Mini-Omni: Language Models Can Hear, Talk While Thinking in Streaming

Zhifei Xie♠♣*
xzf24@mails.tsinghua.edu.cn

Changqiao Wu♠*
wuchangqiao@inspirai.com

♠ Inspirai   ♣ Tsinghua University
https://github.com/gpt-omni/mini-omni

Abstract

Recent advances in language models have achieved significant progress. GPT-4o, as a new milestone, has enabled real-time conversations with humans, demonstrating near-human natural fluency. Such human-computer interaction necessitates models with the capability to perform reasoning directly with the audio modality and generate output in streaming. However, this remains beyond the reach of current academic models, as they typically depend on extra TTS systems for speech synthesis, resulting in undesirable latency. This paper introduces the Mini-Omni, an audio-based end-to-end conversational model, capable of real-time speech interaction. To achieve this capability, we propose a text-instructed speech generation method, along with batch-parallel strategies during inference to further boost the performance. Our method also helps to retain the original model's language capabilities with minimal degradation, enabling other works to establish real-time interaction capabilities. We call this training method **"Any Model Can Talk"**. We also introduce the **VoiceAssistant-400K** dataset to fine-tune models optimized for speech output. To our best knowledge, Mini-Omni is the first fully end-to-end, open-source model for real-time speech interaction, offering valuable potential for future research.
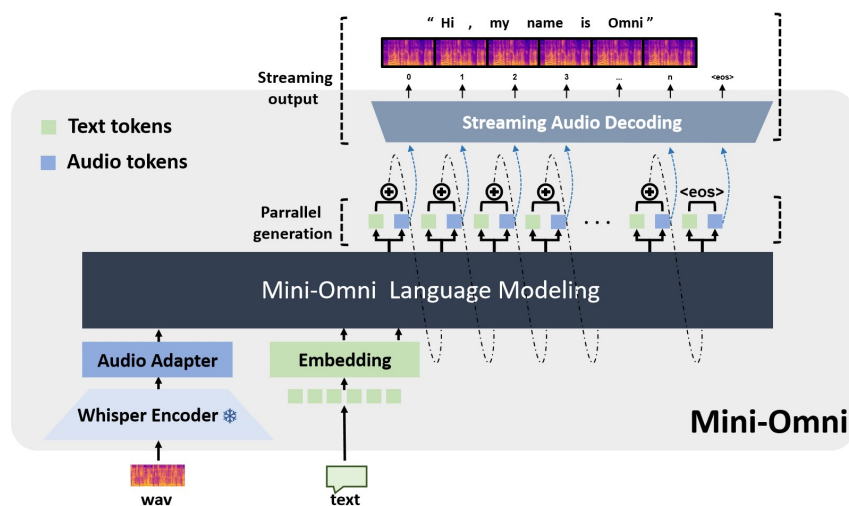
Figure 1: The **Mini-Omni** model architecture.

---

*Equal contribution. Work done during Zhifei Xie's internship at Inspirai.

Technical report.

1 Introduction

Recent developments in large language models have progressed rapidly, with models becoming increasingly powerful, such as off-the-shelf Llama 3.1 [meta, 2024], Mixtral [mixtral, 2024], Qwen-2 [Yang et al., 2024a], and the well-known GPT-4. As an extension of their capabilities, language models are beginning to master understanding other modalities, exemplified by LLaVA [Liu et al., 2024], Qwen2-Audio [Chu et al., 2024] and Video-llama [Zhang et al., 2023b]. Despite their strength in specific tasks, a significant gap remains that hinders further integration of large language models into daily application: real-time voice interaction capability. GPT-4o [openai, 2024], introduced by OpenAI, is the first model to feature real-time multimodal speech interaction capabilities. It can understand and engage with vision, audio, and text while enabling real-time speech conversations, although it remains closed-source. Other models typically adopt two approaches to incorporate speech capabilities. The first is a cascade method, where the language model generates text, followed by a text-to-speech (TTS) model for audio synthesis. This approach introduces significant latency due to the time required for text generation, severely impacting user experience. The second, an end-to-end method like SpeechGPT [Zhang et al., 2023a], generates text before continuing to generate audio. However, this still requires waiting for text generation. Large language models need real end-to-end speech output capabilities to provide real-time feedback.

