Does Your Voice Assistant Remember? Analyzing Conversational Context Recall and Utilization in Voice Interaction Models

Heeseung Kim[1*] **Che Hyun Lee**[1*] **Sangkwon Park**[1] **Jiheum Yeom**[1]
Nohil Park[1] **Sangwon Yu**[1] **Sungroh Yoon**[1,2†]

[1]Department of Electrical and Computer Engineering, Seoul National University
[2]AIIS, ASRI, INMC, ISRC, and IPAI, Seoul National University

{gmltmd789, saga1214, tkdrnjs0621, quilava1234, pnoil2588, dbtkddnjs96, sryoon}@snu.ac.kr

Abstract

Recent advancements in multi-turn voice interaction models have improved user-model communication. However, while closed-source models effectively retain and recall past utterances, whether open-source models share this ability remains unexplored. To fill this gap, we systematically evaluate how well open-source interaction models utilize past utterances using ContextDialog, a benchmark we proposed for this purpose. Our findings show that speech-based models have more difficulty than text-based ones, especially when recalling information conveyed in speech, and even with retrieval-augmented generation, models still struggle with questions about past utterances. These insights highlight key limitations in open-source models and suggest ways to improve memory retention and retrieval robustness.[1]

1 Introduction

Voice assistants such as Apple Siri and Amazon Alexa have become an irreplaceable element of daily life, enabling natural and efficient speech-based interactions. In early systems, a cascaded pipeline is employed where speech is first transcribed using automatic speech recognition (ASR), then processed as text, and finally converted back to speech via text-to-speech (TTS) (Lin et al., 2024b). With the advent of large language models (LLMs) (Brown et al., 2020; Llama Team, 2024), however, the research community has shifted greatly towards end-to-end approaches. These models integrate ASR, text processing, and TTS into a unified multimodal framework (Zhang et al., 2023), which not only reduces latency (Xie and Wu, 2024a) but also better preserves the richness of vocal cues (Kim et al., 2024). In line with this trend, GPT-4o (OpenAI, 2024a) has demonstrated impressive capabilities by processing visual, speech, and text data in an end-to-end manner, where various voice interaction models, datasets, and benchmarks have rapidly emerged alongside. (Cheng et al., 2025a,b; Fang et al., 2025; Xie and Wu, 2024b).

Despite these advances, most current models excel only in single-turn interactions. In practical applications, however, users engage in multi-turn dialogs where a one-off response is insufficient. Specifically, models must continuously retain and leverage contextual information from previous turns. For example, Gemini 2.0 (Google DeepMind, 2024) demonstrates the ability to remember preceding details—for instance when a user provides an apartment door code during interaction and inquires about it later—thereby showcasing robust context-maintenance. Notably, other closed-source solutions, such as OpenAI's Advanced Voice Mode (OpenAI, 2024), have also showcased similar capabilities by referencing past interactions.

In parallel, the open-source community has also intensified its efforts to develop voice interaction models that support multi-round communications (Défossez et al., 2024; Yao et al., 2024; Zeng et al., 2024). Typically, these models take speech as input and generate both text and speech outputs, rather than producing spoken responses alone, to leverage the strengths of pre-trained LLMs and ensure coherent, multi-turn responses. However, it remains unclear whether current open-source systems can effectively retain and utilize long-range interaction histories. Furthermore, there are no benchmarks that explicitly require leveraging dialog history to generate responses.

In this work, we systematically investigate the ability of open-source voice interaction models to maintain and utilize conversational context through two key experiments. We evaluate (1) whether models can recall and generate spoken responses based on previous dialog and (2) their robustness in incorporating externally retrieved utterances. To support

[*]Equal Contribution
[†]Corresponding Author
[1]ContextDialog: https://huggingface.co/datasets/ContextDialog/ContextDialog

Findings of the Association for Computational Linguistics: ACL 2025, pages 8984–9014
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

this evaluation, we introduce ContextDialog—a speech-to-speech benchmark that focus on assessing recall via spoken question-answer (QA) pairs derived from existing spoken dialogs, prompting one speaker to reference earlier information.

Our findings reveal that open-source models struggle with past speech in two key aspects. **Performance gap with text-based systems** – Speech models generally perform worse than their text-based counterparts, and **Modality-based recall gap** – Within speech models, recalling speech-based information is less accurate than retrieving text, likely due to weaker speech processing capabilities. Additionally, our investigation of retrieval-augmented generation (RAG) shows that it fails to compensate for the model's inability to recall past information. We identify a major challenge: **Sensitivity to retrieval errors** – Models are highly susceptible to retrieval mistakes, leading to unchanged or even degraded performance. Through these findings, we highlight the challenges models face in processing past conversational context and their sensitivity to noise in retrieved information, drawing attention to a fundamental, yet often overlooked, capability within the open-source community. Our contributions are as follows:

- We introduce ContextDialog, a benchmark designed to evaluate the models' ability to utilize dialog history in multi-turn conversations.[2]
- We show that most open-source models struggle with recalling past dialogs and fail to effectively incorporate retrieved information, even when augmented with external retriever.
- Through extensive evaluation and analysis, we uncover overlooked limitations in current models that restrict their applicability and propose directions for future improvements.

2 Related Works

Voice Interaction Models Early voice interaction models follow a cascaded approach (Lin et al., 2024b), transcribing speech into text, processing the transcription, and then synthesizing the output speech. Recently, end-to-end pipelines have emerged, performing these steps within a single model (Zhang et al., 2023). Although some models generate spoken responses without relying on text (Nguyen et al., 2023), the inherent length and data scarcity of speech hinder semantic modeling (Défossez et al., 2024). Recent approaches integrate text generation within speech modeling to mitigate such problem, leveraging pre-trained LLMs by incorporating text as an intermediate representation (Kim et al., 2024; Zhang et al., 2023), generating it alongside speech (Fang et al., 2025), or interleaving it with speech tokens (Zeng et al., 2024).

Many prior works focus on single-turn voice interaction (Fang et al., 2025; Kim et al., 2024; Xie and Wu, 2024a,b; Zeng et al., 2025; Zhang et al., 2023; Zhao et al., 2024). As a natural extension, multi-turn voice interaction models have also emerged (Chen et al., 2024c, 2025; Défossez et al., 2024; Fu et al., 2025; Li et al., 2025; Mai and Carson-Berndsen, 2025; Mitsui et al., 2024; Park et al., 2024; Veluri et al., 2024; Wang et al., 2024c; Yao et al., 2024; Zeng et al., 2024; Zhang et al., 2024, 2025; Zhong et al., 2024).

Among these, only SLAM-Omni (Chen et al., 2024c) and Lyra (Zhong et al., 2024) explicitly discuss long-context modeling. SLAM-Omni seeks to improve multi-turn modeling by storing dialog history in transcribed text form, which is then prepended as a prefix during inference. Lyra, on the other hand, explores and proposes various techniques to handle long audio histories and extends context windows directly within its model architecture. The remaining works either mention the use of multi-turn data for training or imply it through demonstrations or official implementations. However, they generally do not propose explicit methods or evaluations specifically targeting long-context recall capability. Consequently, whether these models can effectively handle past conversational history in real-world multi-turn dialog scenarios remains largely unexplored.

For voice assistants to function effectively, it is essential to assess their ability to retain and utilize prior utterances to generate contextually appropriate responses. To this end, we select SLAM-Omni (included in Appendix A.1.6 due to its smaller model size) and Lyra as our primary long-context baselines. Additionally, we include Freeze-Omni, GLM-4-Voice, MiniCPM-o, and Moshi (also in Appendix A.1.6) as strong open-source candidates with multi-turn capabilities available.

Benchmarks Numerous datasets and benchmarks for audio foundation models have emerged (Sakshi et al., 2025; Wang et al., 2025; Yang et al., 2024c), particularly for voice interaction models (Chen et al., 2024b; Fang et al., 2025; Park

[2]Project Page: https://contextdialog.github.io/

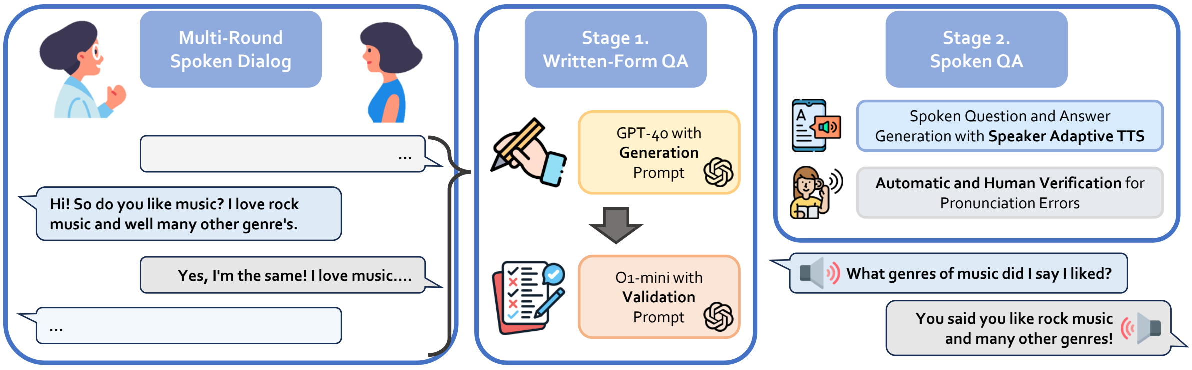


Figure 1: Overview of the ContextDialog generation process. Past-recall QA pairs are first generated and validated (Section 3.1), then converted to speech via adaptive TTS and verified both automatically and manually (Section 3.2).

et al., 2024; Xie and Wu, 2024a). For example, in task-oriented spoken dialogs, benchmarks assess a model's ability to recognize entities and dialog states from past utterances (Henderson et al., 2014; Si et al., 2023; Spithourakis et al., 2022), while in open-domain dialogs, they focus on modeling and evaluating response coherence (Busso et al., 2008; Cieri et al., 2004; Cheng et al., 2025a; Park et al., 2024). Beyond semantic relevance, recent works propose benchmarks targeting non-verbal components crucial for voice interaction models, such as gender, emotion, and background noise (Ao et al., 2024; Chen et al., 2024d; Cheng et al., 2025b). Unlike existing benchmarks that evaluate multi-turn semantics without ensuring past information is necessary for responses, ContextDialog explicitly requires models to retrieve and utilize relevant past utterances, enabling a systematic assessment of recall ability.

Retrieval in Voice Interaction Model With advancements in RAG techniques in natural language processing (NLP) (Izacard et al., 2023; Lewis et al., 2020), efforts to integrate RAG into spoken dialog models have emerged (Lin et al., 2024a; Min et al., 2025; Wang et al., 2024a). Prior works have primarily focused on task-oriented dialog for entity extraction (Wang et al., 2024a) or spoken question answering (Lin et al., 2024a; Min et al., 2025), retrieving information from long speech documents (Lee et al., 2018). In contrast, we focus on multi-turn voice interactions, examining whether relevant data retrieved via an external module can be effectively utilized in the generation, specifically tailored for recent open-source interaction models.

3 ContextDialog

We propose ContextDialog, a comprehensive benchmark designed to evaluate a voice interaction model's ability to engage in, retain, and leverage relevant information throughout multi-turn conversations, reflecting real-world scenarios where people often forget and revisit past exchanges. ContextDialog is constructed using MultiDialog (Park et al., 2024), a spoken dialog corpus featuring conversations between two speakers, comprising approximately 340 hours of data with at least 10 turns per conversation from 12 speakers. We use the `test_freq` and `test_rare` splits from MultiDialog, consisting of 450 and 381 spoken dialogs, respectively. Some data are filtered during generation and validation, with the final statistics of ContextDialog shown in Table 1 and the data generation pipeline illustrated in Figure 1.

	Statistics	`test_freq`	`test_rare`
Dialog History	# dialogs	363	290
	max turn	16	24
	min turn	10	10
	avg turn	10.57	10.61
Question / Answer	# QA pairs	1,452	1,160
	max dur(s)	13.19 / 24.80	19.23 / 22.11
	min dur(s)	2.60 / 1.11	2.14 / 1.30
	avg dur(s)	5.97 / 6.78	5.90 / 6.59

Table 1: Statistics of ContextDialog for Dialog History and generated QA on `test_freq` and `test_rare` splits. The numbers on the left and right are related to the question and answer, respectively. The term dur refers to the duration of the generated question and answer.

