VITA-Audio: Fast Interleaved Cross-Modal Token Generation for Efficient Large Speech-Language Model

Zuwei Long[1,†], **Yunhang Shen**[1,†,♠], **Chaoyou Fu**[2,⋆], **Heting Gao**[1], **Lijiang Li**[2]
Peixian Chen[1], **Mengdan Zhang**[1], **Hang Shao**[1], **Jian Li**[1], **Jinlong Peng**[1]
Haoyu Cao[1], **Ke Li**[1], **Rongrong Ji**[3], **Xing Sun**[1,⋆]

[1]Tencent Youtu Lab, [2]Nanjing University, [3]Xiamen University
[†] Equal Contribution [♠] Project Leader [⋆] Corresponding Author

https://github.com/VITA-MLLM/VITA-Audio

Abstract

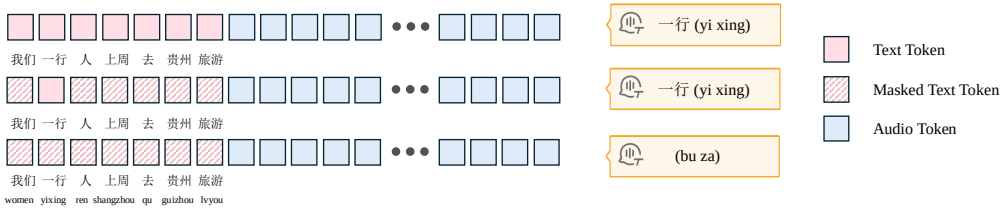
With the growing requirement for natural human-computer interaction, speech-based systems receive increasing attention as speech is one of the most common forms of daily communication. However, the existing speech models still experience high latency when generating the first audio token during streaming, which poses a significant bottleneck for deployment. To address this issue, we propose VITA-Audio, an end-to-end large speech model with fast audio-text token generation. Specifically, we introduce a lightweight Multiple Cross-modal Token Prediction (MCTP) module that efficiently generates multiple audio tokens within a single model forward pass, which not only accelerates the inference but also significantly reduces the latency for generating the first audio in streaming scenarios. In addition, a four-stage progressive training strategy is explored to achieve model acceleration with minimal loss of speech quality. To our knowledge, VITA-Audio is the first multi-modal large language model capable of generating audio output during the first forward pass, enabling real-time conversational capabilities with minimal latency. VITA-Audio is **fully reproducible** and is trained on open-source data **only**. Experimental results demonstrate that our model achieves an inference speedup of $3 \sim 5\times$ at 7B parameter scale, but also significantly outperforms open-source models of similar model size on multiple benchmarks for automatic speech recognition (ASR), text-to-speech (TTS), and spoken question answering (SQA).

1 Introduction

Real-time speech systems have become a crucial research focus for enabling natural dialogue. Traditional speech systems predominantly adopt a modular design that decomposes real-time speech processing into three discrete components: automatic speech recognition (ASR), large language models (LLMs), and text-to-speech (TTS) [54, 32, 69]. However, this cascaded approach suffers from cumulative latency, loss of paralinguistic information (*e.g.*, emotional prosody, rhythm) during modality conversion, and error accumulation between modules, substantially lowering the practical utility of cascaded architectures in real-time interactive scenarios. To address the limitations of traditional methods, many recent studies have adopted an end-to-end approach to handle inputs and outputs of the model [58, 10, 23]. These methods directly input speech into LLMs through an audio encoder and then synthesize speech response with discrete tokens [66] or LLM hidden states [56].

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

(a) The visualization of the attention maps between text tokens and audio tokens.



(b) The transcription results of the generated audio into text under different attention masks.

Figure 1: (a) The audio sequence generated by the speech model exhibits a strong correlation with the corresponding text tokens. (b) With irrelevant text tokens being masked out, the model is still able to generate the correct audio, and the pronunciation remains contextually appropriate. However, if all text tokens are masked, the model outputs random audio. This suggests that the hidden states from the LLM include sufficient contextual information for generating the corresponding audio tokens. Consequently, the mapping from text hidden states to audio tokens is accomplished using relatively simple modules, without the need for the extensive semantic modeling typically required by LLMs.

While existing end-to-end speech models generate output in a streaming fashion to reduce the response latency, their first token delay is still high. Specifically, current speech models cannot directly deliver the first streaming audio chunk upon completing the first LLM forward pass, *i.e.*, decoding the first text token. In the applications requiring high real-time performance, this delay poses a significant bottleneck to the deployment of LLMs for speech processing. This prompts a pertinent question:

How can we achieve more real-time audio generation within end-to-end speech models?

To explore this issue, we visualized the hidden states of the final decoder layer of the speech model. As shown in Fig. 1a, the audio tokens generated by the speech model show increased attention to the text tokens they correspond to. As the generation of audio tokens progresses, the text tokens attended by the new audio token advance accordingly. This finding is also reported in many literature on attention-based speech systems [7, 34]

In Fig. 1b, we show a Chinese sentence with a homograph "行" as an example. The pronunciation of this character can be /xing/ or /hang/ depending on its context. In Chinese, 'yihang' emphasizes spatial arrangement (e.g. a row/line of egrets), while 'yixing' focuses on the concept of a group or unit (e.g. a group/party of people). Our speech model correctly decides the character's pronunciation to be the former, given the hidden states of historical inputs. We then modify the model inference process by masking out all text hidden states, except the one corresponding to the token "一行"(/yixing/) before generating its corresponding audio tokens. This modification prevents subsequently generated audio tokens from directly attending to other text tokens, although they can still attend to previously generated audio tokens. We find that the subsequently generated audio tokens accurately produce the sounds as "yixing", which remains contextually appropriate. The same observation also holds for other non-homograph tokens. We therefore argue that **the hidden states from the LLM include sufficient contextual information for generating its corresponding audio tokens, and attending to additional texts is unnecessary**. Finally, we experiment with masking out all text tokens. This time, the generated audio fails to align with its text and sounds like random non-speech babbles even though the model has access to the previously generated audio tokens.

These findings suggest that the speech model learns to primarily focus on the small span of corresponding text hidden states without heavily modeling the semantic space of the entire text and

Table 1: Comparison of recent speech models, VITA-Audio leverages the hidden state to enhance model performance, adopts an end-to-end architecture, and achieves zero audio token delay.

Model	Audio Token Delay	Leveraging Hidden States	End-to-End
Freeze-Omni [56]	Text Length	✓	✗
MinMo [9]	5	✓	✗
Mini-Omni [58]	7	✗	✓
Moshi [19]	1	✗	✓
GLM-4-Voice [66]	13	✗	✓
LUCY [28]	7	✗	✓
VITA-Audio	0	✓	✓

audio sequence. This discovery instills confidence that we can learn the simple mapping relationship between text hidden states and audio tokens with relatively simple modules and without relying on the extensive semantic modeling of LLMs.

In this paper, we introduce VITA-Audio, a lightweight framework that uses separate efficient modules, named Multiple Cross-modal Token Prediction (MCTP), to efficiently generate audio responses from text embeddings and LLM hidden states. This approach enables obtaining both text tokens and an audio chunk in a single LLM forward pass, achieving zero delay in audio tokens. A comparison of the delay of the first audio token is presented in Table 1 , where we define "audio token delay" as the number of additional LLM forward steps required to generate the first audio token after the first LLM forward pass. We distinguish this delay from "audio generation delay" which is the number of additional LLM forward passes to generate a meaningful and consistent chunk of audio. VITA-Audio has both zero audio token delay and zero audio generation delay.

To this end, through a four-stage progressive training strategy, we construct a set of lightweight yet powerful MCTP modules, which predict **10** audio tokens directly from historical inputs and LLM hidden states without requiring additional LLM forward passes, thus significantly enhancing the model's inference speed without sacrificing audio quality.

In summary, our main contributions are as follows.