Enhancing models with speech output capabilities is a challenging task, primarily due to four factors: (1) **Complexity of Audio Reasoning**: Our experiments indicate that direct training for audio modality reasoning is highly challenging, often resulting in incoherent outputs from the model. (2) **Model Complexity**: Incorporating additional modules for speech input and output increases the overall complexity. (3) **Difficulty in Modality Alignment**: The reasoning abilities developed for text are difficult to transfer to the audio domain. (4) **Resource Demands**: Adapting a model's text capabilities to the speech modality requires converting all data labels into audio and retraining, significantly increasing resource consumption.

In this paper, we propose **Mini-Omni**, the first open-source multi-model large language model with real-time conversational capabilities, featuring fully end-to-end speech input and output abilities. It also includes various other audio-to-text functionalities such as Automatic Speech Recognition (ASR). We adapt currently available off-the-shelf methods for discretizing speech tokens and employ the simplest model architecture, making it easy for our model and approach to be adapted by other researchers. Direct audio reasoning poses significant challenges; however, our approach successfully addresses this using only a 0.5B model and a limited amount of synthesized audio data. Importantly, our training framework achieves this without heavy reliance on extensive model capabilities or large volumes of data.

To leverage and preserve the original capabilities of the language model, we propose a parallel generation paradigm in which the transformer simultaneously produces audio and text tokens. Subsequently, we observed a minimal impact of the audio modality on text capabilities and further introduced **batch-based parallel generation**, which significantly enhances the model's reasoning ability during streaming audio output. As a poinerr, we opted not to sacrifice audio quality for a simpler and lower bitrate audio encoder, in order to reduce the complexity of audio inference in the model. However, to ensure audio quality, we selected SNAC [Siuzdak, 2024], a music-grade encoder features 8 layers of codebooks and processes hundreds of tokens per second. Innovatively, we applied **text-instructed delayed parallel generation** to address the issue of long SNAC codebook sequences. Experiments show that the audio output quality is on par with common TTS systems.

We also propose a method that requires minimal training and modification of the original model, enabling other works to rapidly develop their own speech capabilities. We refer to this approach as **"Any Model Can Talk"**, designed to achieve speech output using a limited amount of additional data. The approach extend speech capabilities through additional adapters and pre-trained models, fine-tuning with a small amount of synthesized data. This is combined with the aforementioned parallel modeling approach to enable streaming output in the new modality while preserving the original model's reasoning capabilities.

To evaluate the capabilities of **Mini-Omni**, we first assessed its performance on traditional text-to-speech multi-modal tasks, including text-based question answering (textQA), automatic speech recognition (ASR), text-to-speech response, and speech-based question answering (speechQA). The model demonstrated strong proficiency in these fundamental tasks. Additionally, we conduct a series of experiments to investigate the impact on the original model's capabilities and assess the

effectiveness and variations of our inference method. Preliminary experiments demonstrate that batch parallel inference preserves the model's original capabilities. We will conduct further experiments and provide additional details in due course.

Lastly, we observed that most open-source QA datasets contain mixed code or overly lengthy text, rendering them unsuitable for speech model. To overcome this limitation, we introduce the **VoiceAssistant-400K** dataset, comprising over 400,000 entries specifically generated by GPT-4o for speech assistant supervised fine-tuning (SFT).

In summary, we make the following contributions:

- We introduce **Mini-Omni**, the first open-source end-to-end multimodal large model with audio input and audio streaming output capabilities. We propose a unique text-instruct parallel generation method that enables speech inference outputs aligned with textual capabilities, achieved with minimal data. We further enhance this with delayed parallelism, accelerating audio inference speed.

- We introduce "**Any Model Can Talk**", an innovative approach that enhances performance without altering the architecture of large models by focusing on training and inference. Our method employs a three-phase training process for speech-to-text and text-to-speech adapters, including annealing and SFT. Our method involves minimal training and modification of the original model, aiming to provide a reference for incorporating interaction capabilities into other models.

- We identified shortcomings in existing open-source QA datasets when training audio assistants and proposed a dedicated dataset for speech model outputs, called **VoiceAssistant-400K**. This dataset, synthesized using GPT-4o, can be used to fine-tune models to develop the tone of a voice assistant.