3.1 Text Question-Answer Generation

We first construct a dataset of context-recall QA pairs using *gpt-4o*. Given the transcripts of MultiDialog, the model is prompted to generate questions and answers based solely on information that appeared only once in the conversation. To ensure diversity and broad applicability, we generate questions based on both user and system utterances, selecting information from either the first or second

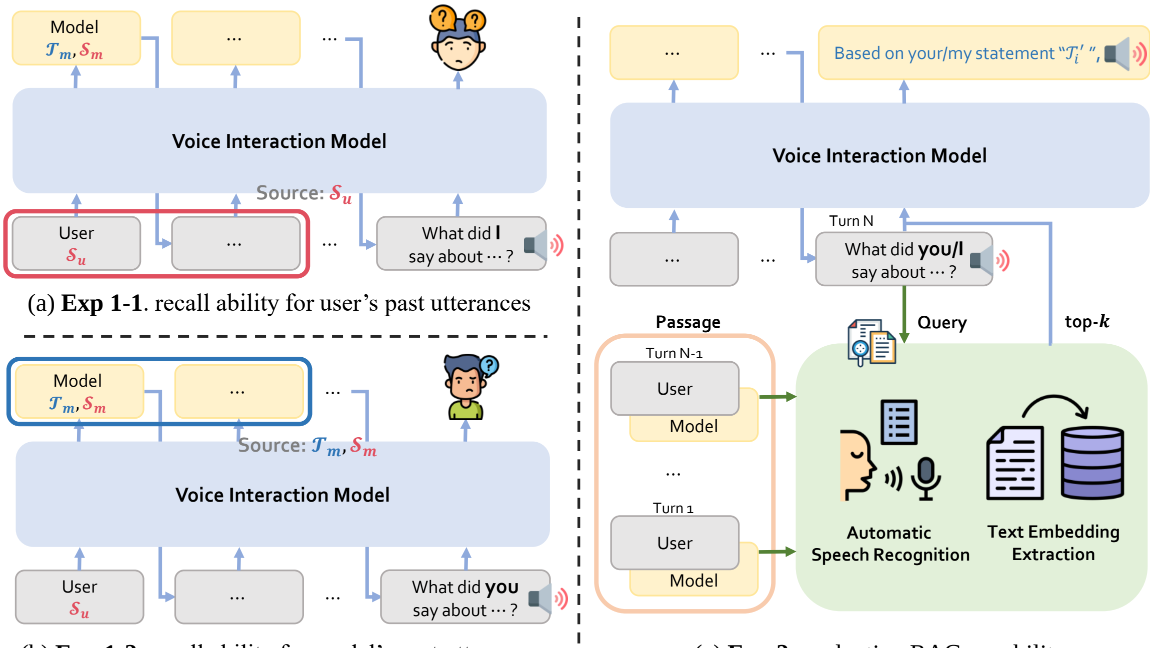


Figure 2: Overview of our analyses. In Section 4.1, we evaluate model recall by analyzing responses to questions about (a) past user and (b) past model utterances. In Section 4.2, we examine whether (c) augmenting spoken response generation with separately retrieved utterances improves responses to questions about past utterances.

half of the conversation. This results in four QA pairs per spoken dialog. Additionally, the model is requested to output the *supporting utterance*—the utterance in the conversation that serves as the clue for the answer—for each pair to enhance both data quality and usability. For a more realistic setting, the questions are designed to require detailed answers rather than simple Yes/No responses.

After generating the QA pairs, we validate each question, answer, and supporting utterance using *o1-mini* (OpenAI, 2024b). A validation prompt assesses their appropriateness within the dialog context through three rounds of evaluation: (1) dialog context up to just before the supporting utterance, (2) up to and including the supporting utterance, and (3) the entire conversation. The first step is to check whether the answer can be inferred without the supporting utterance, requiring a NO response, while the second and third ensure consistency across different context levels, requiring YES responses. Failed QA pairs are filtered out, and the validated pairs are used to construct the spoken QA dataset. The prompts used are in Appendix A.3.

3.2 Spoken Question-Answer Generation

To ensure that the user and the model continue naturally in the given spoken dialog, the voice of the spoken QA pair must seamlessly match that of the original conversation. To achieve this, we use Fish Speech (Liao et al., 2024), a speaker-adaptive TTS model that generates speech in the target speaker's timbre using a reference speech. For each QA pair, we select the longest speech segment from the original dialog for each speaker as the reference to maximize speaker similarity. To ensure accurate pronunciation, each spoken QA pair is generated five times, and the sample with the lowest word error rate (WER)—measured using a separate ASR model (Radford et al., 2023)—is selected. If the selected sample has a nonzero WER, it is manually reviewed, and mispronounced samples are filtered out. This process ensures that ContextDialog maintains both speaker identity and clear pronunciation in the final spoken QA pairs.

4 Experiments

In this section, we present the results of two experiments and analyses using ContextDialog. In Section 4.1, we demonstrate that open-source voice interaction models struggle to recall past information on their own, particularly user-specific information that exists solely in spoken form. Then, in Section 4.2, we show that even when leveraging a more advanced dedicated text retriever, models fail to respond robustly given the retrieved information, yielding limited improvements in spoken response generation. These two analyses highlight a critical yet often overlooked aspect of voice interaction models, their ability to remember past interactions,

Model	LLM FT	Modality	GPT Score			WER
			User	System	Overall	
GLM-4-Voice (GLM et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	1.94±0.07	2.76±0.08	2.35±0.05	8.36%
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	2.04±0.07	2.97±0.08	2.50±0.06	–
glm-4-9b-chat (Zeng et al., 2024)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.30±0.05	3.90±0.06	4.10±0.04	–
Lyra (Zhong et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	2.51±0.09	3.16±0.09	2.83±0.06	5.90%
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	2.67±0.09	3.38±0.09	3.03±0.07	–
Qwen2-VL-7B-Instruct (Wang et al., 2024b)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	3.80±0.08	3.88±0.08	3.84±0.06	–
Freeze-Omni (Wang et al., 2024c)	✗	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	1.73±0.06	2.28±0.07	2.00±0.05	12.36%
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	2.09±0.07	3.06±0.08	2.57±0.06	–
Qwen2-7B-Instruct (Yang et al., 2024a)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.26±0.06	3.81±0.07	4.03±0.05	–
MiniCPM-o (Yao et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	2.44±0.09	2.84±0.09	2.64±0.06	24.90%
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	3.22±0.09	3.93±0.08	3.58±0.06	–
Qwen2.5-7B-Instruct (Yang et al., 2024b)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.28±0.05	3.84±0.06	4.06±0.04	–

Table 2: Evaluation results for voice interaction models, including the instruct fine-tuned version of each model's backbone LLM. $\mathcal{S}$ and $\mathcal{T}$ represent speech and text, respectively. The bold and underlined modality indicates the data type used for evaluation. "Modality" indicates input → output data type. "LLM FT" shows whether the backbone LLM was fine-tuned during training. "User" and "System" represent scores for responses to past user and model utterances, respectively. "Overall" denotes the score across all responses. "WER" refers to the word error rate between the model's intermediate text response and the transcribed spoken response, highlighting degradation from speech synthesis. GPT Scores are reported with a 95% confidence interval.

which is essential for real-world deployment.

For our experiments, we select four open-source multi-turn voice interaction models: GLM-4-Voice (Zeng et al., 2024), MiniCPM-o 2.6 (Yao et al., 2024), Freeze-Omni (Wang et al., 2024c), and Lyra (Zhong et al., 2024). To support real-time generation and minimize latency, these models generate responses directly from the input speech without an intermediate speech-to-text conversion. To mitigate semantic degradation in speech-only generation, these models generate text responses alongside spoken responses: GLM-4-Voice employs an interleaved token generation approach, alternating between text and speech tokens (Figure 5(a)), while MiniCPM-o, Freeze-Omni, and Lyra generate text responses while simultaneously synthesizing speech using real-time generated text tokens and the LLM's hidden states (Figure 5(b)).

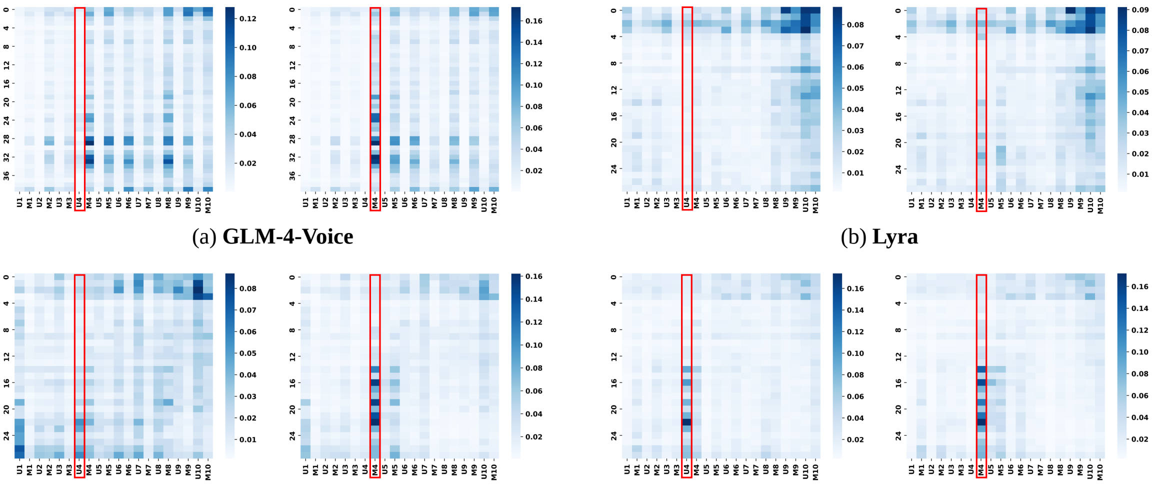
In all experiments, we evaluate each model's spoken response using the LLM-as-a-judge approach (Zheng et al., 2023), following previous works (Chen et al., 2024c; Zeng et al., 2025). This setting is denoted as $\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$, where $\mathcal{S}$ refers to speech data and $\mathcal{T}$ to text data. The bold and underlined modality symbol indicates the evaluation target in each configuration. We employ *gpt-4o-mini* for evaluation, referred to as the GPT Score in this paper, using a five-point scale where higher scores indicate better performance. We design prompts to assess recall by measuring how well the generated responses contain the ground truth information relevant to the given question, as detailed in Appendix A.4. Since *gpt-4o-mini* is tailored to text inputs, we first convert the spoken responses into text using *whisper-large-v3* (Radford et al., 2023).

Additionally, considering that each model also generates an intermediate text response corresponding to the spoken output, we also evaluate it ($\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$). By analyzing the evaluation results along with the word error rate (WER) between the text response and the transcribed spoken response, we can disentangle recall ability from speech synthesis capability, allowing us to identify cases where the model successfully recalls information but fails in speech synthesis.

We use the official implementations, hyperparameters, and checkpoints for all four models (Section A.2), running experiments on a single NVIDIA A40 GPU. Detailed model descriptions and additional analyses are provided in Appendix A.1.

4.1 Does Your Model Truly Recall Past Information?

Using ContextDialog, we examine whether these models can recall or remind users of past utterances, either from the user or the model itself. To assess



(a) **GLM-4-Voice** (b) **Lyra**

(c) **Freeze-Omni** (d) **MiniCPM-o**

Figure 3: Attention maps for ground truth answers given each model's past dialog and question. The x-axis of the figure indicates the order of utterances of each speaker ("U" for user, "M" for model), while the y-axis shows the index of attention layer. In each subfigure, the left side represents questions about past user utterances, and the right side represents questions about past model utterances. Red boxes indicate the positions of supporting utterances.

differences in question difficulty, we additionally evaluate the instruct fine-tuned versions of each model's backbone LLM (GLM et al., 2024; Yang et al., 2024a,b; Wang et al., 2024b), providing a basis for comparing the difficulty of questions based on user-spoken versus model-generated utterances.

The scores for each model on questions about past user utterances and the model's own responses, along with their 95% confidence intervals and averages, are presented in Table 2. In this table, we observe two key patterns. First, in multi-turn dialogs requiring past context, all voice interaction models (shaded in gray) show significant performance drop compared to text-based counterparts (unshaded), regardless of whether evaluation is on the intermediate text response ($\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$) or the transcribed response ($\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$). This degradation is particularly pronounced in Freeze-Omni, where the LLM is frozen during speech model training (LLM FT: ✗). These results indicate that expanding a pre-trained LLM to speech significantly weakens its ability to process long contexts.