- We introduce VITA-Audio, the first end-to-end speech model capable of generating audio during the first forward pass. Leveraging audio generation without relying on extensive text semantic modeling capabilities, VITA-Audio designs lightweight MCTP modules to generate decodable audio token chunks with zero audio token delay, thus overcoming the real-time limitations in traditional cascaded models and existing end-to-end methods.
- VITA-Audio achieves remarkable end-to-end inference acceleration by generating ten audio tokens in a single forward pass, resulting in $3 \sim 5\times$ speedup when implemented on a 7B LLM while preserving the ability of high-quality speech synthesis.
- We fully release VITA-Audio to the open-source community. Although VITA-Audio is trained on open-source data only, comprehensive evaluations reveal that VITA-Audio achieves the state-of-the-art performance on multiple benchmarks for ASR, TTS, and SQA tasks, outperforming existing models in both efficiency and accuracy, especially the open-source ones with a similar parameter scale, therefore setting a new standard for real-time speech-to-speech models.

2 Related Work

Large language models (LLMs) have revolutionized human-computer interaction with advanced natural language processing. Extending these capabilities to speech—a natural communication modality—has become a key research focus. Traditional speech interaction systems [54, 32, 69] typically adopt a cascaded architecture, combining separate ASR, LLM, and TTS modules. However, this approach suffers from increased latency, loss of paralinguistic cues, and error propagation.

Recent work [24, 49] has improved integration by connecting audio encoders to LLMs via trainable adapters, but still relies on independent TTS modules. To address this, some methods incorporate

LLM hidden states into audio decoders. For example, Llama-Omni [23] uses a non-autoregressive transformer to predict audio tokens from upsampled LLM states, while Freeze-omni [56] freezes the LLM and combines autoregressive and non-autoregressive decoders. Minmo [9] integrates a language model with CosyVoice2 [22] for mixed speech-text processing.

End-to-end models further unify TTS within LLMs, enabling direct text and speech generation. These models follow either parallel or interleaved audio-text modeling. In the parallel modeling paradigm, the model uses different heads to process hidden states, generating both text and multiple audio tokens [10, 19]. Since the input to the LLM is altered during autoregression, maintaining the original capabilities of the LLM presents significant challenges. To perform inference without large-scale pretraining, Mini-Omni [58] and LUCY [28] rely on batch parallel decoding to preserve the inference capability of the LLM.

Compared to parallel-paradigm models, interleave-paradigm models appear to better preserve language capabilities, as suggested by the performance comparison on spoken question-answering benchmarks [66]. We attribute this difference to the fact that parallel-paradigm models use an average of text and audio representations as input, which significantly diverges from the inputs used during pretraining. However, interleave-paradigm models face a latency issue due to their sequential prediction of audio tokens, especially when the audio token rate is high.

VITA-Audio leverages the strengths of these architectures by adopting the interleaved modeling paradigm and introducing MCTP for audio generation. The former maximally preserves the LLM's language ability, and the latter reduces inference latency by generating multiple audio tokens in a single forward pass.

3 Method

3.1 Overview

As illustrated in Fig. 2, VITA-Audio consists of four major components: an audio encoder, an audio decoder, a large language model backbone, and a set of Cross-modal Token Prediction (MCTP) modules. The audio signal is first processed by the audio encoder, whose output is then fed into the LLM for further processing. During each forward pass, the LLM alternately generates text and audio tokens. The hidden states from the final layer of LLM, along with the embedding of the predicted token, are provided as input to the MCTP modules. The historical input tokens, the tokens predicted by the LLM, and by the MCTP modules are concatenated to form the inputs to the next LLM forward pass. Finally, the audio tokens generated by both the LLM and the MCTP modules are aggregated and passed to the audio decoder to generate the final audio output.

3.2 Multiple Cross-Modal Token Prediction (MCTP) Module

As described in Section 1, the text and speech modalities exhibit a monotonic alignment pattern. This cross-modal alignment allows us to avoid complex modeling of the semantic latent space and to focus on learning a simple text-to-speech mapping relationship, which we propose to use lightweight modules to learn. In the preliminary experiments, we use a few lightweight Transformer blocks to predict multiple audio tokens from LLM hidden states, and embed the predicted tokens into the LLM's autoregressive inference.

Standard autoregressive modeling can be formulated as:

$$p_t(Y_{t-1}, ..., Y_0) \equiv P[Y_t|Y_{t-1}, ..., Y_0], \tag{1}$$

where Y_t denotes the predicted audio token at time step t, and p_t represents the conditional probability distribution based on the historical sequence. When extended to multi-step prediction, *i.e.*, predicting the $i\text{-}th$ audio token at time step $t + i$, the formulation becomes as:

$$p_{t+i}(Y_{t-1}, ..., Y_0) \equiv \widetilde{P}[Y_{t+i}|Y_{t-1}, ..., Y_0]. \tag{2}$$

At this point, there is a significant deviation in the consistency of the distribution between $\widetilde{P}$ and P. As i increases, the difference between the two distributions will progressively widen, resulting in a growing accumulation of errors and leading to poor mapping between text and audio.

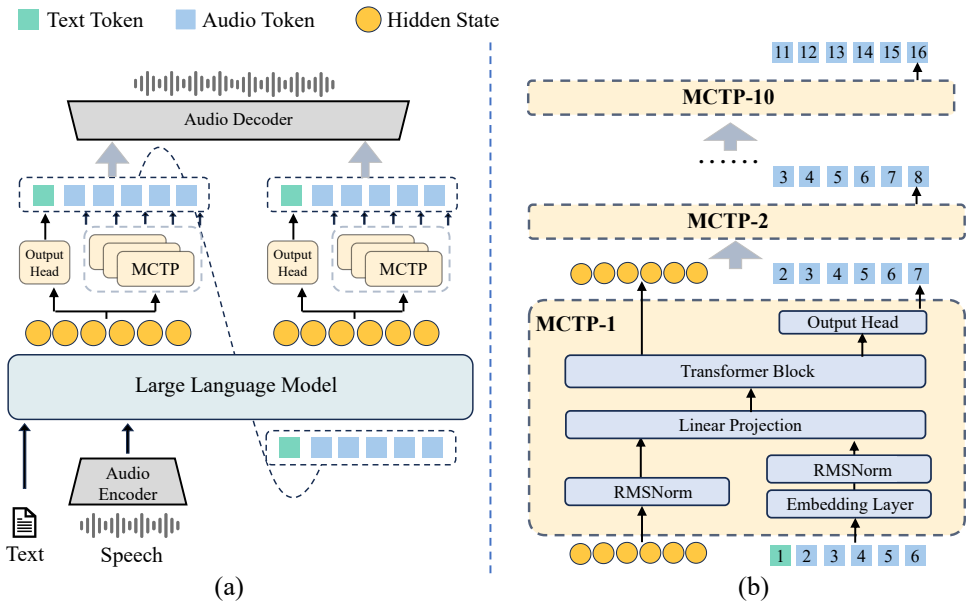


Figure 2: Architecture overview. (a) VITA-Audio is an end-to-end large speech model equipped with 10 light-weight Multiple Cross-modal Token Prediction (MCTP) modules that enable speech generation with extremely low latency. As shown in Fig. 1, we observe that the hidden states of certain text tokens in the LLM backbone contain sufficient semantic information for generating the corresponding audio tokens, which means that it is unnecessary to attend to additional text tokens when generating audio. Thus, we propose to utilize a set of light-weight MCTP modules to model the mapping from LLM hidden states to the audio tokens. (b) The details of the MCTP modules. Our MCTP module has a light-weight architecture, which enables it to finish one forward pass within 0.0024 seconds (11% of the LLM backbone). The MCTP module is capable of generating 10 audio tokens from the LLM hidden states and the text embedding, and the generated audio tokens can be decoded by the audio decoder directly. The utilization of MCTP modules enables VITA-Audio to generate audio responses in one LLM forward pass, which achieves extremely fast generation speed.

To address this issue, we adopt a cascaded prediction architecture. Specifically, the hidden states and output sequence from the preceding modules are employed as joint input conditions for the subsequent modules:

$$p_{t+i}(Y_{t-1},\ldots,Y_0) \equiv \widetilde{P}[Y_{t+i}|Y_{t-1},\ldots,Y_0,h_{t+i-1},o_{t+i-1},\ldots,o_t], \tag{3}$$

where h_{t+i-1} and o_{t+i-1} represent the hidden state and output sequence of the preceding module, respectively. By introducing progressively updated contextual information, modules can achieve incremental optimization of cross-modal mapping, ensuring accurate modality synchronization at each time step.