2 Related Work

Multimodal Understanding Recently, researchers have been increasingly focused on advancing unified models for cross-modal understanding. These approaches typically employ a well-pretrained neural network as the encoder for relevant modalities, using a lightweight adapter to align the encoder's output with the text input of language model. Classical works such as LLaVA [Liu et al., 2024], Flamingo [Alayrac et al., 2022] and BLIP [Li et al., 2022] are used for visual understanding, while in the audio domain, models like Whisper [Radford et al., 2023] and Beats [Chen et al., 2022] are commonly utilized as encoders for semantic and acoustic features. In Llama 3.1, Whisper is employed, while SpeechVerse [Das et al., 2024] leverages WavLM [Hu et al., 2024]; SALMONN [Tang et al., 2023], combine Whisper and Beats to extract features. Such works are often constrained to producing output in the text modality.

Audio Language Modeling Recently, an increasing number of studies have employed audio tokenization to bridge the gap between audio and text. Audio tokenization converts continuous audio signals into discrete audio tokens, enabling large language models to perform inference and even cross-modal interactions. As a result, a variety of speech-text tasks, such as ASR, TTS, music understanding and generation, and sound editing, can be accomplished. MegaTTS [Jiang et al., 2023] utilized audio codecs for speech synthesis, while efforts like InstructTTS [Yang et al., 2024b], SpearTTS [Kharitonov et al., 2023], and Voicebox [Le et al., 2024] have further explored optimizations in decoding methods and conditioning techniques, employing Diffusion as the converter from tokens to audio.

Real-Time Human-Machine Interaction Models Since the introduction of GPT-4o [openai, 2024], real-time conversational models have achieved unprecedented results, providing near-instantaneous voice feedback to user inputs, marking a significant milestone for the next generation of multi-modal large models. However, the technical implementations remain proprietary. Models with real-time interaction capabilities are currently scarce. SpeechGPT [Zhang et al., 2023a] is an early end-to-end speech interaction model; however, it still suffers from latency due to the Audio-Text-Text-Audio(A-T-T-A) process, similar to Spectron [Nachmani et al., 2023]. LauraGPT [Chen et al., 2023] also employs a similar approach but not for voice conversation scenario. VITA [Fu et al., 2024] and Qwen-audio2 [Chu et al., 2024] are two models that support voice input, but they output text and rely on external TTS systems for speech synthesis. **Mini-Omni** is a fully end-to-end speech-to-speech conversational model. Through our exploration, we have identified the biggest challenge in advancing this field: the logical inconsistency in reasoning when only the audio modality is present, which we will address in the following chapter.

3 Mini-Omni

Our innovation stems from existing methods such as SpeechGPT [Zhang et al., 2023a] and Spectron [Nachmani et al., 2023] utilize the A-T-T-A approach, which mitigates the challenges of direct audio learning by guiding the speech generation process through text. However, generating text first and then audio is suboptimal for real-time dialogue scenarios. To address this, we propose a novel method for simultaneous text and audio generation. This approach hypothesizes that text outputs have higher information density, allowing for the same response with fewer tokens. During the generation of audio tokens, the model effectively conditions on corresponding text tokens, akin to an online TTS system. Prior to generating audio tokens, padding with N tokens ensures that the corresponding text tokens are produced first, allowing this to serve as a hyperparameter adjustment. Additionally, the model can also condition on speaker and style embeddings, facilitating control over speaker characteristics and stylistic elements. In this section, we will detail how we implement our idea step by step.

3.1 Audio Language Modeling

Consider $Y = (y_i \in \mathcal{V}_{\text{txt}} \mid i = 1, \ldots, t_{\text{txt}})$ as a text utterance from a vocabulary $\mathcal{V}_{\text{txt}}$ with length t_{txt}. The probability of Y can be expressed as $p(Y) = \prod_{i=1}^{t_{\text{txt}}} p(y_i \mid y_1, \ldots, y_{i-1})$. Now, when dealing with a continuous speech signal, we can convert it into discrete speech tokens (dst), represented as $D = (d_i \in \mathcal{V}_{\text{dst}} \mid i = 1, \cdots, t_{\text{dst}})$ using a tokenizer. In this context $\mathcal{V}_{\text{dst}}$ is the vocabulary of discrete speech tokens. These discrete speech tokens can be treated as spoken language within $\mathcal{V}_{\text{dst}}$ and modeled in a manner similar to text. We combine text and speech in a new vocabulary $\mathcal{V}_{\text{voxt}}$ by $\mathcal{V}_{\text{voxt}} = \mathcal{V}_{\text{txt}} \cup \mathcal{V}_{\text{dst}}$. Therefore, we can model the probability of both speech and text tokens as Z, where $Z = (z_i \in \mathcal{V} \mid i = 1, \cdots, t)$. This probability is expressed as $p(Z) = \prod_{i=1}^{t} p(z_i \mid z_1, \cdots, z_{i-1})$, Z represent discrete speech tokens $D(\mathcal{V} = \mathcal{V}_{\text{dst}})$ or text tokens $Y(\mathcal{V} = \mathcal{V}_{\text{txt}})$ or various combinations of Y and D. For the audio and text tokens generated simultaneously, the negative log-likelihood loss can be formulated as in Equation (1).