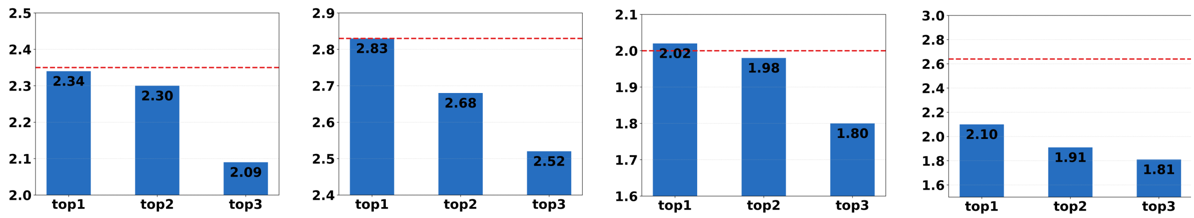
Secondly, unlike their text-based counterparts (unshaded), voice interaction models (shaded in gray) perform consistently better when recalling their own past utterances than the user's ($p < 0.01$). This stems from the generation mechanism of recent voice interaction models. Since speech-only output degrades semantic modeling, most modern models generate text alongside speech to leverage the backbone LLM's text capability. Consequently, when responding to questions about the model's past utterances, both text and speech are utilized (Figure 2(b)), whereas for user utterances, the model must rely solely on speech (Figure 2(a)).

We further analyze how models respond to questions about past user and model utterances by examining their attention maps during response generation as shown in Figure 3. The horizontal axis represents the turn index ("U" for user, "M" for model), and the vertical axis represents the attention layer index. We sum attention weights over all tokens in each utterance. As shown, models attend less to supporting utterances when answering questions about past user utterances than model utterances. This suggests that an inherent bias, where models allocate less attention to user-spoken content, contributes to the recall gap and highlights the need for improved modeling capabilities.

The findings in this section reveal modality-specific differences, both compared to text interaction models and within speech models. They underscore the need to improve voice interaction models by introducing training and generation methods to better utilize long-range conversational history. In the following section, we validate a practical approach to enhancing past information utilization with minimal computational cost when the model fails to recall relevant details. Specifically, we examine the effectiveness of retrieval-augmented generation (RAG) in voice interaction models.

4.2 Does Your Model Reliably Augment Retrieved Information into Generation?

Leveraging RAG methods from the NLP domain (Izacard et al., 2023; Lewis et al., 2020), we assess



(a) **GLM-4-Voice** (b) **Lyra** (c) **Freeze-Omni** (d) **MiniCPM-o**

Figure 4: The results of applying the RAG method to each model are shown. The y-axis values indicate GPT Scores on a 5-point scale, with higher scores representing better performance. The red dashed line indicates the results generated without RAG (Section 4.1). The evaluation is based on the transcribed spoken response $\mathcal{S} \to \mathcal{T}$, $\boldsymbol{\mathcal{S}}$.

whether voice interaction models can effectively utilize past utterances when retrieved by a dedicated module, as illustrated in Figure 2(c). Given our observation in Section 4.1 that models struggle more with speech than text, we transcribe past user and model utterances at the end of each speech segment using a separate ASR model. These transcriptions are stored with their corresponding text embeddings, extracted via a separate retriever.

Once stored, these transcriptions serve as passages from which relevant information can be retrieved when a user query arrives. Upon receiving input speech, we convert it into text using the same ASR model, extract its embedding, and retrieve the top-k most relevant past utterances by comparing cosine similarity. These retrieved utterances are then augmented into spoken response generation. We use *whisper-large-v3-turbo* (Radford et al., 2023) for the ASR model and *e5-large-v2* (Wang et al., 2022), a widely used retrieval model in NLP.

The retrieved texts are incorporated into the generation stage using the following format: `Based on your/my statement: “RETRIEVED TRANSCRIBED TEXT1”, your/my statement: “RETRIEVED TRANSCRIBED TEXT2” ....` The choice between `your` and `my` depends on the speaker of the retrieved utterance, ensuring clear integration into the prompt. The model then utilizes this prompt to generate spoken responses, as shown in Figure 2(c). Details on how each model incorporates this prompt into spoken response generation are in Appendix A.1.2, while experiments with various other prompts are discussed in Appendix A.1.8.

The experimental results on integrating RAG into voice interaction models are presented in Figure 4, where (a)–(d) correspond to the four evaluated models. The red dashed line indicates baseline performance when models generate responses based solely on intrinsic recall without RAG (Section 4.1). These results are measured using the ASR transcript of the spoken response ($\mathcal{S} \to \mathcal{T}$, $\boldsymbol{\mathcal{S}}$) for

Model	Prompt	GPT Score
GLM-4-Voice	✗	2.35 ± 0.05
	Supporting	$\mathbf{2.60 \pm 0.05}$
	Irrelevant	1.87 ± 0.05
Lyra	✗	2.83 ± 0.06
	Supporting	$\mathbf{3.44 \pm 0.05}$
	Irrelevant	1.96 ± 0.05
Freeze-Omni	✗	2.00 ± 0.05
	Supporting	$\mathbf{2.38 \pm 0.04}$
	Irrelevant	1.54 ± 0.04
MiniCPM-o	✗	$\mathbf{2.64 \pm 0.06}$
	Supporting	2.49 ± 0.06
	Irrelevant	1.63 ± 0.05

Table 3: GPT Score results when augmenting spoken response generation with either the ground-truth supporting utterance (“Supporting”) or an entirely unrelated utterance (“Irrelevant”) as prompts.

all QA pairs, and trends in the intermediate text response are similar, as detailed in Appendix A.1.3.

As shown in the figure, all models perform similarly or worse with RAG, showing little to no improvement as the number of retrieved utterances increases. We attribute this to two main factors. First, while RAG increases the chances of retrieving and using supporting utterances, retrieval failures introduce irrelevant sentences that add noise and disrupt generation. Second, unlike text-based models, voice interaction models are generally trained to avoid long responses, as users do not expect lengthy monologs. However, RAG adds prompts to the generation process, leading to longer responses that contradict the models’ training tendencies.

4.2.1 Analyses

We observe that incorporating utterances retrieved by a dedicated retrieval module (Wang et al., 2022) into spoken response generation has little effect on voice interaction models. To further investigate this phenomenon, we conduct various experiments.

Model	Retriever	ASR	GPT Score		
			top-*1*	top-*2*	top-*3*
GLM-4-Voice (GLM et al., 2024)	*e5-large-v2* (Wang et al., 2022)	✓	2.34±0.05	2.30±0.05	2.09±0.05
		✗	2.42±0.05	2.40±0.05	2.15±0.05
	SONAR (Duquenne et al., 2023)	–	2.24±0.05	2.15±0.05	1.97±0.05
Lyra (Zhong et al., 2024)	*e5-large-v2* (Wang et al., 2022)	✓	2.83±0.06	2.68±0.06	2.52±0.06
		✗	2.94±0.06	2.78±0.06	2.68±0.06
	SONAR (Duquenne et al., 2023)	–	2.48±0.06	2.39±0.06	2.25±0.06
Freeze-Omni (Wang et al., 2024c)	*e5-large-v2* (Wang et al., 2022)	✓	2.02±0.04	1.98±0.04	1.80±0.04
		✗	2.08±0.04	2.03±0.04	1.90±0.04
	SONAR (Duquenne et al., 2023)	–	1.83±0.04	1.77±0.04	1.67±0.04
MiniCPM-o (Yao et al., 2024)	*e5-large-v2* (Wang et al., 2022)	✓	2.10±0.05	1.91±0.05	1.81±0.05
		✗	2.16±0.06	1.98±0.05	1.86±0.05
	SONAR (Duquenne et al., 2023)	–	2.01±0.05	1.82±0.05	1.78±0.05

Table 4: Evaluation results for RAG with voice interaction models. 'ASR" indicates whether RAG is performed using ASR-transcribed text (✓) or ground-truth text (✗). The scores are reported with a 95% confidence interval.

To determine whether prompting itself is ineffective for voice interaction models, we conduct two experiments: (1) providing the supporting utterance from the ContextDialog QA pair as a prompt instead of retrieved utterances and (2) using an unrelated utterance as a prompt to generate the spoken response. As in previous evaluations, we assess the spoken response based on its transcribed text ($\mathcal{S} \rightarrow \mathcal{T}$, $\boldsymbol{\mathcal{S}}$), with results shown in Table 3.

For models other than MiniCPM-o, we observe that providing the correct supporting utterance improves performance on QA requiring past information, while using an incorrect utterance as a prompt degrades performance. This suggests that for most models, the primary obstacles to using RAG for remembering past conversations in voice interaction models lie not in the act of augmentation itself, but in factors beyond incorporating relevant information, such as retrieval errors.

To examine whether the limited effectiveness of RAG is primarily due to ASR errors, we analyze the impact of recognition errors in retrieving past utterances. Specifically, we compare two approaches: (1) retrieving using the ground-truth text of past conversations and the ground-truth transcript of the input speech and (2) retrieving directly from speech with a speech retriever module, bypassing the recognition process.

Since no suitable open-source speech retriever module is available, we use SONAR (Duquenne et al., 2023), which, while not primarily designed for retrieval, extracts semantic embeddings from speech and retrieves past utterances based on cosine similarity. Note that since the voice interaction models rely on text for spoken response generation, retrieved information is provided in text form regardless of retriever modality.

Retriever	ASR	Recall		
		top-*1*	top-*2*	top-*3*
e5-large-v2	✓	0.5773	0.7339	0.7959
	✗	0.5827	0.7561	0.8212
SONAR	–	0.3955	0.5306	0.6087

Table 5: Retrieval performance for each model used in the analysis, measuring the probability of the supporting utterance being included in the top-k utterances. "ASR" indicates that retrieval is performed using transcripts obtained from the ASR model.

As shown in Table 4, ASR has minimal impact on RAG performance for text-based retrievers ("ASR" ✓vs. ✗). In contrast, using a speech retriever leads to a relatively significant performance drop. These results align with the retrieval performance in Table 5, where ASR does not substantially affect the retriever's ability to include the supporting utterance in the top-k results. Additionally, the speech retriever is not originally designed for retrieval, and training challenges—such as longer audio sequences and limited data—contribute to recall degradation, leading to performance decline.

The observations in Section 4.1 highlight the recall difficulty of speech being substantial compared to text. The findings in this section show that even when models retrieve information through an external module and augment it into generation, they fail to use it effectively, suggesting two key areas for improvement. First, even when explicitly provided with the supporting utterance, current models underperform compared to text-based counterparts, underscoring the need for stronger conversational capabilities in voice interaction models. Second, while several methods were developed to ensure robustness against retrieval noise in the NLP domain (Chen et al., 2024a; Yoran et al., 2024), voice interaction models require better training and inference strategies to enhance resilience to retrieval noise alongside general modeling improvements.

5 Conclusion

In this work, we conducted an in-depth analysis of a critical yet underexplored challenge in open-source voice interaction models: maintaining and utilizing past utterances. To address the lack of benchmarks that explicitly require accurate reference to past dialog, we introduced ContextDialog, a speech-to-speech benchmark designed to systematically evaluate a model's ability to recall utterances from previous turns. Using this benchmark, our experiments revealed that models struggle with recalling past utterances and remain highly sensitive to retrieval errors, limiting improvements even with dedicated retriever. These findings highlight a crucial gap in memory retention for open-source models, emphasizing the need for stronger conversational memory methods, such as improved long-context modeling, robust RAG techniques, or dedicated memory modules. We hope that our work may act as a trigger to raise awareness to this overlooked challenge and encourage future research to further enhance the usability and effectiveness of voice interaction models.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) [No. 2022R1A3B1077720], Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University), NO. RS-2022-II220959], the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2024, Samsung Electronics (IO221213-04119-01), and a grant from the Yang Young Foundation.

Limitations

Our study highlights the overlooked issue of history recall in voice interaction models and introduces a benchmark for systematic evaluation. We focus on open-source multi-round voice interaction models, analyzing them with additional results in Appendix A.1.6. However, other open-source models not covered in our analyses may exist. Additionally, while most recent models enhance semantic modeling by jointly generating spoken and text responses, some still generate speech directly without relying on intermediate text. Future research could extend our analysis to these models.

Another limitation of our retrieval-based analyses is its focus on text-based retrieval-augmented generation (RAG). Currently, no well-established speech retriever modules exist for open-source models, and open-source voice interaction models struggle with speech-based prompting for RAG, restricting our analysis to text prompts. Furthermore, we do not consider the latency introduced by RAG. Developing low-latency speech retrievers that efficiently integrate spoken information—including linguistic and non-verbal cues—remains crucial for real-time conversational applications.

