM-4-Voice (Zeng et al., 2025, 2024a) propose training models using speech-text interleaved data to encourage cross-modal knowledge transfer. Moshi (Défossez et al., 2024), OmniFlatTen (Zhang et al., 2024b) and LSLM (Ma et al., 2024b) propose models capable of full-duplex con-

versations. IntrinsicVoice (Zhang et al., 2024d) proposes a GroupFormer architecture to shorten speech length to be closer to that of text. In contrast to native SpeechLMs, modular SpeechLMs add speech-related modules on top of LLMs. Early works achieve speech understanding tasks by combining speech encoders with LLMs, but are unable to perform speech generation (Wu et al., 2023; Wang et al., 2023; Chu et al., 2023; Yu et al., 2024; Ma et al., 2024c; Hono et al., 2024; Chen et al., 2024b; Tang et al., 2024; Chu et al., 2024; Fathullah et al., 2024). To achieve speech generation, LLaMA-Omni (Fang et al., 2025), Freeze-Omni (Wang et al., 2024), and OpenOmni (Luo et al., 2025) integrate a streaming speech decoder after LLMs, similar to previous techniques in simultaneous translation (Zhang et al., 2024c; Ma et al., 2024a, 2025). Mini-Omni (Xie and Wu, 2024) and SLAM-Omni (Chen et al., 2024a) enable LLMs to generate speech tokens simultaneously while generating text tokens. The most related work to ours is the concurrent work Minmo (Chen et al., 2025), which also adopts an autoregressive streaming speech decoder similar to CosyVoice 2. In comparison, Minmo is trained on 1.4M hours of data, while we train on only a few thousand hours of data, providing a more efficient training solution. Additionally, we conduct detailed ablation studies on LLM sizes, read-write strategies, and model architecture to offer a more comprehensive understanding of the model.

7 Conclusion

In this paper, we introduce LLaMA-Omni 2, a series of speech language models ranging from 0.5B to 14B parameters, designed to enable real-time, high-quality speech interaction. LLaMA-Omni 2 achieves streaming speech generation by integrating an autoregressive text-to-speech language

model and a causal flow matching model. Experimental results on spoken question answering and speech instruction following tasks show that LLaMA-Omni 2 outperforms previous state-of-the-art speech language models, including LLaMA-Omni and GLM-4-Voice. Additionally, LLaMA-Omni 2 can achieve latency under 600ms, meeting real-time interaction requirements. We also conduct detailed ablation studies to understand the impact of various factors on overall performance. In the future, we will explore enhancing LLaMA-Omni 2 to generate more human-like speech, incorporating features such as emotion and dialects.

Limitations

One limitation of our model is that currently it cannot generate speech responses with different styles (such as emotion or speech rate) based on the content of the input speech or underlying paralinguistic information, as we have only trained on conventional speech-to-speech dialogue data. However, we believe this functionality can be achieved through a data-driven approach, as our model is end-to-end trained and could acquire this capability after further training with suitable data. We plan to explore this in the future.

Ethical Considerations

Since LLaMA-Omni 2 is built on LLMs, it carries some of the same risks as LLMs, such as the potential for factual errors or other hallucination issues in its outputs. We recommend that the model's outputs be checked in practical use to ensure they comply with the required standards.

Acknowledgements

We thank all the anonymous reviewers for their insightful and valuable comments. This paper is supported by National Natural Science Foundation of China (Grant No.62376260).

References

- Keyu An, Qian Chen, Chong Deng, Zhihao Du, Changfeng Gao, Zhifu Gao, Yue Gu, Ting He, Hangrui Hu, Kai Hu, et al. 2024. Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms. *arXiv preprint arXiv:2407.04051*.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. [Semantic parsing on Freebase from question-answer pairs](#). In *Proceedings of the 2013*

Conference on Empirical Methods in Natural Language Processing, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

- Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, et al. 2025. Minmo: A multimodal large language model for seamless voice interaction. *arXiv preprint arXiv:2501.06282*.
- Wenxi Chen, Ziyang Ma, Ruiqi Yan, Yuzhe Liang, Xiquan Li, Ruiyang Xu, Zhikang Niu, Yanqiao Zhu, Yifan Yang, Zhanxun Liu, et al. 2024a. Slam-omni: Timbre-controllable voice interaction system with single-stage training. *arXiv preprint arXiv:2412.15649*.
- Xi Chen, Songyang Zhang, Qibing Bai, Kai Chen, and Satoshi Nakamura. 2024b. [LLaST: Improved end-to-end speech translation system leveraged by large language models](#). In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6976–6987, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.
- Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.
- Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.
- Leigh Clark, Philip Doyle, Diego Garaialde, Emer Gilmartin, Stephan Schlögl, Jens Edlund, Matthew Aylett, João Cabral, Cosmin Munteanu, Justin Edwards, and Benjamin R Cowan. 2019. [The state of speech in HCI: Trends, themes and challenges](#). *Interacting with Computers*, 31(4):349–371.
- Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Yiwen Guo, and Irwin King. 2024. Recent advances in speech language models: A survey. *arXiv preprint arXiv:2410.03751*.
- Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. [Moshi: a speech-text foundation model for real-time dialogue](#). Technical report.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.
- Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. 2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*.

- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2025. **LLaMA-omni: Seamless speech interaction with large language models**. In *The Thirteenth International Conference on Learning Representations*.
- Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Ke Li, Junteng Jia, Yuan Shangguan, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. 2024. Audiochatllama: Towards general-purpose speech abilities for llms. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5522–5532.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. **Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks**. In *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, page 369–376, New York, NY, USA. Association for Computing Machinery.
- Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, et al. 2024a. Textually pretrained speech language models. *Advances in Neural Information Processing Systems*, 36.
- Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, et al. 2024b. Textually pretrained speech language models. *Advances in Neural Information Processing Systems*, 36.
- Yukiya Hono, Koh Mitsuda, Tianyu Zhao, Kentaro Mitsui, Toshiaki Wakatsuki, and Kei Sawada. 2024. **Integrating pre-trained speech and language models for end-to-end speech recognition**. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13289–13305, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.
- Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, et al. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.
- Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. **Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis**. In *Advances in Neural Information Processing Systems*, volume 33, pages 17022–17033. Curran Associates, Inc.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.
- Shijia Liao, Yuxuan Wang, Tianyu Li, Yifan Cheng, Ruoyi Zhang, Rongzhi Zhou, and Yijin Xing. 2024. **Fish-speech: Leveraging large language models for advanced multilingual text-to-speech synthesis**. Preprint, arXiv:2411.01156.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. **Flow matching for generative modeling**. In *The Eleventh International Conference on Learning Representations*.
- Run Luo, Ting-En Lin, Haonan Zhang, Yuchuan Wu, Xiong Liu, Min Yang, Yongbin Li, Longze Chen, Jiaming Li, Lei Zhang, et al. 2025. Openomni: Large language models pivot zero-shot omnimodal alignment across language with real-time self-aware emotional speech synthesis. *arXiv preprint arXiv:2501.04561*.
- Zhengrui Ma, Qingkai Fang, Shaolei Zhang, Shoutao Guo, Yang Feng, and Min Zhang. 2024a. **A non-autoregressive generation framework for end-to-end simultaneous speech-to-any translation**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557–1575, Bangkok, Thailand. Association for Computational Linguistics.
- Zhengrui Ma, Yang Feng, and Min Zhang. 2025. **Overcoming non-monotonicity in transducer-based streaming generation**. In *Proceedings of ICML 2025*.
- Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2024b. Language model can listen while speaking. *arXiv preprint arXiv:2408.02622*.
- Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, et al. 2024c. An embarrassingly simple approach for llm with strong asr capacity. *arXiv preprint arXiv:2402.08846*.
- Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2024. **Finite scalar quantization: VQ-VAE made simple**. In *The Twelfth International Conference on Learning Representations*.
- Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. 2024. **Spoken question answering and speech continuation using spectrogram-powered LLM**. In *The Twelfth International Conference on Learning Representations*.
- Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, et al. 2024. Spirit-llm: Interleaved