Inspired by DeepSeek V3 [18], we adopt an isomorphic Multi-Token Prediction (MTP) framework to construct our MCTP module. Unlike DeepSeekV3 and speculative decoding [38], where the former focuses on improving training and the latter requires verification to ensure that the sampling distribution matches exactly with that of the original model, we use the MCTP module for audio-text mapping, presumably a simpler task than semantic modeling. As a result, we require a comparatively small amount of text data to train our model.

Since the embedding layer and output heads are shared with the LLM, the audio tokens generated by the MCTP module are directly incorporated into the autoregressive process of the LLM. As illustrated in Fig. 2, the hidden states and output token, from the LLM or the preceding MCTP module, are concatenated with the input tokens and fed into a Transformer block for next-step processing. The resulting hidden states and token are then passed to the next MCTP module. Upon completion of a forward pass, the audio tokens generated by either the LLM or the MCTP modules are aggregated as the input sequence for the subsequent LLM forward iteration.

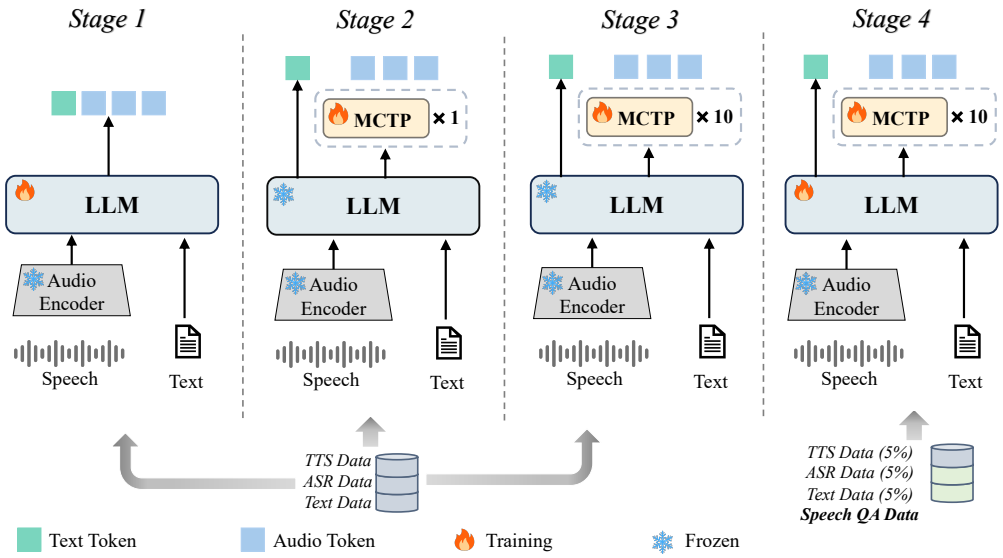


Figure 3: Training pipeline of VITA-Audio. The first stage (Audio-Text Alignment) enhances the LLM by extending its audio modeling capability through large-scale speech pre-training. The second stage (Single MCTP module Training) connects an MCTP module with the LLM to predict one subsequent token based on the input tokens and the LLM's hidden states. The third stage (Multiple MCTP Modules Training) increases the number of MCTP modules in the model to predict more tokens in each model forward. The last stage (Supervised Fine-tuning) provides the speech-to-speech capability to the model by optimizing it on the large-scale speech QA dataset.

3.3 Training

3.3.1 Data Construction

VITA-Audio is trained exclusively on open-source datasets, integrating multi-domain and multi-language speech data resources. The training dataset encompasses a diverse range of sources. Detailed descriptions of the datasets used at each stage are provided in Table E1 in the Appendix.

All training data are uniformly packed into sequences of fixed length (8K tokens), an approach that enables effective training on samples of varying lengths [47]. We reinitialize the positional embeddings and attention masks for all packed samples to ensure that the model attends exclusively to tokens within the same original sample. This processing strategy not only eliminates potential artifacts introduced by data concatenation but also significantly enhances training stability and reduces computational overhead.

3.3.2 Training Pipline

For VITA-Audio to output a consistent sequence of audio tokens in a single forward pass, each MCTP module must model a distinct distribution. As a result, training all the MCTP modules simultaneously becomes a challenging task, especially when the number of modules is large, due to potentially misaligned optimization objectives. We propose a four-stage training strategy, as shown in Fig. 3, to progressively equip the MCTP modules with the ability to map text to its audio, thereby reducing the difficulty of their convergence. Further training details are provided in Sec. B of the Appendix.

3.4 Inference

In order to address diverse scenarios, four distinct inference paradigms have been designed as shown in Fig. 4.

For the ASR and TTS tasks, we propose VITA-Audio-Turbo. In each forward pass, the LLM generates one token, followed by the generation of ten tokens by the MCTP modules. This paradigm

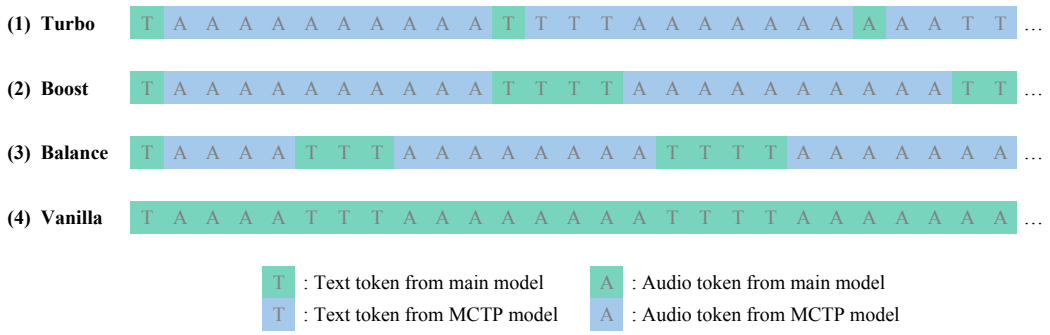


Figure 4: The four text-audio interleaved inference modes are illustrated as follows: 1) Turbo: As the fastest inference mode, it generates 1 token by the main model and 10 additional tokens via MCTP in each forward pass. To ensure that a valid audio chunk is decoded after the first forward pass, the first generated 11 tokens are split into 1 text token and 10 audio tokens. Then, the Turbo mode iteratively generates 4 text tokens and 10 audio tokens in the following forward. 2) Boost: To enhance the quality of text tokens, Boost mode follows the text-audio cyclic pattern of Turbo mode, with the main model generating every text token and MCTP generating every audio token. 3) Balance: To keep a balanced text-audio ratio, *i.e.*, $1:2$, the balance mode further changes the text-audio cyclic pattern of the Boost mode. Specifically, the balance mode sequentially generates 1 text token from the main model, 4 audio tokens (2 tag tokens mark the beginning and end of audios, and 2 common tokens denote the audio content) from MCTP, 3 text tokens from the main model, 8 text tokens (2 tag tokens mark the beginning and end of audios, and 6 common tokens denote the audio content) from MCTP, and then iteratively generates 4 text tokens from the main model and 10 audio tokens (2 tag tokens mark the beginning and end of audios, and 8 common tokens denote the audio content) from MCTP. 4) Vanilla: As the slowest inference mode, Vanilla mode follows the text-audio cyclic pattern of Balance mode, with the main model generating every token.

is the most efficient among the options; however, the performance of speech dialogue tasks degrades due to the need to predict the text token.

For speech dialogue tasks, we introduce VITA-Audio-Boost and VITA-Audio-Balance. Their main difference lies in audio token generation: VITA-Audio-Boost generates eight directly decodable audio tokens in the first forward pass, whereas VITA-Audio-Balance adheres to a strict $1:2$ text-to-audio token ratio for enhanced speech quality. The latter requires two forward passes to generate enough audio tokens for decoding. To optimize model performance, distinct models were trained for each of the two modes.

VITA-Audio-Vanilla is designed for scenarios that require higher language performance. It generates tokens solely using the main LLM, sacrificing efficiency but offering a slight performance boost.