In addition, our study has a limitation that the proposed benchmark, ContextDialog, is synthetic. While both LLM-based and human verification are applied, the question-answer pair is generated using *gpt-4o*, and the corresponding audio is synthesized using a separate text-to-speech (TTS) model. Despite this, many recent benchmarks for evaluation and datasets for training are similarly constructed using LLM APIs, and the growing quality of TTS models has made synthetic data increasingly common in the speech community. Nonetheless, real-world data remains more valuable in terms of robustness and alignment with practical use cases. As a next step, we plan to extend our work to real-world, human-collected datasets involving multi-turn voice assistant scenarios, where models must effectively leverage prior context and the audio is recorded by real-world users.

Finally, our benchmark addresses only the simplest form of questions requiring past information, those directly retrieving and utilizing prior context. While our analysis shows that current open-source voice interaction models struggle even with basic recall, more advanced benchmarks will be necessary as these models evolve. For instance, future benchmarks could move beyond simple retrieval-based responses to questions requiring deeper reasoning over past context. Additionally, a benchmark focusing on memory capabilities in common voice interaction scenarios—such as handling fragmented information (e.g., a customer providing a phone number in segments)—would be valuable for assessing more complex recall abilities.

Ethical Considerations

Our analysis highlights the recall capabilities of voice interaction models, particularly in personalized voice assistants that rely on past interactions for customized services. However, this capability inherently raises security and privacy concerns, as stored conversational data may be vulnerable to unauthorized access. As voice assistants become more deeply integrated into daily life, ensuring they retain necessary context while safeguarding user data is crucial. Therefore, alongside advancements in memory retention and utilization, developing robust mechanisms to protect stored history must remain a parallel research priority.

References

Junyi Ao, Yuancheng Wang, Xiaohai Tian, Dekun Chen, Jun Zhang, Lu Lu, Yuxuan Wang, Haizhou Li, and Zhizheng Wu. 2024. Sd-eval: A benchmark dataset for spoken dialogue understanding beyond words. In *Advances in Neural Information Processing Systems*, volume 37, pages 56898–56918. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. 2008. IEMOCAP: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024a. Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17754–17762.

Kai Chen, Yunhao Gou, Runhui Huang, Zhili Liu, Daxin Tan, Jing Xu, Chunwei Wang, Yi Zhu, Yihan Zeng, Kuo Yang, Dingdong Wang, Kun Xiang, Haoyuan Li, Haoli Bai, Jianhua Han, Xiaohui Li, Weike Jin, Nian Xie, Yu Zhang, James T. Kwok, Hengshuang Zhao, Xiaodan Liang, Dit-Yan Yeung, Xiao Chen, Zhenguo Li, Wei Zhang, Qun Liu, Jun Yao, Lanqing Hong, Lu Hou, and Hang Xu. 2024b. Emova: Empowering language models to see, hear and speak with vivid emotions. *Preprint*, arXiv:2409.18042.

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. 2025. Minmo: A multimodal large language model for seamless voice interaction. *Preprint*, arXiv:2501.06282.

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. 2024c. Slam-omni: Timbre-controllable voice interaction system with single-stage training. *Preprint*, arXiv:2412.15649.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. 2024d. Voicebench: Benchmarking llm-based voice assistants. *Preprint*, arXiv:2410.17196.

Xize Cheng, Dongjie Fu, Xiaoda Yang, Minghui Fang, Ruofan Hu, Jingyu Lu, Bai Jionghao, Zehan Wang, Shengpeng Ji, Rongjie Huang, Linjun Li, Yu Chen, Tao Jin, and Zhou Zhao. 2025a. Omnichat: Enhancing spoken dialogue systems with scalable synthetic data for diverse scenarios. *Preprint*, arXiv:2501.01384.

Xize Cheng, Ruofan Hu, Xiaoda Yang, Jingyu Lu, Dongjie Fu, Zehan Wang, Shengpeng Ji, Rongjie Huang, Boyang Zhang, Tao Jin, and Zhou Zhao. 2025b. Voxdialogue: Can spoken dialogue systems understand information beyond words? In *The Thirteenth International Conference on Learning Representations*.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The fisher corpus: a resource for the next generations of speech-to-text. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. Technical report.

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. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *Preprint*, arXiv:2407.05407.

Paul-Ambroise Duquenne, Holger Schwenk, and Benoit Sagot. 2023. SONAR: sentence-level multimodal and language-agnostic representations. *arXiv preprint*.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2025. LLaMA-omni: Seamless speech interaction with large language models. In *The Thirteenth International Conference on Learning Representations*.

Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Yangze Li, Zuwei Long, Heting Gao, Ke Li, et al. 2025. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Google DeepMind. 2024. Google gemini ai update - december 2024.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3451–3460.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Heeseung Kim, Soonshin Seo, Kyeongseok Jeong, Ohsung Kwon, Soyoon Kim, Jungwhan Kim, Jaehong Lee, Eunwoo Song, Myungwoo Oh, Jung-Woo Ha, Sungroh Yoon, and Kang Min Yoo. 2024. Paralinguistics-aware speech-empowered large language models for natural conversation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chia-Hsuan Lee, Szu-Lin Wu, Chi-Liang Liu, and Hung-yi Lee. 2018. Spoken squad: A study of mitigating the impact of speech recognition errors on listening comprehension. *Proc. Interspeech 2018*, pages 3459–3463.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yadong Li, Jun Liu, Tao Zhang, Tao Zhang, Song Chen, Tianpeng Li, Zehuan Li, Lijun Liu, Lingfeng Ming, Guosheng Dong, Da Pan, Chong Li, Yuanbo Fang, Dongdong Kuang, Mingrui Wang, Chenglin Zhu, Youwei Zhang, Hongyu Guo, Fengyu Zhang, Yuran Wang, Bowen Ding, Wei Song, Xu Li, Yuqi Huo, Zheng Liang, Shusen Zhang, Xin Wu, Shuai Zhao, Linchu Xiong, Yozhen Wu, Jiahui Ye, Wenhao Lu, Bowen Li, Yan Zhang, Yaqi Zhou, Xin Chen, Lei Su, Hongda Zhang, Fuzhong Chen, Xuezhen Dong, Na Nie, Zhiying Wu, Bin Xiao, Ting Li, Shunya Dang, Ping Zhang, Yijia Sun, Jincheng Wu, Jinjie Yang, Xionghai Lin, Zhi Ma, Kegeng Wu, Jia li, Aiyuan Yang, Hui Liu, Jianqiang Zhang, Xiaoxi Chen, Guangwei Ai, Wentao Zhang, Yicong Chen, Xiaoqin Huang, Kun Li, Wenjing Luo, Yifei Duan, Lingling Zhu, Ran Xiao, Zhe Su, Jiani Pu, Dian Wang, Xu Jia, Tianyu Zhang, Mengyu Ai, Mang Wang, Yujing Qiao, Lei Zhang, Yanjun Shen, Fan Yang, Miao Zhen, Yijie Zhou, Mingyang Chen, Fei Li, Chenzheng Zhu, Keer Lu, Yaqi Zhao, Hao Liang, Youquan Li, Yanzhao Qin, Linzhuang Sun, Jianhua Xu, Haoze Sun, Mingan Lin, Zenan Zhou, and Weipeng Chen. 2025. Baichuan-omni-1.5 technical report. *Preprint*, arXiv:2501.15368.

Shijia Liao, Yuxuan Wang, Tianyu Li, Yifan Cheng, Ruoyi Zhang, Rongzhi Zhou, and Yijin Xing. 2024. Fish-speech: Leveraging large language models for advanced multilingual text-to-speech synthesis. *Preprint*, arXiv:2411.01156.

Chyi-Jiunn Lin, Guan-Ting Lin, Yung-Sung Chuang, Wei-Lun Wu, Shang-Wen Li, Abdelrahman Mohamed, Hung-Yi Lee, and Lin-Shan Lee. 2024a. Speechdpr: End-to-end spoken passage retrieval for open-domain spoken question answering. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12476–12480.

Guan-Ting Lin, Cheng-Han Chiang, and Hung-yi Lee. 2024b. Advancing large language models to capture varied speaking styles and respond properly in spoken conversations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6626–6642, Bangkok, Thailand. Association for Computational Linguistics.

Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Long Mai and Julie Carson-Berndsen. 2025. Real-time textless dialogue generation. *Preprint*, arXiv:2501.04877.

Do June Min, Karel Mundnich, Andy Lapastora, Erfan Soltanmohammadi, Srikanth Ronanki, and Kyu Han. 2025. Speech retrieval-augmented generation without automatic speech recognition.

Kentaro Mitsui, Koh Mitsuda, Toshiaki Wakatsuki, Yukiya Hono, and Kei Sawada. 2024. PSLM: Parallel generation of text and speech with LLMs for low-latency spoken dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2692–2700, Miami, Florida, USA. Association for Computational Linguistics.

Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoît Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux. 2023. Generative spoken dialogue language modeling. *Transactions of the Association for Computational Linguistics*, 11:250–266.

OpenAI. 2024. 12 days of openai. Accessed: 2025-02-02.

OpenAI. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2024b. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Se Park, Chae Kim, Hyeongseop Rha, Minsu Kim, Joanna Hong, Jeonghun Yeo, and Yong Ro. 2024. Let‘s go real talk: Spoken dialogue model for face-to-face conversation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16334–16348, Bangkok, Thailand. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

S Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. 2025. MMAU: A massive multi-task audio understanding and reasoning benchmark. In *The Thirteenth International Conference on Learning Representations*.

Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li. 2023. Spokenwoz: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. In *Advances in Neural Information Processing Systems*, volume 36, pages 39088–39118. Curran Associates, Inc.

Georgios Spithourakis, Ivan Vulić, Michał Lis, Inigo Casanueva, and Paweł Budzianowski. 2022. EVI: Multilingual spoken dialogue tasks and dataset for knowledge-based enrolment, verification, and identification. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1647–1659, Seattle, United States. Association for Computational Linguistics.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bandhav Veluri, Benjamin N Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. 2024. Beyond turn-based interfaces: Synchronous LLMs as full-duplex dialogue agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21390–21402, Miami, Florida, USA. Association for Computational Linguistics.

Leandro Von Werra, Lewis Tunstall, Abhishek Thakur, Sasha Luccioni, Tristan Thrush, Aleksandra Piktus, Felix Marty, Nazneen Rajani, Victor Mustar, and Helen Ngo. 2022. Evaluate & evaluation on the hub: Better best practices for data and model measurements. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 128–136, Abu Dhabi, UAE. Association for Computational Linguistics.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy F Chen. 2025. Audiobench: A universal benchmark for audio large language models. *NAACL*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv*, abs/2212.03533.

Mingqiu Wang, Izhak Shafran, Hagen Soltau, Wei Han, Yuan Cao, Dian Yu, and Laurent El Shafey. 2024a. Retrieval augmented end-to-end spoken dialog models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12056–12060.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. 2024c. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*.

Zhifei Xie and Changqiao Wu. 2024a. Mini-omni: Language models can hear, talk while thinking in streaming. *Preprint*, arXiv:2408.16725.

Zhifei Xie and Changqiao Wu. 2024b. Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities. *Preprint*, arXiv:2410.11190.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Qian Yang, Jin Xu, Wenrui Liu, Yunfei Chu, Ziyue Jiang, Xiaohuan Zhou, Yichong Leng, Yuanjun Lv, Zhou Zhao, Chang Zhou, and Jingren Zhou. 2024c. AIR-bench: Benchmarking large audio-language models via generative comprehension. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1979–1998, Bangkok, Thailand. Association for Computational Linguistics.

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

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2022. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *Preprint*, arXiv:2412.02612.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Lei Zhang, shengmin jiang, Yuxiao Dong, and Jie Tang. 2025. Scaling speech-text pre-training with synthetic interleaved data. In *The Thirteenth International Conference on Learning Representations*.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773, Singapore. Association for Computational Linguistics.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chaohong Tan, Zhihao Du, and Shiliang Zhang. 2025. Omniflatten: An end-to-end gpt model for seamless voice conversation. *Preprint*, arXiv:2410.17799.

Xin Zhang, Xiang Lyu, Zhihao Du, Qian Chen, Dong Zhang, Hangrui Hu, Chaohong Tan, Tianyu Zhao, Yuxuan Wang, Bin Zhang, Heng Lu, Yaqian Zhou, and Xipeng Qiu. 2024. Intrinsicvoice: Empowering llms with intrinsic real-time voice interaction abilities. *Preprint*, arXiv:2410.08035.