$$\mathcal{L}(T, A|C) = \sum_{j=1}^{m} \sum_{i=1}^{n_j} \log P(T_{i,j}, A_{i,j} | T_{<i,j}, A_{<i,j}; X_j) \tag{1}$$

where T, A is the text-audio output pairs in the training corpus C, and m is the number of training examples. X_j is the input condition of j-th example, n_j is max number of tokens of sample T_j and A_j, $T_{i,j}$ and $A_{i,j}$ represent the i-th text token and audio token of j-th sample.

3.2 Decoding Strategies

Audio Generation with text instruction. Language models have undergone substantial advancements, demonstrating exceptional reasoning capabilities within the text modality. In response, **Mini-Omni** has been restructured to transfer these reasoning abilities to streaming audio output through a text-audio parallel decoding approach. This method simultaneously outputs both audio and text tokens, with the audio generated via text-to-speech synthesis, ensuring real-time delivery while leveraging the text-based reasoning strengths. To align with the inputs of large models, all sequences generated in parallel are summed before producing the next token, as illustrated in Figure 1. This approach enables the model to achieve real-time voice output in chat scenarios with minimal first token delay.

Text-delay Parallel Decoding. Parallel generation was first introduced by MusicGen [Copet et al., 2024] to accelerate the music generation process, and we have integrated this approach into the text modality to enhance reasoning capabilities. Parallel decoding is feasible because audio token codebooks used in language model training typically consist of multiple layers; generating all layers simultaneously can significantly increase model speed. For real-time speech output models, parallel decoding is even more critical, allowing for the generation of hundreds of audio tokens per second on standard devices. In this paper, we employ SNAC as the audio encoder, which comprises seven token layers with complementary relationships. Therefore, we employ eight sub-Language Model heads to generate eight tokens, including text, in a single step, while maintaining a one-step delay between adjacent layers. Since audio tokens are derived from text synthesis, the text token is output first, followed by SNAC tokens from the first to the seventh layer. The process of text-first delay parallel decoding we propose is illustrated in Figure 2(b).
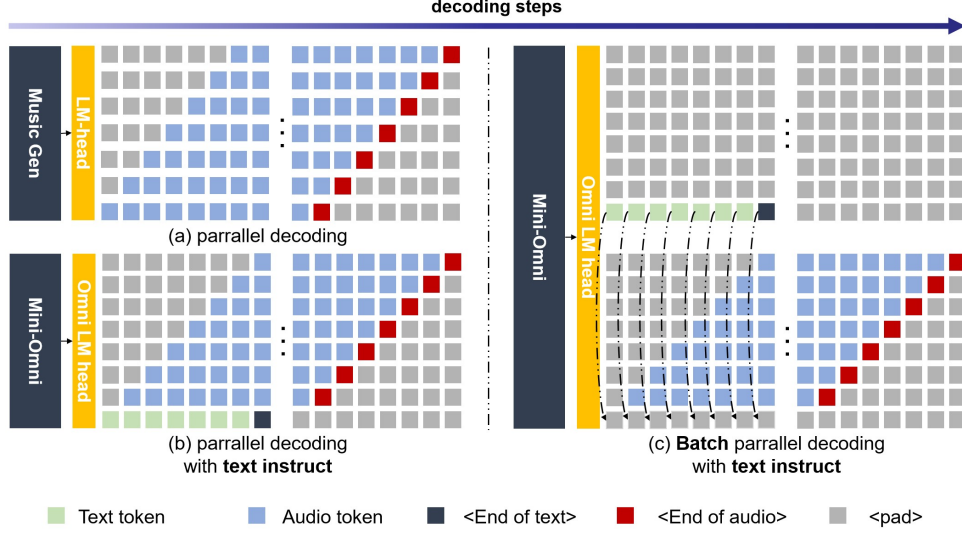
Figure 2: **Mini-Omni** incorporates text-instruct mechanisms alongside Batch parallel generation techniques.