4 Experiment

4.1 Experiment Settings

We use the Qwen2.5-7B-Instruct [45] as the pre-trained text LLM. The initial version of VITA-Audio utilizes the speech tokenizer and speech decoder in GLM-4-Voice [66], which effectively captures semantic information at an ultra-low bitrate. In the second version, *i.e.*, VITA-Audio-Plus further replaces the GLM-4-Voice tokenizer with SenseVoiceSmall [1] and an MLP-based adapter. The detail comparison between VITA-Audio and VITA-Audio-Plus is listed in Table E2.

4.2 Evaluation on Spoken Question Answering

We evaluate the spoken question answering capability of VITA-Audio on three public English datasets: Web-Questions [5], Llama-Question [42], and TriviaQA [33]. Two evaluation methods are employed: S→T, where the text responses generated by the model are evaluated directly, and S→S, where the model's speech responses are transcribed using Whisper [46] before evaluation.

Table 2: Results on Spoken Question Answering (SQA) benchmarks. "Sx" denotes the x-th training stage of speech models.

Model		#Params	Llama Question		TriviaQA		Web Question		Mean	
			S → T	S → S	S → T	S → S	S → T	S → S	S → T	S → S
Proprietary Models										
MinMo	[9]	7B	78.9	64.1	48.3	37.5	55.0	39.9	60.7	47.2
Open-source Models										
Moshi	[19]	7B	62.3	21.0	22.8	7.3	26.6	9.2	37.2	12.5
GLM-4-Voice	[66]	9B	64.7	50.7	39.1	26.5	**55.0**	39.9	45.3	31.0
LUCY (S2)	[28]	7B	59.6	51.0	23.2	18.2	26.6	18.2	36.5	29.1
VITA-Audio-Boost		7B	68.7	60.3	30.5	29.3	32.9	30.4	44.0	40.0
VITA-Audio-Vanilla		7B	71.3	66.3	31.9	30.1	33.5	31.4	45.6	42.6
VITA-Audio-Plus-Boost		7B	**76.3**	64.6	43.6	39.5	44.2	40.0	54.7	48.0
VITA-Audio-Plus-Vanilla		7B	75.6	**68.0**	**45.9**	**42.7**	45.0	**41.7**	**55.5**	**50.8**

Table 3: Results on Text to Speech (TTS) Benchmarks. "Sx" denotes the x-th training stage.

Model		Seed-TTS			LibriTTS
		test-zh	*test-en*	*test-hard*	*test-clean*
		CER (%) ↓	WER (%) ↓	WER (%) ↓	WER (%) ↓
Seed-TTS	[2]	1.12	2.25	7.59	–
CosyVoice	[21]	3.63	4.29	11.75	2.89
CosyVoice2	[22]	1.45	2.57	**6.83**	2.47
VITA-1.5 (S3)	[25]	8.44	2.63	–	–
GLM-4-Voice	[66]	2.91	2.10	–	5.64
VITA-Audio-Turbo (S1)		1.18	1.92	10.58	1.96
VITA-Audio-Turbo (S2)		**0.96**	1.92	9.72	1.98
VITA-Audio-Turbo (S3)		1.05	**1.77**	9.86	1.99
VITA-Audio-Turbo (S4)		1.07	2.26	10.08	2.08
VITA-Audio-Plus-Boost		1.32	2.21	12.05	2.21
VITA-Audio-Plus-Vanilla		1.13	1.85	10.21	**1.89**

We compare VITA-Audio with the latest speech models that have comparable parameter sizes, and the results are shown in Table 2. Our model demonstrates superior performance in the S→T task and achieves SOTA results in the S→S setup. It is particularly noteworthy that the training approach of VITA-Audio ensures minimal degradation between S→T and S→S, with a performance drop of only 9%. This indicates that VITA-Audio achieves high-quality alignment between text and speech modalities, with benefits extending beyond processing speed alone.

4.3 Evaluation on Fundamental Speech Competence

TTS We evaluate the TTS performance of VITA-Audio on Seed-TTS [2] and LibriTTS [65] benchmarks. We use Whisper-Large-V3 [46] and Paraformer[29] to transcribe into text the generated English and Chinese speech, respectively.

We present the results of VITA-Audio at each stage in Table 3. In these results, the output of VITA-Audio's first stage (S1) consists of text tokens directly generated by the LLM, while the outputs of the second (S2), third (S3), and fourth (S4) stages are alternately generated by both the LLM and the MCTP module. The experiments demonstrate that VITA-Audio outperforms other open-source models with a similar number of parameters. Additionally, it should be noted that, despite using ten MCTP modules for accelerated inference, VITA-Audio's TTS capabilities are largely preserved throughout the training process, further validating the effectiveness of the MCTP module in aligning text and audio.

ASR We evaluate the ASR performance of the four stages of VITA-Audio on WenetSpeech [67], AIshell [6], LibriSpeech [43], and Fleurs [15], and a subset of the results are reported in Table 4.

Table 4: Results on Automatic Speech Recognition (ASR) Benchmarks. "Sx" denotes the x-th training stage. Compared to other methods, **VITA-Audio is trained with open-source data only**.

Model		WenetSpeech		AIShell	LibriSpeech	
		test_meeting ↓	test_net ↓	test ↓	test-clean ↓	test-other ↓
Qwen2-Audio-base	[12]	8.40	7.64	1.52	**1.74**	**4.04**
Baichuan-Audio-base	[37]	13.28	10.13	1.93	3.02	6.04
VITA-1.5 (S3)	[24]	10.0	8.4	2.2	3.4	7.5
Freeze-Omni	[56]	13.46	11.8	2.48	3.82	9.79
LUCY (S1)	[28]	10.42	8.78	2.40	3.36	8.05
Step-Audio-chat	[49]	10.83	9.47	2.14	3.19	10.67
Qwen2.5-Omni	[59]	7.71	**6.04**	**1.13**	2.37	4.21
VITA-Audio-Vanilla		17.34	13.45	4.46	2.98	8.07
VITA-Audio-Plus-Boost		9.38	8.97	4.72	3.13	7.07
VITA-Audio-Plus-Vanilla(S1)		**6.68**	6.59	1.51	1.91	4.29
VITA-Audio-Plus-Vanilla		7.12	6.90	1.94	2.00	4.60

Table 5: Boostup Ratio under Different Inference Paradigms.

Mode	Model Size	#GPU	Total Second ↓	Token Per Second ↓	Speedup ↑
Vanilla	0.5B	1	53.89	76.00	1.00 ×
Boost			20.65	198.35	2.61 ×
Balance			20.71	197.78	2.60 ×
Turbo			11.83	346.24	4.56 ×
Vanilla	7B	1	63.38	64.62	1.00 ×
Boost			23.97	170.88	2.64 ×
Balance			23.94	171.09	2.64 ×
Turbo			13.43	304.99	4.72 ×
Vanilla	72B	2	255.13	16.05	1.00 ×
Boost			84.98	48.20	3.00 ×
Balance			85.13	48.11	3.00 ×
Turbo			39.5	103.60	6.46 ×

More detailed results can be found in Table E3 and Table E4 in the Appendix. The results for other works are partially reproduced from their respective original works for comparison.

It can be observed that VITA-Audio-plus-vanilla demonstrates highly competitive performance across various benchmarks. Moreover, VITA-Audio-plus-Boost achieves remarkably fast inference speed while still maintaining strong overall performance.

4.4 Evaluation of Latency

Inference Speedup Efficient mapping between text and speech is the core of VITA-Audio. To demonstrate its effectiveness, we compare the inference time across different modes of VITA-Audio for various model sizes. Specifically, we evaluate the inference speed on GPUs capable of 148 TFLOPS under bfloat16 precision, with the output fixed at 4096 tokens, and record the total time as the model's inference time. All models, regardless of size, are randomly initialized, and this initialization do not affect the inference time measurements. We use Transformers [57] and FlashAttention-2 [17].

As mentioned in Section 3.4, VITA-Audio-Vanilla only uses the main model for output; VITA-Audio-Turbo uses both the main model and all the MCTP modules during each forward pass; and VITA-Audio-Boost and VITA-Audio-Balance progressively increase the number of MCTP modules used to ensure higher accuracy.