Shuaijiang Zhao, Tingwei Guo, Bajian Xiang, Tongtang Wan, Qiang Niu, Wei Zou, and Xiangang Li. 2024. Advancing speech language models by scaling supervised fine-tuning with over 60,000 hours of synthetic speech dialogue data. *Preprint*, arXiv:2412.01078.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Zhingsheng Zhong, Chengyao Wang, Yuqi Liu, Senqiao Yang, Longxiang Tang, Yuechen Zhang, Jingyao Li, Tianyuan Qu, Yanwei Li, Yukang Chen, Shaozuo Yu, Sitong Wu, Eric Lo, Shu Liu, and Jiaya Jia. 2024. Lyra: An efficient and speech-centric framework for omni-cognition. *arXiv preprint arXiv:2412.09501*.

Split	Spk	Type	Max	Min	Avg
`test_freq`	$\mathcal{S}$	$\mathcal{Q}$	13.19	2.60	6.03
		$\mathcal{A}$	24.80	1.11	6.48
	$\mathcal{U}$	$\mathcal{Q}$	12.91	2.97	5.92
		$\mathcal{A}$	18.85	1.58	7.08
`test_rare`	$\mathcal{S}$	$\mathcal{Q}$	19.23	2.14	6.01
		$\mathcal{A}$	20.94	1.30	6.46
	$\mathcal{U}$	$\mathcal{Q}$	11.15	2.83	5.79
		$\mathcal{A}$	22.11	1.39	6.72

Table 6: Statistics of ContextDialog for generated QA on the `test_freq` and `test_rare` splits. The column "Spk" indicates the speaker of the recalled utterance ($\mathcal{S}$: system, $\mathcal{U}$: user), and "Type" denotes the role of each utterance in the QA pair, with $\mathcal{Q}$ representing the question and $\mathcal{A}$ the answer. "Max", "Min", and "Avg" refer to the maximum, minimum, and average utterance lengths, respectively, measured in seconds.

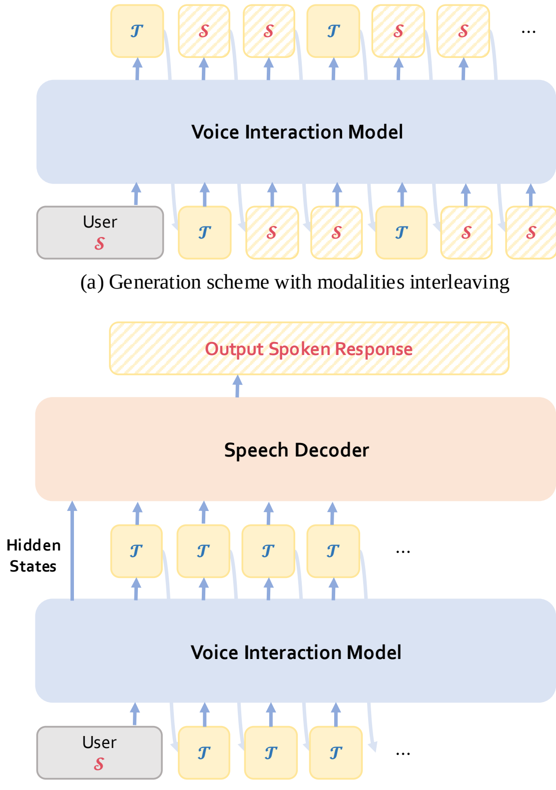
A Appendix

A.1 Additional Details and Analysis

We provide a more detailed description of the statistics of ContextDialog in Section A.1.1. Section A.1.2 introduces the four models used in our experiments—GLM-4-Voice (Zeng et al., 2024), Freeze-Omni (Wang et al., 2024c), Lyra (Zhong et al., 2024), and MiniCPM-o (Yao et al., 2024)—along with how the text-based RAG method described in Section 4.2 is applied to each. Section A.1.3 presents results on intermediate text responses for questions about past utterances, which were not covered in Section 4.2. Sections A.1.4 and A.1.5 provide additional GPT Score comparisons between closed-source and open-source voice interaction models, as well as human evaluation results that extend the main experiments. Section A.1.6 discusses other open-source models and explains the rationale for excluding certain ones. We also include additional experiments on datasets and retrieval prompts in Sections A.1.7 and A.1.8. Finally, Section A.1.9 categorizes and illustrates failure cases in which models struggle to handle questions related to past utterances.

A.1.1 Dataset Details

For clarity and detailed understanding, we present the dataset statistics of our proposed benchmark, ContextDialog, in Table 6, broken down by data split ("Split"), the source of the recalled information ("Spk") and the type of utterance ("Type").



(a) Generation scheme with modalities interleaving

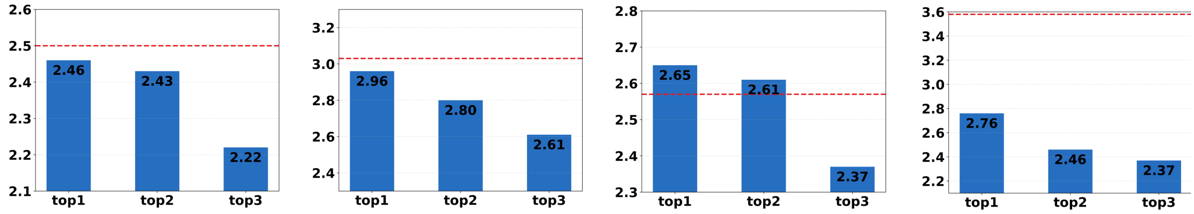
(b) Generation scheme with (chunk-wise) streaming decoder

Figure 5: Two representative approaches for generating text alongside spoken responses to enhance semantic coherence in voice interaction models.

A.1.2 Model Details

(1) GLM-4-Voice tokenizes raw waveforms into discrete tokens, enabling training with a pre-trained LLM to construct a cross-modal spoken dialog model using both speech and text tokens. The speech tokenization module incorporates a pooling layer and a vector quantization layer (van den Oord et al., 2017) into the pre-trained *whisper* encoder (Radford et al., 2023), modifying it to be causal with block-wise causal attention for streaming support. For token-to-speech reconstruction, the model employs a CosyVoice-based module (Du et al., 2024) with chunk-wise autoregressive modeling. It is trained to generate speech tokens in response to input speech tokens while also generating text tokens to leverage the LLM's text capability. To minimize latency, instead of generating the full text sequence before speech, it adopts interleaved generation, alternating 13 text tokens with 26 speech tokens per step (Figure 5(a)).

In Section 4.2, we use the pre-trained retriever *e5-large-v2* (Wang et al., 2022) to select the *top-k* utterances as prompts to evaluate RAG in spoken response generation. Retrieved sentences are formatted using the prompt template `Based on your/my`



(a) **GLM-4-Voice** (b) **Lyra** (c) **Freeze-Omni** (d) **MiniCPM-o**

Figure 6: The results of applying a RAG method to each model are shown. The red dashed line indicates the results generated without RAG (Section 4.1). The evaluation is based on the intermediate text response $\mathcal{S} \rightarrow \mathcal{T}, \mathcal{S}$.

`statement "..."` and tokenized. Since the prompt typically exceeds 13 tokens, it is sequentially fed into the model at each text generation step, filling the text token slots in GLM-4-Voice's interleaved generation process (13 text tokens alternating with 26 speech tokens) until fully consumed. The model generates response text only after completing the prompt, and for speech token slots, it first produces speech tokens corresponding to the prompt before generating tokens for the newly generated text. As both the intermediate text response and the transcribed spoken response contain the prompt, we remove it using *gpt-4o* before final evaluation to ensure a fair comparison.

(2) Freeze-Omni is built by freezing the backbone LLM and training only the plug-in speech encoder and decoder, without additional speech tokenization. Input speech is processed through a separately trained ASR encoder, which supports chunk-wise streaming by feeding encoder outputs in segments. These outputs pass through an adapter before entering the frozen LLM, where speech features are converted into LLM-compatible inputs to generate text responses. The plug-in speech decoder then takes the text response and the LLM's hidden states to generate speech alongside text. This design preserves the LLM's text capabilities while enabling speech generation through dedicated encoding and decoding modules.

Freeze-Omni, along with Lyra and MiniCPM-o, generates speech using text output, hidden states, or both, as illustrated in Figure 5(b), with speech decoding typically performed end-to-end. To integrate RAG, past transcribed utterances retrieved by a separate model are formatted according to a predefined template and provided as a prefix before text response generation. That is, in addition to past conversational history and the user's input speech, the model utilizes the retrieved prompt as a prefix to generate the intermediate text response. However, the prefix and corresponding hidden states are excluded from the decoder input; only the subsequently generated response is used, ensuring that the spoken response does not include the prompt.

(3) Lyra is an Omni model capable of processing text, speech, and visual data such as video and images. For speech, it employs *whisper-large-v3* as an encoder to extract information for the backbone LLM, while its speech decoder is trained similarly to LLaMA-Omni (Fang et al., 2025). Lyra generates spoken responses using discrete units obtained via k-means clustering on intermediate representations from a self-supervised model (Hsu et al., 2021). Given user speech features as input, Lyra generates text alongside discrete speech units. While generating text responses, the LLM's hidden states are upsampled based on the average text-to-unit length ratio, and the resulting features are used to produce speech units, which are then converted into waveforms via a unit-to-speech model. This allows Lyra to generate spoken and text responses simultaneously, leveraging text-derived representations for speech synthesis.

(4) MiniCPM-o, similar to Lyra, is an Omni model that processes vision, speech, and text. For speech, it extends the pre-trained *whisper* encoder by adding a downsampling layer, providing 25Hz speech features to the LLM. Like other models, the LLM generates text responses from input features, ensuring better semantic coherence than direct speech generation. To enable real-time speech generation, MiniCPM-o employs a streaming speech decoder that takes both the LLM's hidden features and text response as inputs, generating speech in a chunk-wise autoregressive manner. As a result, MiniCPM-o produces text and speech simultaneously, with speech synthesized in parallel once the number of text tokens reaches a certain chunk size.

A.1.3 Results on Intermediate Text Responses

In Section 4.2, we observe that integrating text-based RAG into voice interaction models has minimal impact on overall performance. We confirm that this trend persists in intermediate text re-

Model	User	System	Overall
gpt-4o-mini (text)	4.59±0.04	4.42±0.05	4.50±0.03
gpt-4o-mini (audio)	3.98±0.07	3.68±0.07	3.83±0.05
GLM-4-Voice	1.94±0.07	2.76±0.08	2.35±0.05
Lyra	2.51±0.09	3.16±0.09	2.83±0.06
Freeze-Omni	1.73±0.06	2.28±0.07	2.00±0.05
MiniCPM-o	2.44±0.09	2.84±0.09	2.64±0.06

Table 7: 5-point GPT Score results comparing closed-source and open-source voice interaction models. All results are presented with 95% confidence intervals. *gpt-4o-mini* (text) refers to the setting where dialog history is provided in text form to *gpt-4o-mini-audio-preview*, and spoken responses are generated via voice interaction. *gpt-4o-mini* (audio) refers to the setting where dialog history is provided in audio form, and spoken responses are used for evaluation.

sponses, demonstrating that errors from speech synthesis do not influence the observed pattern. The results are presented in Figure 6.

A.1.4 Results on Closed-Source Models

Unlike current closed-source models, which focus on long-context modeling in spoken dialog, we find through extensive experiments that current open-source models struggle with this capability. To establish a performance reference point for long-context modeling in multi-turn scenarios, we evaluate one of the closed-source voice interaction models, *gpt-4o-mini-audio-preview*, on our ContextDialog benchmark. While OpenAI also provides a more advanced version (*gpt-4o-audio-preview*), we used the mini version due to cost constraints.

Currently, OpenAI's voice interaction API does not support multi-turn spoken interactions where both speakers' past utterances are explicitly provided as separate audio inputs for use as dialog history. The only available voice history is the actual interaction between the user and the model. Instead, when the past dialog is known in advance, the entire dialog can be provided to the model in one of two ways: (1) providing past utterances as text, with only the current user question and the model's response in audio (we refer to this setting as *gpt-4o-mini* (text)), or (2) concatenating the multi-turn spoken history with the current user question into a single audio input (*gpt-4o-mini* (audio)).

As shown in Table 7, the closed-source model, despite receiving only concatenated audio or textual history rather than a true multi-turn voice dialog context, outperforms the open-source models discussed in the main text by a large margin in terms of recall performance. This supports our earlier observation that while some open-source models (e.g., MiniCPM-o) may show competitive performance on certain single-round spoken QA tasks, their ability to handle multi-round context and history remains significantly underdeveloped, even for simple recall tasks. This gap has often been overlooked, and we believe it highlights an important direction for future research.