Batch Parallel Decoding. Although the previously introduced parallel generation method effectively transfers reasoning capabilities from the text modality to the audio modality, our experiments reveal that the model's reasoning performance still varies between text and audio tasks, with audio responses tending to be simpler. We hypothesize that this is due to limitations in model capacity or insufficient audio data. To address this issue and further enhance the model's reasoning capabilities during dialogue, maximizing the transfer of its text-based abilities, we experimentally employ a Batch approach. Given the model's stronger performance in the text modality, we expand the inference task for a single input to a batch size of 2: one sample requires both text and audio responses, as described earlier, while the other sample only requires a text response, focusing on text-based audio synthesis. However, the text token output from the first sample is discarded, and the text output from the second sample is embedded into the corresponding text token positions of the first sample. Simultaneously, the audio from the first sample is streamed using the content from the text-only response of the second sample; we term this process batch parallel decoding. Through this method, we effectively and almost entirely transfer the model's text-based capabilities to the audio modality with minimal resource overhead, significantly enhancing its reasoning abilities in the new modality. The inference process of batch parallel decoding is illustrated in Figure 2(c). We believe batch parallel decoding represents a key algorithmic innovation that enables such a small model to exhibit strong conversational abilities.

3.3 Any Model Can Talk

In this section, we present our training methodology. Our approach is designed to preserve the capabilities of the original model as much as possible. This is achieved firstly due to the strong performance of our base model, and secondly because our method can be applied to other works that excel in text output but lack robust speech interaction capabilities.

Audio Encoding: The audio input primarily focuses on feature extraction from the input audio, with options including Hubert or a separately pretrained audio encoder. Given our focus on speech input, Whisper [Radford et al., 2023] and Qwen2-audio [Chu et al., 2024] also demonstrate effective performance for general audio tasks. For audio output, selecting audio tokens with a multi-codebook approach better captures audio details. We experimented with flattening for audio token modeling, but it resulted in excessively long tokens, which are detrimental to streaming and lead to unstable learning. Instead, parallel decoding, inspired by MusicGen [Copet et al., 2024], employs a delay pattern combined with text conditions, as illustrated in Figure 2.

Three-Stage Training. Our training methodology is divided into three distinct stages: **(1) Modality Alignment.** The goal of this stage is to enhance the text model's ability to understand and generate speech. The core model of **Mini-Omni** is entirely frozen, with gradients allowed only in two adapters. During this stage, we use data from speech recognition and speech synthesis to train the model's
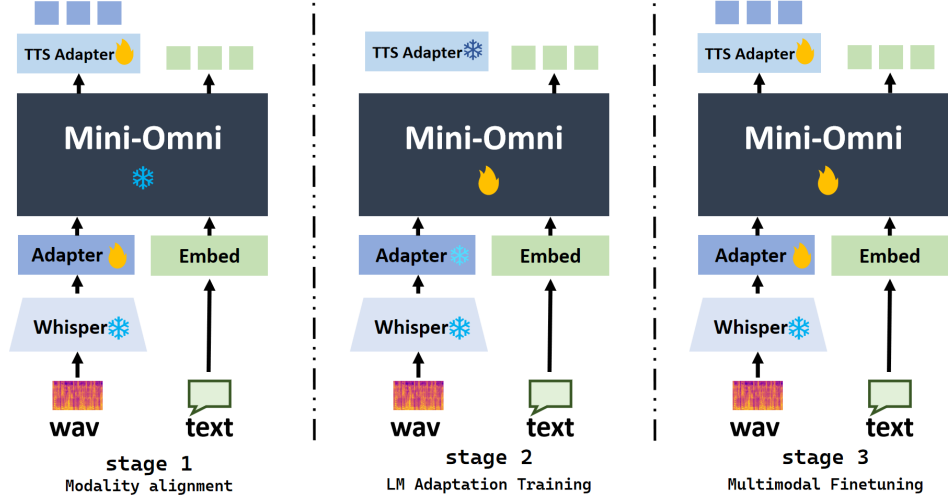
5

Figure 3: Mini-Omni's three-stage training phases: modality expansion, modality adaptation training, and holistic fine-tuning.

speech recognition and synthesis capabilities. **(2) Adaption Training.** Once the new modality is aligned with the text model's input, the adapters are frozen. In this stage, we focus solely on training the model's text capabilities when given audio inputs, as audio output is simply synthesized from text. The model is trained using data from speech recognition, spoken question answering, and text response tasks. **(3) Multi-modal Finetuning.** In the final stage, the entire model is fine-tuned using comprehensive data. At this point, all model weights are unfrozen and trained. Since the primary modality alignment tasks are handled during adapter training, the original model's capabilities are maximally preserved.