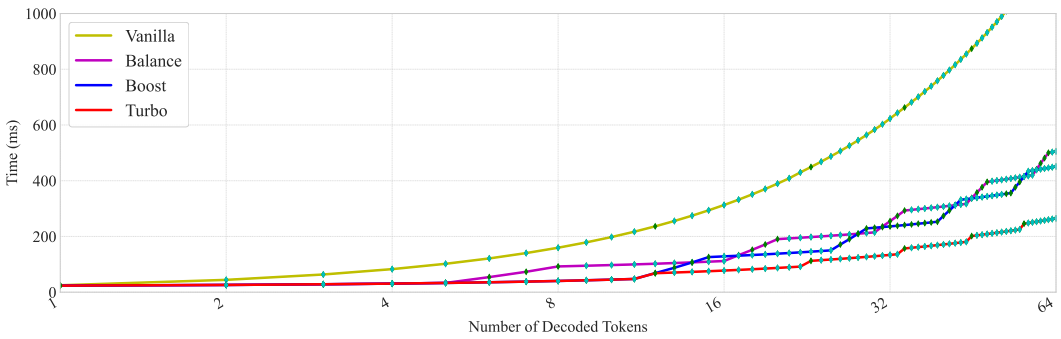


Figure 5: Token generation speed curves of four text-audio interleaved modes.

As shown in Table 5, we present a comparison of the time consumption for different inference modes and model sizes. Noted that the time consumption does not include the cost of the audio encoder and audio decoder. We observe that in VITA-Audio-Turbo, a speedup of approximately $5\times$ is achieved across models ranging from 0.5B to 72B, greatly enhancing the output token throughput. VITA-Audio-Boost also achieves around $3\times$ acceleration across various sizes, resulting in a desirable performance for real-time speech dialogue systems. For example, 72 B VITA-Audio generates approximately 40 tokens per second, which, excluding generated text tokens, corresponds to roughly three seconds of audio and associated text when using a 12.5Hz speech tokenizer. This performance is sufficiently fast for human-computer interaction.

Latency In human-computer interaction, latency is a crucial metric, as it determines whether users can interact with the model in real-time. Given that most speech models support streaming output, the key to reducing perceived latency lies in shortening the time required to generate the first chunk of audio.

We visualize the timeline of model decoding phrase in Fig. 5. The green marks denote the tokens generated by the main model, and the blue marks are the tokens generated by MCTP modules. We set the number of prefiil tokens to 32. And Fig. 5 shows that VITA-Audio-Turbo completes the generation of the first audio chunk in about 50 ms, while VITA-Audio-Vanilla requires about 220 ms. VITA-Audio-Boost and VITA-Audio-Balance generate fewer audio tokens in the first forward and more text audio tokens in the following forward. Thus, they are slower than VITA-Audio-Turbo but still significantly faster than VITA-Audio-Vanilla.

Thanks to the advantage of zero audio generation delay, VITA-Audio produces multiple audio tokens in the first forward pass, allowing the first audio token chunk to be generated during the initial forward pass, which can then be used for decoding. This significantly reduces the perceived delay. In the experimental environment previously mentioned, VITA-Audio reduces the time to generate the first audio token chunk from 236 to 53 ms, as shown in Table E7.

5 Conclusion

In this paper, we introduce VITA-Audio, a lightweight framework that uses separate efficient modules, named Multiple Cross-modal Token Prediction (MCTP) modules, to efficiently generate audio responses from text embeddings and LLM hidden states. MCTP learns the simple mapping relationship between text hidden states and audio tokens with relatively simple modules and without relying on the extensive semantic modeling of LLMs. Our model achieves new state-of-the-art performance on multiple benchmarks for ASR, TTS, and SQA tasks, outperforming existing models in efficiency and accuracy, especially the open-source ones with a similar parameter scale. Therefore, it sets a new standard for real-time speech-to-speech models.

Acknowledgments

This work is partially funded by National Natural Science Foundation of China (Grant No. 62506158 and No. 62441234), and CCF-Tencent Rhino-Bird Open Research Fund.

References

[1] Keyu An, Qian Chen, Chong Deng, Zhihao Du, Changfeng Gao, Zhifu Gao, Yue Gu, Ting He, Hangrui Hu, Kai Hu, Shengpeng Ji, Yabin Li, Zerui Li, Heng Lu, Haoneng Luo, Xiang Lv, Bin Ma, Ziyang Ma, Chongjia Ni, Changhe Song, Jiaqi Shi, Xian Shi, Hao Wang, Wen Wang, Yuxuan Wang, Zhangyu Xiao, Zhijie Yan, Yexin Yang, Bin Zhang, Qinglin Zhang, Shiliang Zhang, Nan Zhao, and Siqi Zheng. FunAudioLLM: Voice Understanding and Generation Foundation Models for Natural Interaction Between Humans and LLMs. *arXiv:2407.04051*, 2024. 7, 4

[2] Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, Mingqing Gong, Peisong Huang, Qingqing Huang, Zhiying Huang, Yuanyuan Huo, Dongya Jia, Chumin Li, Feiya Li, Hui Li, Jiaxin Li, Xiaoyang Li, Xingxing Li, Lin Liu, Shouda Liu, Sichao Liu, Xudong Liu, Yuchen Liu, Zhengxi Liu, Lu Lu, Junjie Pan, Xin Wang, Yuping Wang, Yuxuan Wang, Zhen Wei, Jian Wu, Chao Yao, Yifeng Yang, Yuanhao Yi, Junteng Zhang, Qidi Zhang, Shuo Zhang, Wenjie Zhang, Yang Zhang, Zilin Zhao, Dejian Zhong, and Xiaobin Zhuang. Seed-TTS: A Family of High-Quality Versatile Speech Generation Models. *arXiv:2406.02430*, 2024. 8

[3] Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. LongAlign: A Recipe for Long Context Alignment of Large Language Models. 2024. 1, 3

[4] Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongWriter: Unleashing 10,000+ Word Generation from Long Context LLMs. 2024. 1, 3

[5] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic Parsing on Freebase from Question-Answer Pairs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013. 7

[6] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. AISHELL-1: An Open-Source Mandarin Speech Corpus and a Speech Recognition Baseline. *Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, 2018. 8, 1, 3, 5

[7] William Chan, Navdeep Jaitly, Quoc V Le, and Vinyals Google Brain. Listen, Attend and Spell. 2015. 2

[8] Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2021. 1, 3

[9] Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, Yabin Li, Xiang Lv, Jiaqing Liu, Haoneng Luo, Bin Ma, Chongjia Ni, Xian Shi, Jialong Tang, Hui Wang, Hao Wang, Wen Wang, Yuxuan Wang, Yunlan Xu, Fan Yu, Zhijie Yan, Yexin Yang, Baosong Yang, Xian Yang, Guanrou Yang, Tianyu Zhao, Qinglin Zhang, Shiliang Zhang, Nan Zhao, Pei Zhang, Chong Zhang, and Jinren Zhou. MinMo: A Multimodal Large Language Model for Seamless Voice Interaction. *arXiv:2501.06282*, 2025. 3, 4, 8, 5

[10] Wenxi Chen, Ziyang Ma, Ruiqi Yan, Yuzhe Liang, Xiquan Li, Ruiyang Xu, Zhikang Niu, Yanqiao Zhu, Yifan Yang, Zhanxun Liu, Kai Yu, Yuxuan Hu, Jinyu Li, Yan Lu, Shujie Liu, and Xie Chen. SLAM-Omni: Timbre-Controllable Voice Interaction System with Single-Stage Training. *arXiv:2412.15649*, 2024. 1, 4

[11] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. LongLoRA: Efficient Fine-Tuning of Long-Context Large Language Models. 2023. 1, 3

[12] 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. Qwen2-Audio Technical Report. *arXiv:2407.10759*, 2024. 9, 4, 5

[13] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 6

[14] Together Computer. Long Data Collections, 2023. 1, 3

[15] Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. FLEURS: FEW-Shot Learning Evaluation of Universal Representations of Speech. *2022 IEEE Spoken Language Technology Workshop, SLT 2022 - Proceedings*, 2023. 8, 4

[16] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM, 2023. 1, 3

[17] Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. *The International Conference on Learning Representations (ICLR)*, 2023. 9

[18] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. DeepSeek-V3 Technical Report. 2024. 5

[19] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: A Speech-Text Foundation Model for Real-Time Dialogue. *arXiv:2410.00037*, 2024. 3, 4, 8, 5