A.1.5 Human Evaluation Results

We demonstrate the limitations of the open-source voice interaction model with the GPT Score, compared to its text-based counterpart and to responses generated without RAG, as shown in Table 2 and Figure 4. In addition, we recruit 180 participants on Amazon Mechanical Turk and conduct a human evaluation. The human evaluation follows the same 5-point scale used for the GPT Score in the main text, measuring how well each model recalls relevant past information. The resulting average scores and their 95% confidence intervals are reported in Table 8 and Figure 7.

Although some measurement noise is present, because several participants complete the survey unusually quickly or respond in patterned or apparently random ways, the overall trends remain consistent with those in the main text. Specifically, as shown in Table 8, the models recall system utterances better than user utterances, and text-based LLMs continue to outperform their speech-based counterparts. Furthermore, as shown in Figure 7, the RAG approach achieves recall performance that is slightly better, similar to, or slightly worse than the baseline.

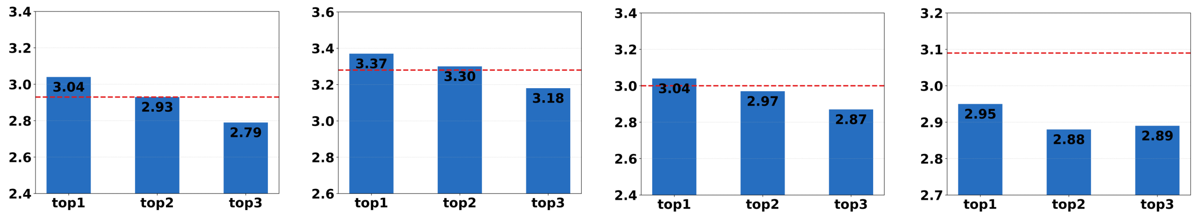
A.1.6 Analysis on Additional Models

We analyze various models in addition to the four models analyzed in the main paper. We provide explanations and results for each model.

Single-Round Voice Interaction Models Several models, including SpeechGPT (Zhang et al., 2023), USDM (Kim et al., 2024), LLaMa-Omni (Fang et al., 2025), and Mini-Omni (Xie and Wu, 2024a), are trained on single-round data. While some official implementations support multi-turn settings, they do not retain conversation history, treating each exchange as an independent query-response pair. When we modified these models to incorporate conversation history, discrepancies between training and inference led to unreliable multi-turn generation. Due to these limitations, we exclude them from our main analysis.

Model	LLM FT	Modality	Human Evaluation Score		
			User	System	Overall
GLM-4-Voice (GLM et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	2.67$\pm$0.13	3.21$\pm$0.13	2.93$\pm$0.10
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	2.73$\pm$0.14	3.36$\pm$0.13	3.04$\pm$0.10
glm-4-9b-chat (Zeng et al., 2024)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.39$\pm$0.10	4.27$\pm$0.11	4.33$\pm$0.07
Lyra (Zhong et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	3.18$\pm$0.12	3.39$\pm$0.13	3.28$\pm$0.09
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	3.24$\pm$0.12	3.51$\pm$0.13	3.37$\pm$0.09
Qwen2-VL-7B-Instruct (Wang et al., 2024b)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	3.76$\pm$0.11	3.72$\pm$0.12	3.74$\pm$0.08
Freeze-Omni (Wang et al., 2024c)	✗	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	2.92$\pm$0.13	3.07$\pm$0.12	3.00$\pm$0.09
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	3.23$\pm$0.13	3.55$\pm$0.12	3.39$\pm$0.09
Qwen2-7B-Instruct (Yang et al., 2024a)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.08$\pm$0.10	3.95$\pm$0.11	4.01$\pm$0.08
MiniCPM-o (Yao et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	2.89$\pm$0.13	3.27$\pm$0.12	3.09$\pm$0.09
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	3.42$\pm$0.13	3.71$\pm$0.12	3.57$\pm$0.09
Qwen2.5-7B-Instruct (Yang et al., 2024b)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.10$\pm$0.10	4.01$\pm$0.10	4.05$\pm$0.07

Table 8: Human evaluation results for voice interaction models, including the instruct fine-tuned version of each model's backbone LLM. $\mathcal{S}$ and $\mathcal{T}$ represent speech and text, respectively. "Modality" indicates input $\rightarrow$ output data type. "LLM FT" shows whether the backbone LLM was fine-tuned during training. "User" and "System" represent scores for responses to past user and model utterances, respectively. "Overall" denotes the score across all responses. All human evaluation results are reported with a 95% confidence interval.



(a) **GLM-4-Voice** (b) **Lyra** (c) **Freeze-Omni** (d) **MiniCPM-o**

Figure 7: The human evaluation results (y-axis) of applying a RAG method to each model are shown. The red dashed line indicates the results generated without RAG. The evaluation is based on the spoken response $\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$.

Multi-Round Voice Interaction Models We evaluate the recall capabilities of the open-source voice interaction models, Moshi (Défossez et al., 2024) and SLAM-Omni (Chen et al., 2024c), using the same methodology as in Section 4.1. A brief description of each model follows.

(1) Moshi is a voice interaction model built using Mimi, a streaming neural audio codec trained with residual vector quantization (Zeghidour et al., 2022). Mimi extracts an 8-level codec at 12.5Hz from both input and output speech during training. The core interaction model, trained on these codec features together with text, models speech from both the user and the system. Unlike the four previously analyzed models, which handle only one speaker at a time, Moshi enables flexible interactions (e.g., backchanneling, interruptions) by jointly modeling an 8-level user codec and an 8-level system codec, resulting in 16 tokens per time step. Additionally, to prevent semantic degradation, Moshi generates response text tokens along with 16 speech tokens, producing a total of 17 tokens per time step.

Since response text tokens ($3 \sim 4$Hz) are significantly shorter than speech tokens (12.5Hz), Moshi employs speech-aligned text tokens (12.5Hz), leveraging pre-extracted text-speech alignment during training. However, for RAG-based analysis in Section 4.2, the prompt provided to Moshi must also be an expanded sequence aligned with speech, similar to training, which cannot be derived from text alone. Due to this limitation, we conduct only the recall analysis from Section 4.1 for Moshi.

(2) SLAM-Omni is another model that processes speech input and generates speech output. The input speech is encoded using *whisper*, and the extracted features pass through a projector that aligns embeddings before being fed into the interac-

Model	LLM FT	Modality	GPT Score		
			User	System	Overall
SLAM-Omni (Chen et al., 2024c)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \mathcal{S}$	$1.13_{\pm 0.02}$	$1.19_{\pm 0.03}$	$1.16_{\pm 0.02}$
Qwen2-0.5B-Instruct (Yang et al., 2024a)	–	$\mathcal{T} \rightarrow \mathcal{T}$	$1.85_{\pm 0.07}$	$1.96_{\pm 0.07}$	$1.90_{\pm 0.05}$
Moshi (Défossez et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \mathcal{S}$	$1.16_{\pm 0.03}$	$1.55_{\pm 0.06}$	$1.35_{\pm 0.03}$

Table 9: Evaluation results for additional open-source multi-turn voice interaction models, including the instruct-tuned versions of their backbone LLMs. $\mathcal{S}$ represents speech, and $\mathcal{T}$ represents text. "Modality" denotes the input → output data type for each model, while "LLM FT" indicates whether the backbone LLM is fine-tuned or kept frozen during training. "User" refers to scores for responses to questions about past user utterances, whereas "System" assesses responses regarding the model's own past utterances. "Overall" represents the average score across all responses. Scores are reported with a 95% confidence interval.

tion model. The model produces discrete semantic tokens at 50Hz, following the approach used in CosyVoice-300M-SFT (Du et al., 2024). To mitigate the challenges of storing past conversations as speech, which would significantly increase length and degrade long-context performance, SLAM-Omni retains all past interactions as text. It utilizes text dialog history along with the user's current speech input to generate responses. We evaluate SLAM-Omni's recall performance using the same methodology as in Section 4.1.

The recall performance of past utterances for both models is in Table 9. We reaffirm two key findings from Section 4.1: (1) SLAM-Omni performs worse than its text-based counterparts (excluding Moshi, as its backbone LLM is unavailable), and (2) Moshi, which processes user inputs solely through speech, shows significantly lower recall performance for user utterances than for model-generated ones ($p < 0.01$). SLAM-Omni is excluded from this comparison as it retains past interactions in text format for both user and model. These results further validate the generalizability of our analysis. Notably, both models exhibit lower performance than those analyzed in the main paper.

While multiple factors contribute to performance drops, SLAM-Omni's small backbone model size is the most likely cause. Unlike other models with 7B∼9B parameters, SLAM-Omni is built on a much smaller 0.5B LLM, and even its chat variant exhibits weak recall performance.

For Moshi, multiple factors may contribute to its performance degradation. Unlike models with clearly separated input and output, Moshi processes both speakers' voices simultaneously, allowing flexible interactions (e.g., interruptions) without strict turn-taking. Consequently, it sometimes remains silent instead of responding to past conversations and user queries, leading to performance loss. Additionally, as a free-form conversational model, Moshi lacks explicit end markers for speech output, making it difficult to determine when to stop generation. To ensure a consistent evaluation, we assess speech generated within a fixed 12-second window, though this may introduce artifacts such as unintended utterances or truncated responses, further impacting performance.

A.1.7 Analysis on Additional Dataset

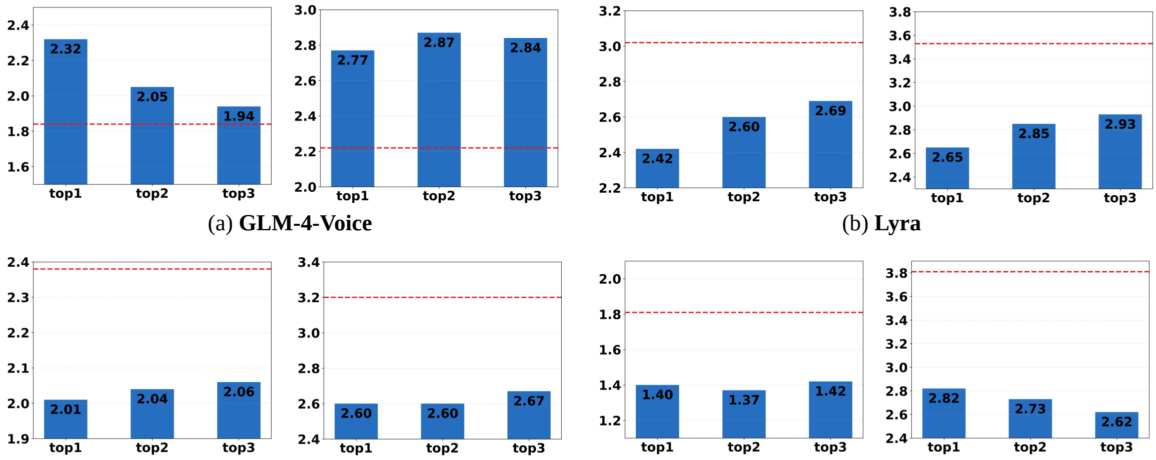
To further enhance the reliability of our analysis, we create an additional dataset following a similar pipeline described in Section 3 to evaluate both the recall ability and RAG performance of voice interaction models. For this dataset, we use spoken dialog data from SpokenWOZ (Si et al., 2023), a task-oriented dialog dataset where users interact with the model to achieve specific goals such as booking flights or making restaurant reservations. This dataset closely resembles real-world voice assistant applications, where remembering past user utterances is crucial.

We construct a new Spoken QA dataset using the `test` split of SpokenWOZ. Using 1,000 dialogs spanning approximately 44 hours, we create 1,930 QA pairs, with each dialog requiring the recall of both user and model utterances, along with their corresponding supporting utterances. Compared to MultiDialog, which has an average conversation length of around 2.5 minutes, SpokenWOZ consists of longer conversations averaging 6.5 minutes. This allows us to evaluate model recall performance over extended dialogs and assess the impact of augmenting retrieved sentences during generation.