Model Input Ids. Given the eight parallel output sequences, the input also requires eight sequences, leading to significant complexity. Therefore, we briefly outline the organization of model inputs here. The model can accept either text or audio inputs, which are placed in the corresponding modality sequences. For audio inputs, the input tokens and Whisper features are transformed into tensors of the same dimension via adapters and then concatenated. Depending on the task, we place the <answer> special token in different positions to guide the model's output, achieving multi-modal output. The organization of some tasks is illustrated in Figure 4. Before being fed into the model, all sequences are summed and averaged to integrate features.
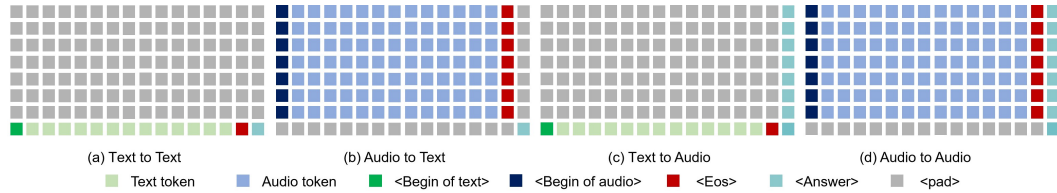

Figure 4: Diagram of the input section of **Mini-Omni** parallel generation. The **<answer>** special token is placed at the end of the sequence to be generated, as determined by the task.

4 Experiments

his section presents the foundational capability test results for **Mini-Omni**. We first describe the training datasets, data processing methods, and hyperparameters. We then evaluate the model's performance on core tasks like speech recognition and provide several use case examples. We will include all relevant experiments in the next version as soon as possible.

6

4.1 Datasets

To establish foundational speech capabilities, we trained the model using three speech recognition datasets totaling approximately 8,000 hours, focusing on speech understanding and synthesis. For text modality, we incorporated 2 million data points from the Open-Orca [OpenOrca] dataset and integrated them with other modalities to preserve textual accuracy. Moss's SFT dataset [Sun et al., 2024] was utilized with zero-shot TTS to synthesize 1.5 million speech QA pairs. To avoid unsuitable code and symbolic outputs, we created the VoiceAssistant-400K dataset with GPT-4o. Datasets are detailed in Table 1. Stage 1 involves ASR data for training speech adapters. Stage 2 uses TextQA and AudioQA for audio/text input and text response training. Stage 3 focuses on multimodal interaction using the audio modality of AudioQA. Final stage training includes annealing and fine-tuning with Voice QA.

Task	Stages	Dataset	Modality	items
ASR	1,2,3	Libritts [Zen et al., 2019]	A1|T1	586 h
		VCTK [datashare, 2024]	A1|T1	44 h
		Multilingual LibriSpeech [Pratap et al., 2020]	A1|T1	8000h
Text QA	2,3	Open-Orca [OpenOrca]	T1|T2	2000K
Audio QA	3	Moss-002-sft-data [Sun et al., 2024]	A1|T1|A2|T2	1500K
voice QA	final	Alpaca-GPT4 [vicgalle, 2024]	A1|T1|A2|T2	55k
		Identity finetune [sayan1101, 2024]	A1|T1|A2|T2	2k
		QAassistant [Mihaiii, 2024a]	A1|T1|A2|T2	27k
		Rlhf [Anthropic, 2024]	A1|T1|A2|T2	367k
		Trivia-singlechoice [Mihaiii, 2024c]	A1|T1|A2|T2	17k
		Trivia-Multichoice [Mihaiii, 2024b]	A1|T1|A2|T2	20k
		OpenAssistant [OpenAssistan, 2024]	A1|T1|A2|T2	2k

Table 1: The datasets and their usage for training **Mini-Omni** are as follows: In the modality notation, T and A represent the text and audio modalities, with subscripts 1 and 2 indicating input or output.