[20] Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. AISHELL-2: Transforming Mandarin ASR Research into Industrial Scale. *arXiv:1808.10583*, 2018. 3, 5

[21] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, Zhifu Gao, and Zhijie Yan. CosyVoice: A Scalable Multilingual Zero-Shot Text-To-Speech Synthesizer Based on Supervised Semantic Tokens. *arXiv:2407.05407*, 2024. 8

[22] Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, Fan Yu, Huadai Liu, Zhengyan Sheng, Yue Gu, Chong Deng, Wen Wang, Shiliang Zhang, Zhijie Yan, and Jingren Zhou. CosyVoice 2: Scalable Streaming Speech Synthesis with Large Language Models. *arXiv:2412.10117*, 2024. 4, 8

[23] Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. LLaMA-Omni: Seamless Speech Interaction with Large Language Models. *arXiv:2409.06666*, 2024. 1, 4, 5

[24] Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Xiong Wang, Di Yin, Long Ma, Xiawu Zheng, Ran He, Rongrong Ji, Yunsheng Wu, Caifeng Shan, and Xing Sun. VITA: Towards Open-Source Interactive Omni Multimodal LLM. *arXiv:2408.05211*, 2024. 3, 9

[25] Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Yangze Li, Zuwei Long, Heting Gao, Ke Li, Xiawu Zheng, Rongrong Ji, Xing Sun, Caifeng Shan, and Ran He. VITA-1.5: Towards GPT-4o Level Real-Time Vision and Speech Interaction. *arXiv:2501.01957*, 2025. 8, 4, 5

[26] Yihui Fu, Luyao Cheng, Shubo Lv, Yukai Jv, Yuxiang Kong, Zhuo Chen, Yanxin Hu, Lei Xie, Jian Wu, Hui Bu, Xin Xu, Jun Du, and Jingdong Chen. AISHELL-4: An Open Source Dataset for Speech Enhancement, Separation, Recognition and Speaker Diarization in Conference Scenario. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2021. 1, 3

[27] Daniel Galvez, Greg Diamos, Juan Ciro, Juan Felipe Cerón, Keith Achorn, Anjali Gopi, David Kanter, Maximilian Lam, Mark Mazumder, and Vijay Janapa Reddi. the People's Speech: A Large-Scale Diverse English Speech Recognition Dataset for Commercial Usage. 2021. 1, 3

[28] Heting Gao, Hang Shao, Xiong Wang, Chaofan Qiu, Yunhang Shen, Siqi Cai, Yuchen Shi, Zihan Xu, Zuwei Long, Yike Zhang, Shaoqi Dong, Chaoyou Fu, Ke Li, Long Ma, and Xing Sun. LUCY: Linguistic Understanding and Control Yielding Early Stage of Her. *arXiv:2501.16327*, 2025. 3, 4, 8, 9, 1, 5

[29] Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. Paraformer: Fast and Accurate Parallel Transformer for Non-Autoregressive End-To-End Speech Recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2022. 8

[30] Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. Emilia: An Extensive, Multilingual, and Diverse Speech Dataset for Large-Scale Speech Generation. *arXiv:2501.15907*, 2024. 1, 3

[31] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 6

[32] Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, Yi Ren, Yuexian Zou, Zhou Zhao, and Shinji Watanabe. AudioGPT: Understanding and Generating Speech, Music, Sound, and Talking Head. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 1, 3

[33] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017. 7

[34] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint CTC-Attention Based End-To-End Speech Recognition Using Multi-Task Learning. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 2

[35] Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. LongForm: Optimizing Instruction Tuning for Long Text Generation with Corpus Extraction. 2023. 1, 3

[36] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. 2023. 1, 3

[37] Tianpeng Li, Jun Liu, Tao Zhang, Yuanbo Fang, Da Pan, Mingrui Wang, Zheng Liang, Zehuan Li, Mingan Lin, Guosheng Dong, Jianhua Xu, Haoze Sun, Zenan Zhou, and Weipeng Chen. Baichuan-Audio: A Unified Framework for End-To-End Speech Interaction. *arXiv:2502.17239*, 2025. 9, 4, 5

[38] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty. *International Conference on Machine Learning (ICML)*, 2024. 5

[39] Linhan Ma, Dake Guo, Kun Song, Yuepeng Jiang, Shuai Wang, Liumeng Xue, Weiming Xu, Huan Zhao, Binbin Zhang, and Lei Xie. WenetSpeech4TTS: A 12,800-Hour Mandarin TTS Corpus for Large Speech Generation Model Benchmark. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2024. 1, 3

[40] Ltd. Magic Data Technology Co. MAGICDATa Mandarin Chinese Read Speech Corpus, 2019. 1, 3

[41] Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-Math: Unlocking the Potential of SLMs in Grade School Math. 2024. 1, 3

[42] Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, R. J. Skerry-Ryan, and Michelle Tadmor Ramanovich. Spoken Question Answering and Speech Continuation Using Spectrogram-Powered Llm. In *The International Conference on Learning Representations (ICLR)*, 2024. 7

[43] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR Corpus Based on Public Domain Audio Books. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015. 8, 1, 3, 4

[44] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. MLS: A Large-Scale Multilingual Dataset for Speech Research. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2020. 1, 3

[45] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. 2024. 7, 4

[46] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition Via Large-Scale Weak Supervision. *International Conference on Machine Learning (PMLR)*, 2022. 7, 8

[47] Yunhang Shen, Chaoyou Fu, Shaoqi Dong, Xiong Wang, Peixian Chen, Mengdan Zhang, Haoyu Cao, Ke Li, Xiawu Zheng, Yan Zhang, Yiyi Zhou, Rongrong Ji, and Xing Sun. Long-VITA: Scaling Large Multi-Modal Models to 1 Million Tokens with Leading Short-Context Accuracy. *arXiv:2502.05177*, 2025. 6, 1

[48] Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. AISHELL-3: A Multi-Speaker Mandarin TTS Corpus and the Baselines. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2020. 3

[49] Step-audio Team. Step-Audio: Unified Understanding and Generation in Intelligent Speech Interaction. *arXiv:2502.11946*, 2024. 3, 9, 4, 5

[50] Teknium. OpenHermes 2.5: An Open Dataset of Synthetic Data for Generalist LLM Assistants, 2023. 1, 3

[51] Liu Tiedong. Goat, 2023. 1, 3

[52] Atlas Unified. Atlas math sets, 2023. 1, 3

[53] Changhan Wang, Morgane Rivière, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation. In *International Joint Conference on Natural Language Processing (IJCNLP)*, 2021. 1, 3

[54] Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Yuanjun Xiong, Wei Xia, and Mthreads Ai. a Full-Duplex Speech Dialogue Scheme Based on Large Language Models. 1, 3

[55] Wenbin Wang, Yang Song, and Sanjay Jha. GLOBE: A High-Quality English Corpus with Global Accents for Zero-Shot Speaker Adaptive Text-To-Speech. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2024. 1, 3

[56] Xiong Wang, Yangze Li, Chaoyou Fu, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-Omni: A Smart and Low Latency Speech-To-Speech Dialogue Model with Frozen LLM. *arXiv:2411.00774*, 2024. 1, 3, 4, 9, 5

[57] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-Of-The-Art Natural Language Processing. 2019. 9

[58] Zhifei Xie and Changqiao Wu. Mini-Omni: Language Models Can Hear, Talk While Thinking in Streaming. *arXiv:2408.16725*, 2024. 1, 3, 4

[59] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni Technical Report. *arXiv:2503.20215*, 2025. 9, 4, 5

[60] Jianxin Yang. LongQLoRA: Efficient and Effective Method to Extend Context Length of Large Language Models. 2023. 1, 3

[61] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 5

[62] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap Your Own Mathematical Questions for Large Language Models. *The International Conference on Learning Representations (ICLR)*, 2024. 1, 3

[63] Yijiong Yu. "Paraphrasing the Original Text" Makes High Accuracy Long-Context QA. 2023. 1, 3

[64] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building Math Generalist Models Through Hybrid Instruction Tuning. *The International Conference on Learning Representations (ICLR)*, 2024. 1, 3

[65] 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. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2019. 8, 1, 3

[66] Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. GLM-4-Voice: Towards Intelligent and Human-like End-To-End Spoken Chatbot. *arXiv:2412.02612*, 2024. 1, 3, 4, 7, 8, 5