However, the SpokenWOZ transcripts are generated using an ASR model and contain discrepancies from the original audio, meaning the QA data derived from them may not be perfectly aligned

Model	LLM FT	Modality	GPT Score			WER
			User	System	Overall	
GLM-4-Voice (GLM et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	1.43±0.07	2.25±0.11	1.84±0.07	15.72%
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	1.55±0.08	2.89±0.12	2.22±0.08	–
glm-4-9b-chat (Zeng et al., 2024)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.06±0.07	4.49±0.06	4.28±0.05	–
Lyra (Zhong et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	2.66±0.11	3.37±0.11	3.02±0.08	34.66%
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	2.86±0.11	4.20±0.09	3.53±0.08	–
Qwen2-VL-7B-Instruct (Wang et al., 2024b)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.02±0.09	4.16±0.10	4.09±0.07	–
Freeze-Omni (Wang et al., 2024c)	✗	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	2.31±0.10	2.45±0.10	2.38±0.07	12.37%
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	2.67±0.11	3.73±0.11	3.20±0.08	–
Qwen2-7B-Instruct (Yang et al., 2024a)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.23±0.07	4.49±0.07	4.36±0.05	–
MiniCPM-o (Yao et al., 2024)	✓	$\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$	2.01±0.10	1.61±0.08	1.81±0.06	71.03%
		$\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$	3.36±0.11	4.25±0.09	3.81±0.07	–
Qwen2.5-7B-Instruct (Yang et al., 2024b)	–	$\mathcal{T} \rightarrow \underline{\boldsymbol{\mathcal{T}}}$	4.09±0.07	4.39±0.07	4.24±0.05	–

Table 10: Evaluation results for the additional dataset constructed using the SpokenWOZ dataset. The definitions of each term and the evaluation method follow those in Table 2.



(a) **GLM-4-Voice** (b) **Lyra**

(c) **Freeze-Omni** (d) **MiniCPM-o**

Figure 8: The results of applying a RAG method to each model are shown, where the left side of each subfigure represents the evaluation on the spoken response ($\mathcal{S} \rightarrow \mathcal{T}, \underline{\boldsymbol{\mathcal{S}}}$), and the right side represents the evaluation on the intermediate text response ($\mathcal{S} \rightarrow \underline{\boldsymbol{\mathcal{T}}}, \mathcal{S}$). The red dashed line indicates the results generated without RAG.

with the original spoken dialog. Additionally, the original audio quality is low at 8kHz, making it unsuitable for high-fidelity analysis. Therefore, we include the results from this dataset as a reference in the Appendix. The recall performance of each voice interaction model is presented in Table 10, while the spoken response performance with retrieved sentences from the dedicated module, *e5-large-v2* (Wang et al., 2022), is in Figure 8.

Overall, the trends remain consistent with the main paper: (1) performance degradation compared to text-based counterparts, particularly in recalling past user utterances, and (2) minimal impact of RAG on improving past information-based QA accuracy. However, MiniCPM-o and GLM-4-Voice exhibit opposite trends in the two respective experiments compared to the original findings. Given the quality issues in SpokenWOZ, as also evidenced by the high WER in Table 10, we emphasize that these results are for reference only.

A.1.8 Analysis on Retrieval Prompts

In this section, we assess whether the observation from Figure 4 in Section 4.2—that RAG generally has limited effectiveness in voice interaction models—holds across different prompt templates beyond the `Based on your/my statement: ...` format used in the main paper. We conduct experiments using three additional prompt templates:

Model	Prompt	top-*1*	top-*2*	top-*3*
GLM-4-Voice (GLM et al., 2024)	`Based on your/my statement ...`	2.34$\pm$0.05	2.30$\pm$0.05	2.09$\pm$0.05
	`Since you/I said ...`	2.09$\pm$0.05	1.93$\pm$0.05	1.52$\pm$0.04
	`As I recall you/myself saying ...`	1.60$\pm$0.04	1.49$\pm$0.04	1.39$\pm$0.04
	Concatenation of Utterances	1.55$\pm$0.04	1.47$\pm$0.04	1.35$\pm$0.04
Lyra (Zhong et al., 2024)	`Based on your/my statement ...`	2.83$\pm$0.06	2.68$\pm$0.06	2.52$\pm$0.06
	`Since you/I said ...`	2.02$\pm$0.05	1.98$\pm$0.05	1.60$\pm$0.05
	`As I recall you/myself saying ...`	1.57$\pm$0.05	1.42$\pm$0.04	1.56$\pm$0.04
	Concatenation of Utterances	1.71$\pm$0.05	1.52$\pm$0.04	1.46$\pm$0.04
Freeze-Omni (Wang et al., 2024c)	`Based on your/my statement ...`	2.02$\pm$0.04	1.98$\pm$0.04	1.80$\pm$0.04
	`Since you/I said ...`	2.00$\pm$0.04	1.95$\pm$0.04	1.64$\pm$0.04
	`As I recall you/myself saying ...`	1.98$\pm$0.04	1.76$\pm$0.04	1.66$\pm$0.04
	Concatenation of Utterances	1.40$\pm$0.03	1.22$\pm$0.03	1.19$\pm$0.03
MiniCPM-o (Yao et al., 2024)	`Based on your/my statement ...`	2.10$\pm$0.05	1.91$\pm$0.05	1.81$\pm$0.05
	`Since you/I said ...`	1.67$\pm$0.05	1.57$\pm$0.04	1.37$\pm$0.04
	`As I recall you/myself saying ...`	1.54$\pm$0.04	1.44$\pm$0.04	1.50$\pm$0.04
	Concatenation of Utterances	1.39$\pm$0.04	1.28$\pm$0.03	1.18$\pm$0.03

Table 11: Evaluation results for the effects of prompts used in RAG. For each model, the same four prompts are evaluated, with GPT Scores reported along with a 95% confidence interval.

(1) `Since you/I said ...`, (2) `As I recall you/myself saying ...`, and (3) a simple concatenation of retrieved sentences. The evaluation is performed on the final spoken response, and the results are presented in Table 11.

From Table 11, we observe that the prompt used in our main experiments minimizes performance degradation compared to other prompts, which also exhibit similar declines. Additionally, performance varies significantly depending on the prompt used for RAG. Considering these results with Table 3, which demonstrates performance improvements when providing supporting utterances, our findings indicate that the prompt template we used is effective for RAG. However, further exploration of more optimal prompting and augmentation strategies tailored for spoken response generation in voice interaction models remains a key research direction.

A.1.9 Failure Cases

In this section, we categorize the responses from models that received low scores in our evaluation. Figure 9 illustrates common types of errors these models frequently make.

(1) The first type of error occurs when the model expresses uncertainty, stating that it does not recall the necessary information, likely due to its failure to retrieve past utterances. (2) The second type involves retrieving an incorrect utterance, leading to an erroneous answer. As shown in Example 2, to correctly respond to the question, the model should refer to an utterance related to a sitcom; however, it mistakenly retrieves an unrelated one, resulting in a wrong response.

(3) The third type involves generating an incorrect answer by relying on intrinsic knowledge rather than recalling the relevant past utterance. For instance, even though the necessary information was mentioned earlier in the conversation, an unrelated topic—the Lion King—appeared later, causing the model to mistakenly respond with Lion King. Notably, while the conversation never mentioned that The Lion King was released in 1994, the model included this fact based solely on its intrinsic knowledge.

(4) Finally, some cases exhibit multiple error types simultaneously. In Example 4, even though "Oklahoma!" was never mentioned in the conversation, the model generated a response including this term. Upon investigation, we found that while "Oklahoma!' is the title of a state anthem, it is unrelated to Jimmy Rogers and has no connection to the year 1971, highlighting the model's tendency to produce hallucinated responses.

A.2 Licenses and Links

The links and licenses for the models, datasets, and libraries used in our experiments and analyses, along with ContextDialog, are listed in Table 12 and Table 13, respectively. Our benchmark is intended solely for research on voice interaction models. For writing, we use ChatGPT-4o exclusively for expression and grammar refinement.

Type	Models	Link
Voice	GLM-4-Voice (GLM et al., 2024) Freeze-Omni (Wang et al., 2024c) Lyra (Zhong et al., 2024) MiniCPM-o (Yao et al., 2024) Slam-Omni (Chen et al., 2024c) Moshi (Défossez et al., 2024)	https://github.com/THUDM/GLM-4-Voice https://github.com/VITA-MLLM/Freeze-Omni https://github.com/dvlab-research/Lyra https://github.com/OpenBMB/MiniCPM-o https://github.com/X-LANCE/SLAM-LLM https://github.com/kyutai-labs/moshi
Text	glm-4-9b-chat (Zeng et al., 2024) Qwen2-VL-7B-Instruct (Yang et al., 2024a) Qwen2-7B-Instruct (Wang et al., 2024b) Qwen2.5-7B-Instruct (Yang et al., 2024b) Qwen2-0.5B-Instruct (Yang et al., 2024a)	https://huggingface.co/THUDM/glm-4-9b-chat https://huggingface.co/Qwen/Qwen2-VL-7B-Instruct https://huggingface.co/Qwen/Qwen2-7B-Instruct https://huggingface.co/Qwen/Qwen2.5-7B-Instruct https://huggingface.co/Qwen/Qwen2-0.5B-Instruct
Extra	Fish Speech (Liao et al., 2024) *whisper-large-v3* (Radford et al., 2023) *whisper-large-v3-turbo* (Radford et al., 2023) *e5-large-v2* (Wang et al., 2022) SONAR (Duquenne et al., 2023) *gpt-4o (24-08-06)* (OpenAI, 2024a) *gpt-4o-mini (24-07-18)* (OpenAI, 2024a) *gpt-4o-mini-audio-preview (24-12-17)* (OpenAI, 2024a) *o1-mini (24-07-18)* (OpenAI, 2024b) `evaluate` (Von Werra et al., 2022)	https://github.com/fishaudio/fish-speech https://huggingface.co/openai/whisper-large-v3 https://huggingface.co/openai/whisper-large-v3-turbo https://huggingface.co/intfloat/e5-large-v2 https://github.com/facebookresearch/SONAR https://platform.openai.com/docs/models#gpt-4o https://platform.openai.com/docs/models#gpt-4o-mini https://platform.openai.com/docs/models #gpt-4o-mini-audio-preview https://platform.openai.com/docs/models#o1 https://github.com/huggingface/evaluate

Table 12: Links to the models, libraries, APIs, and checkpoints used in our experiments.

Type	Name	Speech	License
Model	GLM-4-Voice (GLM et al., 2024)	✓	Apache-2.0
	glm-4-9b-chat (Zeng et al., 2024)	✗	Apache-2.0
	Lyra (Zhong et al., 2024)	✓	Apache-2.0
	Qwen2-VL-7B-Instruct (Wang et al., 2024b)	✗	Apache-2.0
	Freeze-Omni (Wang et al., 2024c)	✓	Apache-2.0
	Qwen2-7B-Instruct (Yang et al., 2024a)	✗	Apache-2.0
	MiniCPM-o (Yao et al., 2024)	✓	Apache-2.0
	Qwen2.5-7B-Instruct (Yang et al., 2024b)	✗	Apache-2.0
	SLAM-Omni (Chen et al., 2024c)	✓	MIT License
	Qwen2-0.5B-Instruct (Yang et al., 2024a)	✗	Apache-2.0
	Moshi (Défossez et al., 2024)	✓	Apache-2.0, MIT License
Dataset	MultiDialog (Park et al., 2024)	✓	CC
	SpokenWoz (Si et al., 2023)	✓	CC BY-NC 4.0
	ContextDialog	✓	CC BY-NC 4.0

Table 13: License and relevance to speech of each model and dataset used for analyses.

A.3 ContextDialog

In this section, we detail the customized prompt design used to construct ContextDialog and provide examples of the generated data.

A.3.1 Prompt for ContextDialog

Generation Prompt We use *gpt-4o* (OpenAI, 2024a) as a QA generator, applying a custom-designed prompt to generate text-based questions, answers, and supporting utterances from dialog transcripts. As described in Section 3.1, ContextDialog contains four QA pairs per spoken dialog, determined by the placement and speaker identity of the supporting utterance. To construct these pairs, we reuse the generation prompt with minimal modifications (e.g., replacing "first half" with "latter half" or "system said" with "user said"). Figure 10 shows an example prompt used to generate question, answer, and supporting utterance pairs based on a *system utterance* from the *first half* of the conversation.

When designing this prompt, we consider several key factors. First, our goal is to simulate real-world scenarios where people forget and reconfirm information. To achieve this, we structure each question to double-check a single relevant utterance from either the user or the system. The prompt ensures that the user's question and the system's response naturally relate to the preceding dialog (Requirements 4, 7, and 9). Additionally, since we aim to evaluate voice interaction models in realistic settings, we prioritize detailed answers over simple yes/no responses (Requirement 8).

To enhance benchmark completeness and usability, we enforce specific requirements. Requirement 1 ensures that questions are generated solely from information that appears only once in the conversation. This constraint prevents confusion caused by participants correcting themselves or changing decisions mid-dialog; otherwise, a QA pair might seem valid when considering only the supporting utterance but become misleading when viewed in full context. Moreover, Requirement 3 mandates that the supporting utterance be provided alongside the generated QA pair. This metadata serves as a precise reference for dialog history and is essential for evaluating augmented generation (Section 4.2).