4.2 Training Parameters

Our model is trained on 8 A100 GPUs, utilizing a cosine annealing learning rate scheduler with a minimum learning rate of 4e-6 and a maximum learning rate of 4e-4. Each training epoch consists of 40,000 steps, with batch size 192 for each step. The base language model employs Qwen2-0.5B [Yang et al., 2024a], a transformer architecture with 24 blocks and an internal dimension of 896. The speech encoder uses the Whisper-small encoder, with ASR adapter connected via two-layer MLP, and the TTS adapter extends the original model by adding 6 additional transformer blocks. During fine-tuning, we use learn rate from 4e-6 to 5e-5.

4.3 Experimental Results

We first evaluated the model's performance on ASR tasks to assess its speech understanding capabilities. Basic experiments on speech recognition capabilities were conducted using the four test sets from LibriSpeech [Panayotov et al., 2015]: test-clean, test-other, dev-clean, and dev-other. Results are presented in Table 2, where we compare the accuracy of our adopted speech recognition systems, wav2vec2 [Baevski et al., 2020] and Whisper-small, as well as the VITA [Fu et al., 2024]. The findings indicate that while Mini-Omni's speech recognition performance slightly lags behind Whisper-small's [Radford et al., 2023] decoder, it still achieves an excellent level of audio comprehension.

Method	test-clean	test-other	dev-clean	dev-other
wav2vec2-base [Baevski et al., 2020]	6.0	13.4	-	-
VITA [Fu et al., 2024]	8.14	18.41	7.57	16.57
whisper-small [Radford et al., 2023]	3.4	7.6	-	-
Mini-Omni	4.5	9.7	4.6	9.2

Table 2: Comparison of the model's ASR with the base model used.

4.4 Case Study

Here, we present several cases to demonstrate **Mini-Omni**'s capabilities in speech understanding and reasoning. These examples reveal that speech-based reasoning is somewhat weaker compared to text-based reasoning, highlighting the necessity for batch generation. For more impressive examples, please refer to https://github.com/gpt-omni/mini-omni.
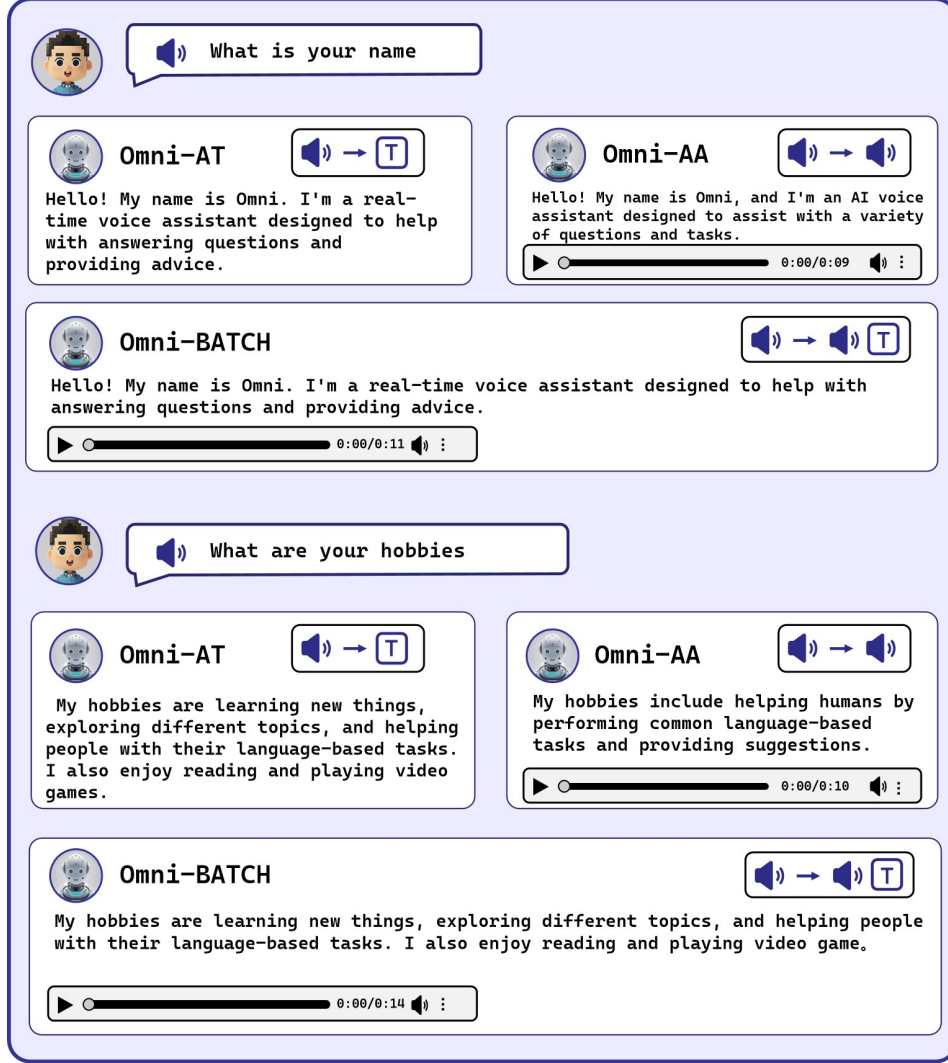