[67] Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, Di Wu, and Zhendong Peng. WenetSpeech: A 10000+ Hours Multi-Domain Mandarin Corpus for Speech Recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022. 8, 1, 3, 5

[68] Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. LongCite: Enabling LLMs to Generate Fine-Grained Citations in Long-Context QA. 2024. 1, 3

[69] Pan Zhang, Xiaoyi Dong, Yuhang Cao, Yuhang Zang, Rui Qian, Xilin Wei, Lin Chen, Yifei Li, Junbo Niu, Shuangrui Ding, Qipeng Guo, Haodong Duan, Xin Chen, Han Lv, Zheng Nie, Min Zhang, Bin Wang, Wenwei Zhang, Xinyue Zhang, Jiaye Ge, Wei Li, Jingwen Li, Zhongying Tu, Conghui He, Xingcheng Zhang, Kai Chen, Yu Qiao, Dahua Lin, and Jiaqi Wang. InternLM-XComposer2.5-OmniLive: A Comprehensive Multimodal System for Long-Term Streaming Video and Audio Interactions. 2024. 1, 3

[70] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less Is More for Alignment. 2023. 1, 3

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. Our main contributions are also detailed in Sec.1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, please see Sec.C for limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide detailed assumptions and proofs in Sec.1 and Sec.3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We use publicly-accessable datasets. We upload the codes and instructions to recover the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use publicly-accessable datasets. We upload the codes and instructions to recover the results. Once the blind review period is finished, we'll open-source all codes, instructions, and model checkpoints.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all the data used in the experiments as well as the detailed experimental settings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, all our experimental results and latency measurements are averaged over multiple runs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the computational power of the GPUs used in our experiments in Sec.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This work involves training and evaluation on academic, publicly available datasets. It is not related to any private or personal data, and there are no explicit negative social impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We do not foresee any high risk for misuse of this work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we credited them in appropriate ways.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: No, our work does not describe the usage of LLMs as an important, original, or non-standard component of the core methods in this research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

VITA-Audio: Fast Interleaved Cross-Modal Token Generation for Efficient Large Speech-Language Model

Supplementary Material

A Training Data

ASR Data We aggregated approximately $100,000$ hours of open-source Automatic Speech Recognition (ASR) data, including WenetSpeech [67], Librispeech [43], Multilingual LibriSpeech [44], Common Voice 17 [43], MMCRSC [40], GigaSpeech [8], People's Speech [27], VoxPopuli [53], and the AISHELL series (AISHELL-1 [6] to AISHELL-4 [26]).

TTS Data Concurrently, we integrate approximately $100,000$ hours of open-source Text-to-Speech (TTS) data, primarily consisting of the Wenetspeech4TTS [39], LibriTTS [65], GLOBE [55], and Emilia [30] datasets.

Speech QA Data For speech question-answering (Speech-QA), we utilize VoiceAssistant-400K [58] and AudioQA-1.0M [28], totaling 1.4 million speech QA data, to enhance the model's speech-to-speech dialogue capabilities.

Text-Only Data The pure text data is collected from OpenHermes-2.5 [50] LIMA [70], databricks-dolly-15k [16], MetaMathQA [62], MathInstruct [64], Orca-Math [41], atlas-math-sets [52], goat [51], and camel-ai-math [36]. Given that discrete audio token sequences exhibit significantly longer lengths compared to their textual counterparts, we incorporate several specialized long-context text datasets following the Long-VITA [47] to enhance contextual modeling capabilities. These include Long-Instruction [63], LongForm [35], LongAlign-10k [3], LongCite-45k [68], LongWriter-6k [4], LongQLoRA [60], LongAlpaca [11], and LongData-Collections [14].

B Training Pipline

Stage 1: Audio-Text Alignment. Building upon pretrained language models, the goal of this stage is to extend the audio modeling capabilities to the LLM through large-scale speech pretraining. We freeze the audio encoder and audio decoder, and train the LLM using ASR, TTS, and Text-only data. During this stage, the output of the LLM can be either pure text tokens or audio tokens.

Stage 2: Single MCTP Module Training. After Stage 1, the model learns both text and audio distributions. The objective of Stage 2 is to train the initial MCTP module to predict one subsequent token based on the output tokens and hidden states from the LLM. This stage employs the same dataset configuration as Stage 1. We initialize the MCTP module using parameters from the final layer of the LLM, with the gradient detached from the LLM.

Stage 3: Multiple MCTP Modules Training. The objective of this stage is to extend the single MCTP module to multiple MCTP modules. Specifically, each MCTP module predicts the token at its corresponding position given the output tokens and hidden states of the previous MCTP module. All subsequent MCTP modules are initialized using the weights of the MTCP module from Stage 2. This stage also incorporates gradient detachment to optimize the model training process.

Stage 4: Supervised Fine-tuning. After the previous three training stages, VITA-Audio has acquired the ability to efficiently and accurately map text to audio. To enable speech-to-speech dialogue capability, we then conduct supervised fine-tuning using speech QA datasets while maintaining a small amount of TTS, ASR, and text-only data to ensure training stability. To optimize training effectiveness, different learning rates are used for the MCTP module and the main LLM. For the speech-to-speech data, we employ an interleaved output format. This design enforces the model to initiate audio token generation during the first forward pass, enabling synchronized decoding of the audio tokens rather than waiting until all text tokens have been generated.

C Limitations

While our approach enables efficient generation of audio tokens, the overall end-to-end latency remains above the theoretical lower bound, primarily due to the constrained generation speed of the audio decoder. Further improving the response speed of the audio decoder in end-to-end speech models is a worthwhile direction for future exploration.

D Data Format

Speech QA Interleaved Data Format

```
{
    "messages": [
    {
        "role": "user",
        "content": "<|begin_of_audio|> audio_sequence_1 <|end_of_audio|>"
    },
    {
        "role": "assistant",
        "content": "text_sequence_1 <|begin_of_audio|>
        audio_sequence_2 <|end_of_audio|> text_sequence_2
        <|begin_of_audio|> audio_sequence_3 <|end_of_audio|>"
    },
    {
        "role": "user",
        "content": "<|begin_of_audio|> audio_sequence_4 <|end_of_audio|>"
    },
    {
        "role": "assistant",
        "content": "text_sequence_3 <|begin_of_audio|>
        audio_sequence_5 <|end_of_audio|> text_sequence_4
        <|begin_of_audio|> audio_sequence_6 <|end_of_audio|>"
    }
    ]
}
```

Prompt for TTS task.

```
{
    "messages": [
    {
        "role": "user",
        "content": "Convert the text to speech.\ntext_sequence"
    }
    ]
}
```

Prompt for ASR task.

```
{
    "messages": [
    {
        "role": "user",
        "content": "Convert the speech to text.\n
        <|begin_of_audio|> audio_sequence <|end_of_audio|>"
    }
    ]
}
```

E Figures and Tables

Table E1: Summary of datasets used in VITA-Audio for different stages.