- spoken and written language model. *arXiv preprint arXiv:2402.05755*.
- Open-Moss. 2025. Speechgpt 2.0-preview. <https://github.com/OpenMOSS/SpeechGPT-2.0-preview>.
- OpenAI. 2022. [Introducing chatgpt](#).
- OpenAI. 2024. [Hello gpt-4o](#).
- Alec Radford. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.
- Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.
- Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. In *Interspeech 2022*, pages 4521–4525.
- Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun MA, and Chao Zhang. 2024. SALMONN: Towards generic hearing abilities for large language models. In *The Twelfth International Conference on Learning Representations*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Qwen Team. 2024. [Qwen2.5: A party of foundation models](#).
- A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.
- Chen Wang, Minpeng Liao, Zhongqiang Huang, Jinliang Lu, Junhong Wu, Yuchen Liu, Chengqing Zong, and Jiajun Zhang. 2023. Blsp: Bootstrapping language-speech pre-training via behavior alignment of continuation writing. *arXiv preprint arXiv:2309.00916*.
- Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. 2024. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*.
- Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linqun Liu, et al. 2023. On decoder-only architecture for speech-to-text and large language model integration. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.
- Zhifei Xie and Changqiao Wu. 2024. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*.
- Wenyi Yu, Changli Tang, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. Connecting speech encoder and large language model for asr. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12637–12641. IEEE.
- Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024a. [Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot](#). *Preprint*, arXiv:2412.02612.
- Aohan Zeng, Zhengxiao Du, Mingdao Liu, Lei Zhang, Shengmin Jiang, Yuxiao Dong, and Jie Tang. 2024b. [Scaling speech-text pre-training with synthetic interleaved data](#). *Preprint*, arXiv:2411.17607.
- Aohan Zeng, Zhengxiao Du, Mingdao Liu, Lei Zhang, shengmin jiang, Yuxiao Dong, and Jie Tang. 2025. [Scaling speech-text pre-training with synthetic interleaved data](#). In *The Thirteenth International Conference on Learning Representations*.
- Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. [SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773, Singapore. Association for Computational Linguistics.
- Dong Zhang, Xin Zhang, Jun Zhan, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2024a. [Speechgpt-gen: Scaling chain-of-information speech generation](#). *arXiv preprint arXiv:2401.13527*.
- Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chaohong Tan, Zhihao Du, et al. 2024b. [Omniflatten: An end-to-end gpt model for seamless voice conversation](#). *arXiv preprint arXiv:2410.17799*.
- Shaolei Zhang, Qingkai Fang, Shoutao Guo, Zhenrui Ma, Min Zhang, and Yang Feng. 2024c. [Streamspeech: Simultaneous speech-to-speech translation with multi-task learning](#). *arXiv preprint arXiv:2406.03049*.
- Xin Zhang, Xiang Lyu, Zhihao Du, Qian Chen, Dong Zhang, Hangrui Hu, Chaohong Tan, Tianyu Zhao,

Yuxuan Wang, Bin Zhang, et al. 2024d. Intrinsicvoice: Empowering llms with intrinsic real-time voice interaction abilities. *arXiv preprint arXiv:2410.08035*.

A Prompt

Prompt for ChatGPT Scoring (Model: GPT-4o)

I need your help to evaluate the performance of several models in a speech interaction scenario. The models receive the user's speech input and respond with speech output. For evaluation purposes, both the user's speech input and the model's speech response have been transcribed into text using Automatic Speech Recognition (ASR). Your task is to rate the model's responses based on the provided user input transcription [Instruction] and the model's output transcription [Response]. Please consider factors such as helpfulness, relevance, fluency, and suitability for speech interaction in your evaluation, and provide a single score on a scale from 1 to 5.

Below are the transcription of user's instruction and models' response:

[Instruction]: {**instruction**}

[Response]: {**response**}

After evaluating, please output the scores in JSON format: {score: ...}. You don't need to provide any explanations.

B Detailed Latency

We list the detailed latency at different stages of the model in Table 6. "LLM" refers to the latency for generating the first \mathcal{R} text tokens, "TTS" refers to the latency for generating the first \mathcal{W} speech tokens, and "FM+Voc" refers to the latency for generating the first speech chunk using the flow matching model and vocoder.

Model	\mathcal{R}	\mathcal{W}	Latency (ms)			Total
			LLM	TTS	FM+Voc	
LLaMA-Omni2-0.5B	3	10	190.95	165.83	185.93	542.71
LLaMA-Omni2-1.5B	3	10	201.01	165.83	185.93	552.76
LLaMA-Omni2-3B	3	10	216.08	165.83	185.93	567.84
LLaMA-Omni2-7B	3	10	231.16	165.83	185.93	582.91
LLaMA-Omni2-14B	3	10	311.56	165.83	185.93	663.32
LLaMA-Omni2-7B	1	5	185.93	85.43	185.93	457.29
LLaMA-Omni2-7B	2	10	206.03	165.83	185.93	557.79
LLaMA-Omni2-7B	3	10	231.16	165.83	185.93	582.91
LLaMA-Omni2-7B	3	15	231.16	246.23	185.93	663.32
LLaMA-Omni2-7B	4	15	251.26	246.23	185.93	683.42
LLaMA-Omni2-7B	5	20	271.36	336.68	190.95	798.99

Table 6: Detailed latency of LLaMA-Omni2 series models.