Figure 5: Real streaming output examples of Mini-Omni

5 Conclusion

In this work, we introduce **Mini-Omni**, the first multi-modal model with direct speech-to-speech capabilities. Building on previous approaches that use text-guided speech generation, we propose a parallel text and audio generation method that leverages minimal additional data and modules to rapidly transfer a language model's text capabilities to the audio modality, supporting streaming output interactions with high model and data efficiency. We explore both text-instructed streaming parallel generation and batch parallel generation, which further enhance the model's reasoning ability and efficiency. Our approach successfully addresses challenging real-time dialogue tasks using a model with only 0.5 billion parameters. We have developed the **Any Model Can Talk** method, based on a pre and post-adapter design, to facilitate rapid speech adaptation of other models with minimal additional training. Additionally, we have released the VoiceAssistant-400K dataset for fine-tuning speech output, designed to minimize the generation of code symbols and assist humans in a voice

assistant-like manner. All our data, inference, and training codes will be progressively open-sourced at `https://github.com/gpt-omni/mini-omni`. We hope to provide guidance and support for other work focused on language model speech interaction.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Anthropic. https://huggingface.co/datasets/anthropic/hh-rlhf, 2024.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Qian Chen, Yunfei Chu, Zhifu Gao, Zerui Li, Kai Hu, Xiaohuan Zhou, Jin Xu, Ziyang Ma, Wen Wang, Siqi Zheng, et al. Lauragpt: Listen, attend, understand, and regenerate audio with gpt. *arXiv preprint arXiv:2310.04673*, 2023.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*, 2022.

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

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Nilaksh Das, Saket Dingliwal, Srikanth Ronanki, Rohit Paturi, David Huang, Prashant Mathur, Jie Yuan, Dhanush Bekal, Xing Niu, Sai Muralidhar Jayanthi, et al. Speechverse: A large-scale generalizable audio language model. *arXiv preprint arXiv:2405.08295*, 2024.

datashare. https://datashare.ed.ac.uk/handle/10283/2651, 2024.

Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Xiong Wang, Di Yin, Long Ma, Xiawu Zheng, et al. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*, 2024.

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

Ziyue Jiang, Yi Ren, Zhenhui Ye, Jinglin Liu, Chen Zhang, Qian Yang, Shengpeng Ji, Rongjie Huang, Chunfeng Wang, Xiang Yin, et al. Mega-tts: Zero-shot text-to-speech at scale with intrinsic inductive bias. *arXiv preprint arXiv:2306.03509*, 2023.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718, 2023.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36, 2024.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

meta. llama3.1, 2024. URL https://llama.meta.com/.

Mihaiii. https://huggingface.co/datasets/mihaiii/qa-assistant-2, 2024a.

Mihaiii. https://huggingface.co/datasets/mihaiii/triviamultichoice, 2024b.

Mihaiii. https://huggingface.co/datasets/mihaiii/triviasinglechoice, 2024c.

mixtral. https://mistral.ai/, 2024.

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

openai. https://openai.com/, 2024.

OpenAssistan. https://huggingface.co/datasets/openassistant/oasst1, 2024.

OpenOrca. https://huggingface.co/datasets/open-orca/openorca/.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*, 2020.

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

sayan1101. https://huggingface.co/datasets/sayan1101/identity-finetune-data, 2024.

Hubert Siuzdak. https://github.com/hubertsiuzdak/snac/, 2024.

Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, Qinyuan Cheng, Xiangyang Liu, Hang Yan, Yunfan Shao, Qiong Tang, Shiduo Zhang, et al. Moss: An open conversational large language model. *Machine Intelligence Research*, pages 1–18, 2024.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*, 2023.

vicgalle. https://huggingface.co/datasets/vicgalle/alpaca-gpt4, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Chao Weng, and Helen Meng. Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024b.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*, 2023a.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023b.