Task	Name		Total Number	Sampling Ratio			
				Stage 1	Stage 2	Stage 3	Stage 4
ASR	WenetSpeech	[67]	10,000H	1.0	1.0	1.0	0.05
	Librispeech	[43]	1,000H	1.0	1.0	1.0	0.05
	Multilingual LibriSpeech	[44]	71,506H	1.0	1.0	1.0	0.05
	Common Voice 17	[43]	2.849H	1.0	1.0	1.0	0.05
	MMCRSC	[40]	755H	1.0	1.0	1.0	0.05
	GigaSpeech	[8]	10,000H	1.0	1.0	1.0	0.05
	People's Speech	[27]	1,000H	1.0	1.0	1.0	0.05
	VoxPopuli	[53]	543H	1.0	1.0	1.0	0.05
	AISHELL-1	[6]	170H	1.0	1.0	1.0	0.05
	AISHELL-2	[20]	1,000H	1.0	1.0	1.0	0.05
	AISHELL-3	[48]	85H	1.0	1.0	1.0	0.05
	AISHELL-4	[26]	120H	1.0	1.0	1.0	0.05
TTS	Wenetspeech4TTS	[39]	12,800H	1.0	1.0	1.0	0.05
	LibriTTS	[65]	585H	1.0	1.0	1.0	0.05
	GLOBE	[55]	535H	1.0	1.0	1.0	0.05
	Emilia	[30]	96,700H	1.0	1.0	1.0	0.05
Speech QA	VoiceAssistant-400K	[58]	400K	0.0	0.0	0.0	2.0
	AudioQA-1.0M	[28]	1M	0.0	0.0	0.0	2.0
Text QA	OpenHermes-2.5	[50]	1M	1.0	1.0	1.0	0.05
	LIMA	[70]	1K	1.0	1.0	1.0	0.05
	databricks-dolly-15k	[16]	15K	1.0	1.0	1.0	0.05
	MetaMathQA	[62]	395K	1.0	1.0	1.0	0.05
	MathInstruct	[64]	262K	1.0	1.0	1.0	0.05
	Orca-Math	[41]	200K	1.0	1.0	1.0	0.05
	atlas-math-sets	[52]	17.8M	1.0	1.0	1.0	0.05
	goat	[51]	1.7M	1.0	1.0	1.0	0.05
	camel-ai-math	[36]	50K	1.0	1.0	1.0	0.05
Long Text QA	Long-Instruction	[63]	16K	1.0	1.0	1.0	0.05
	LongForm	[35]	23K	1.0	1.0	1.0	0.05
	LongAlign-10k	[3]	10K	1.0	1.0	1.0	0.05
	LongCite-45k	[68]	45K	1.0	1.0	1.0	0.05
	LongWriter-6k	[4]	6K	1.0	1.0	1.0	0.05
	LongQLoRA	[60]	39K	1.0	1.0	1.0	0.05
	LongAlpaca	[11]	12K	1.0	1.0	1.0	0.05
	Long-Data-Collections	[14]	98K	1.0	1.0	1.0	0.05

Table E2: Comparison of model structures between VITA-Audio and VITA-Audio-Plus.

Name	Base LLM	Audio Encoder	Audio Adapter	Audio Decoder
VITA-Audio	Qwen2.5-7B [45]	GLM-4-Voice-Tokenizer [66]	–	GLM-4-Voice-Decoder [66]
VITA-Audio-Plus	Qwen2.5-7B [45]	SenseVoiceSmall [1]	MLP	GLM-4-Voice-Decoder [66]

Table E3: Results on Automatic Speech Recognition (ASR) Benchmarks. "Sx" denotes the x-th training stage. Compared to other methods, **VITA-Audio is trained with open-source data only**.

Datasets	Model		WER (%) ↓
LibriSpeech [43] *test-clean* | *test-other*	Qwen2-Audio-base	[12]	**1.74** | **4.04**
	Baichuan-Audio-base	[37]	3.02 | 6.04
	Freeze-Omni	[56]	3.82 | 9.79
	VITA-1.5 (S3)	[25]	3.40 | 7.50
	LUCY (S1)	[28]	3.36 | 8.05
	Step-Audio-chat	[49]	3.19 | 10.67
	Qwen2.5-Omni	[59]	2.37 | 4.21
	VITA-Audio-Turbo		6.29 | 12.86
	VITA-Audio-Vanilla		2.98 | 8.07
	VITA-Audio-Plus-Boost		3.13 | 7.07
	VITA-Audio-Plus-Vanilla (S1)		1.91 | 4.29
	VITA-Audio-Plus-Vanilla		2.00 | 4.60
Fleurs [15] *zh* | *en*	Qwen2-Audio-base	[12]	3.63 | 5.20
	Baichuan-Audio-base	[37]	4.15 | 8.07
	Step-Audio-chat	[49]	4.26 | 8.56
	Qwen2.5-Omni	[59]	**2.92** | **4.17**
	VITA-Audio-Plus-Vanilla		3.69 | 4.54

Table E4: Results on Automatic Speech Recognition (ASR) Benchmarks. "Sx" denotes the x-th training stage. Compared to other methods, **VITA-Audio is trained with open-source data only**.

Datasets	Model		WER (%) ↓
AISHELL-1 [6]	Qwen2-Audio-base	[12]	1.52
	Baichuan-Audio-base	[37]	1.93
	Freeze-Omni	[56]	2.48
	LUCY (S1)	[28]	2.40
	Step-Audio-chat	[49]	2.14
	Qwen2.5-Omni	[59]	**1.13**
	VITA-Audio-Turbo		7.70
	VITA-Audio-Vanilla		4.46
	VITA-Audio-Plus-Boost		4.72
	VITA-Audio-Plus-Vanilla (S1)		1.51
	VITA-Audio-Plus-Vanilla		1.94
AISHELL-2 ios [20]	Qwen2-Audio-base	[12]	3.08
	Baichuan-Audio-base	[37]	3.87
	Step-Audio-chat	[49]	3.89
	Qwen2.5-Omni	[59]	**2.56**
	VITA-Audio-Plus-Vanilla		3.29
WenetSpeech [67] *test-meeting* | *test-net*	Qwen2-Audio-base	[12]	8.40 | 7.64
	Baichuan-Audio-base	[37]	13.28 | 10.13
	Freeze-Omni	[56]	13.46 | 11.80
	VITA-1.5 (S3)	[25]	10.0 | 8.40
	LUCY (S1)	[28]	10.42 | 8.78
	Step-Audio-chat	[49]	10.83 | 9.47
	Qwen2.5-Omni	[59]	7.71 | **6.04**
	VITA-Audio-Turbo		23.97 | 18.66
	VITA-Audio-Vanilla		17.34 | 13.45
	VITA-Audio-Plus-Boost		9.38 | 8.97
	VITA-Audio-Plus-Vanilla (S1)		**6.68** | 6.59
	VITA-Audio-Plus-Vanilla		7.12 | 6.90

Table E5: More results on Spoken Question Answering (SQA) benchmarks. "Sx" denotes the x-th training stage of speech models.

Model		#Params	Llama Question		TriviaQA		Web Question		Mean	
			S → T	S → S	S → T	S → S	S → T	S → S	S → T	S → S
Proprietary Models										
MinMo	[9]	7B	78.9	64.1	48.3	37.5	55.0	39.9	60.7	47.2
Open-source Models										
Moshi	[19]	7B	62.3	21.0	22.8	7.3	26.6	9.2	37.2	12.5
GLM-4-Voice	[66]	9B	64.7	50.7	39.1	26.5	**55.0**	39.9	45.3	31.0
LUCY (S2)	[28]	7B	59.6	51.0	23.2	18.2	26.6	18.2	36.5	29.1
MiniCPM-o2.6	[61]	7B	-	61.0	-	40.0	-	40.2	-	47.0
Llama-Omni	[23]	7B	-	45.3	-	22.9	-	10.7	-	26.3
VITA-Audio-Boost		7B	68.7	60.3	30.5	29.3	32.9	30.4	44.0	40.0
VITA-Audio-Vanilla		7B	71.3	66.3	31.9	30.1	33.5	31.4	45.6	42.6
VITA-Audio-Plus-Boost		7B	**76.3**	64.6	43.6	39.5	44.2	40.0	54.7	48.0
VITA-Audio-Plus-Vanilla		7B	75.6	**68.0**	**45.9**	**42.7**	45.0	**41.7**	**55.5**	**50.8**

Table E6: We have tested VITA-Audio after aligning with ASR and TTS tasks on several text modality benchmarks. After the ASR and TTS alignment, the model's original text understanding capabilities were indeed affected. However, this might be due to the lack of large-scale, high-quality text data used during training. Interestingly, we observed a slight improvement in performance on GSM8K, which may be due to the fact that our training dataset included a significant amount of math-related data.

Model	MMLU[31]	GSM8K[13]
Qwen-7B-Instruct	74.22	80.06
VITA-Audio-Plus-Vanilla	66.92	80.14

Table E7: Generation time (ms) of the first audio segment under different inference modes in streaming inference. To enable more real-time speech generation, we progressively increase the number of steps in the flow matching model during streaming inference. The table shows the decoding time when the sampling step of the flow matching model is set to 1.

Inference Mode	Audio Encoder	First Audio Token Chunk	Audio Decoder	Sum
VITA-Audio-Boost	39	53	151	243
VITA-Audio-Vanilla	39	236	151	426