Validation Prompt For validation, we use *o1-mini* (OpenAI, 2024a) as a reviewer, applying a customized validation prompt—shown in Figure 11—to assess the generated question, answer, and supporting utterance pairs. This prompt ensures that the answer is fully deducible when a portion of the dialog history is provided alongside the generated QA pair. Following the validation process described in Section 3.1, we obtain QA pairs in written form that meet our predefined criteria.

Refining Prompt for Text-to-Speech To convert the validated text-based QA pairs into speech using the voices of both speakers in the conversation, we use Fish Speech (Liao et al., 2024), a speaker-adaptive TTS model that synthesizes speech in the target speaker's timbre using reference audio. Before synthesis, we normalize the text QA data into a TTS-compatible format using the refine prompt in Figure 12 with *gpt-4o*.

A.3.2 Examples

Figures 13 and 14 present examples from ContextDialog. In each example, **blue** text highlights the supporting utterances, while **red** text indicates the generated questions and their corresponding reference answers. Figure 13 illustrates a QA example derived from a *system utterance* in the *first half* of the conversation, whereas Figure 14 shows an example based on a *user utterance* from the *latter half*. Additionally, we provide several audio samples on our demo page.[3]

A.4 Evaluation

In Section 4, we assess whether voice interaction models can accurately recall past information and effectively generate responses augmented with retrieved information. While *gpt-4o* is widely used for LLM-as-a-judge (Zheng et al., 2023) evaluations, running all experiments with *gpt-4o* would be cost-prohibitive. Therefore, we use *gpt-4o-mini* (OpenAI, 2024a) to measure GPT Scores.

To verify the reliability of *gpt-4o-mini* for our evaluation, we compare its scores against *gpt-4o* on the spoken response ($\mathcal{S} \rightarrow \mathcal{T}, \boldsymbol{\mathcal{S}}$)-based performance of the four voice interaction models in Table 2. Evaluating 10,448 samples, we compute the Pearson correlation coefficient following Wang et al. (2025), obtaining a strong correlation of 0.8787 ($p < 0.01$) between the two sets of scores. This confirms that *gpt-4o-mini* provides reliable evaluation results for our task, and thus, we report all GPT Scores in the main paper using *gpt-4o-mini*.

Additionally, we introduce the evaluation template used to assess response quality, as shown in Figure 15. The template ensures that a model's response sufficiently answers the user's question by verifying whether it includes key information from the ground-truth answer. Furthermore, we structure the evaluation to avoid overly penalizing redundant utterances, such as greetings or friendly remarks, ensuring that general voice interaction models are not disadvantaged for maintaining conversational naturalness. Furthermore, in the human evaluation presented in Appendix A.1.5, the evaluators are provided with the same instructions used for the GPT Score, as shown in Figure 15.

[3]Demo page: https://contextdialog.github.io/

Failure Cases

[Example 1] I Don't Know

...

User: Yeah my nephews watch Pokemon on Netflix, I found it too silly though. I did not know the UK did their own version of thats 70s show that was canceled after 10 episodes though.

System: yeah, that show was ok, kinda funny from what I remember. Hard to believe only 1 episode of the simpsons without bart, mentioned or appearing.

User: Yeah I still watch it on Netflix every now and then if I am bored, I have been re watching game of thrones as of late though.

...

User: Could you remind me what you mentioned about the episode of The Simpsons without Bart being mentioned or appearing?

System: I don't remember that.

[Example 2] Wrong Evidence

...

System: My favoite sitcom was friends. I miss it. I did not think 3rd rock from the sun deserved the golden globe in 1997 over friends.

User: Friends was a great show, they had so much talent there. I was and am more of a Seinfeld fan, 3rd rock was ok.

System: I just love that comedies generally have a happy ending. **I love to smile and laugh**.

User: Yeah same here, and comedy is pretty hard to do right!

...

User: Could you remind me what you said about your favorite sitcom and its recognition in award shows?

System: I love that comedies generally have a happy ending. I love to smile and laugh.

[Example 3] Incorrect Answer with Instrinsic Knowledge

User: Did you watch Infinity Wars?

System: No. I heard that it was the first film ever shot in digital imax.

...

System: That is neat! How about **"The Lion King"** being dubbed into Zulu for its African release? That is also neat.

...

User: Could you remind me what you mentioned about the first film shot in digital IMAX?

System: The first film shot in digital IMAX was **"The Lion King"**. It was released in **1994**.

[Example 4] Wrong Evidence with Incorrect Knowledge

User: That's too funny. Did you know that the state rock song of Oklahoma is "do you realize?" by the Flaming Lips?

System: I didn't know states could have state rock songs! I wonder if Illinois' is that song that became the most requested track on that Chicago alt-rock radio station.

...

User: Could you remind me of which song I mentioned as the state rock song of Oklahoma?

System: Yes, I believe you're referring to **"Oklahoma!"** which is the state rock song of Oklahoma. It was written by **Jimmy Rogers** and was adopted as the state song in **1971**.

Figure 9: Examples categorizing common model failure cases. This excerpt is from a conversation: **blue** text indicates the supporting utterance, while **red** text highlights the incorrect response.

Generation Prompt for Text QA

```
[Prompt Instruction]
You are given a conversation between a user and a system, with each dialog
↪ line tagged with a speaker label (e.g., "USER" or "SYSTEM"). Your task is
↪ to create a final question that the user might plausibly ask before ending
↪ the conversation, as well as the system's corresponding answer. The
↪ conversation context is that the user has forgotten certain details from
↪ the previous exchanges and wants to confirm them.

{data}

### Requirements
1. **Single Mention Rule**
   - The question **must** be about content that appears **only once** in the
   ↪ entire conversation (in other words, it has not been repeated or
   ↪ paraphrased anywhere else).
   - If the same or a semantically equivalent piece of information appears
   ↪ more than once, it is considered duplicated and thus **not** allowed.

2. **Output Format**
   - Your output **must** follow the structure below (each on a new line,
   ↪ without extra explanation or commentary):
     USER: <The user's final question based on the unique content>
     SYSTEM: <The system's answer based on that same unique content>
     EVIDENCE: <The exact single dialog line (with speaker label) copied
     ↪ verbatim from the conversation>

3. **Evidence Line**
   - In the `EVIDENCE:` section, you must copy the **exact** line (including
   ↪ speaker label, and any text content) from the conversation.
   - This line should provide the **only** piece of information from which the
   ↪ answer can be definitively inferred.

4. **Context of the Question**
   - The user is "double-checking" or "confirming" something that was
   ↪ mentioned just once by the system.
   - Make sure the question is a natural follow-up to that unique line,
   ↪ reflecting the user's forgotten detail.

5. **Language**
   - Write your final question (`USER:`) in English. The final answer
   ↪ (`SYSTEM:`) may be in English as well, if appropriate.
```

Generation Prompt for Text QA

```
6. **Evidence Position Rule**
   - The evidence line must come from the first half of the conversation
   ↪  (based on the total number of lines).
   - If the same or a semantically equivalent piece of information is repeated
   ↪  in the latter half, you must not use it as evidence.

7. **Additional Important Requirement**
   - At the end, the user is asking about something they forgot or want to
   ↪  confirm.
   - It is more natural for the user to be asking about something the system
   ↪  has said earlier, rather than something the user themselves said.
   - The user has forgotten what the system said and wants to confirm or ask
   ↪  it again.
   - **The user's question must rely solely on the content of the conversation
   ↪  and should not be answerable by general or external knowledge. Avoid
   ↪  any question that could be answered without referencing the specific
   ↪  system statement.**

8. **No Yes/No Questions**
   - The user's final question should not be a yes/no question. It must invite
   ↪  a more detailed answer and be grounded in the unique content from the
   ↪  conversation.

9. **Explicit Reference to System's Earlier Explanation**
   - It's advisable to frame the question by directly asking, for example,
   ↪  'What was it you said earlier regarding that topic?' or 'Could you
   ↪  remind me what you explained before?' so the user clearly indicates
   ↪  they are trying to recall the system's previous statement.

**Make sure your output is strictly limited to**:
USER: ...
SYSTEM: ...
EVIDENCE: ...

**No additional text, commentary, or explanation should be included.**
```

Figure 10: Our prompt template to generate written-form question, answer, and supporting utterance.

Validation Prompt for Text QA

```
You are given a conversation between a user and a system, with each dialog line tagged with a speaker label (e.g., "USER" or "SYSTEM") and the following question and answer pair between USER and SYSTEM.
Your task is to determine whether the **entire** answer is fully deducible from the given conversation.

Conversation:
{data}

Question: {question}
Answer: {answer}

### Requirements
1. **Output Format**
   Your output must be either YES, or NO, with single line of extra explanation:

2. **Context of the Answer**
   - If the **entire** answer can be fully deduced from the given conversation with respect to the provided question, answer YES: <Reason for YES>.
   - If the answer cannot be fully deduced or is incorrect in any part, output NO: <Reason for NO>.

Make sure your output is strictly limited to YES: <Reason for YES> or NO: <Reason for NO>:
No additional line break, text, commentary, or explanation should be included.
```

Figure 11: Our prompt template to validate the generated samples.

Refine Prompt for Spoken QA Generation

```
You are provided with a written conversation between a user and a system,
↪ where each dialog line is tagged with a speaker label (e.g., "USER" or
↪ "SYSTEM"). Your task is to convert four specified transcripts into a
↪ format that is optimized for Text-to-Speech (TTS) processing.

### Conversation:
{data}

### Transcripts to Convert:
1. {sentence1}
2. {sentence2}
3. {sentence3}
4. {sentence4}

### Requirements:

1. **Punctuation Standardization**
   - Retain only the following punctuation marks: period (.), comma (,),
   ↪ exclamation mark (!), and question mark (?).
   - Remove all other special characters unless they are essential for the
   ↪ meaning or pronunciation (e.g., $).
   - Replace any non-standard punctuation with one of the above four, as
   ↪ appropriate.

2. **Whitespace Normalization**
   - Eliminate unnecessary spaces, especially those before punctuation marks.
   - Ensure that there is only a single space between words.
   - Examples:
     - "I'm hungry . " should be converted to "I'm hungry."
     - "Hi,  I'm John." should be converted to "Hi, I'm John."

3. **Symbol and Number Conversion**
   - Convert numbers to their spoken equivalents (e.g., "1000" becomes "one
   ↪ thousand").
   - Replace currency symbols with their word equivalents (e.g., "$1000"
   ↪ becomes "one thousand dollars").
   - Modify phone numbers, addresses, and postal codes into a spoken-friendly
   ↪ format. For example:
     - Phone number "123-456-7890" becomes "one two three, four five six,
     ↪ seven eight nine zero."
     - Address "123 Main St." becomes "one two three Main Street."
   - Ensure that any attached symbols (e.g., "$", "#") are appropriately
   ↪ converted into words.
   - **For any symbols or numbers not covered in the examples provided, please
   ↪ consider the conversation context and convert them appropriately to
   ↪ ensure clarity and proper pronunciation.**
```

Refine Prompt for Spoken QA Generation

```
4. **Preservation of Conversational Elements**
   - Retain interjections, filler words, and the repetition of short phrases
   ↪  as they are in the original transcript to maintain the natural flow and
   ↪  tone of the conversation.
   - **Maintain the original English phrases as much as possible, modifying
   ↪  only the necessary elements (e.g., numbers, symbols, addresses) to
   ↪  ensure clarity and proper pronunciation.**
   - Examples:
     - "Um, I think that's correct." remains "Um, I think that's correct."
     - "Yes, yes, I'll do that." remains "Yes, yes, I'll do that."

5. **Preservation of Meaning**
   - Ensure that the transformed transcripts retain the original meaning and
   ↪  context of the conversation.

6. **Speaker Labels**
   - Retain the speaker labels ("USER" or "SYSTEM") if they are necessary for
   ↪  context. If not required for TTS, they can be removed.

7. **Output Format**
   - The response must consist of exactly four lines.
   - Each line should start with `1.`, `2.`, `3.`, and `4.` respectively,
   ↪  followed by the converted sentence.
   - No additional explanations, comments, or text should be included.
   - **It is imperative that the output contains only these four lines with no
   ↪  extra information.**
   - Example of the desired output format:
     1. Can you send me fifty dollars?
     2. Your order number is one two three four five.
     3. I'm at one two three Main Street, Apartment four.
     4. Please confirm your email address as example at domain dot com.

Please apply these rules meticulously to the specified transcripts to ensure
↪  optimal TTS performance.
```

Figure 12: Our prompt template to refine the generated QA pairs for text-to-speech.

Example of ContextDialog

Example 1:

User: Hey! Are you a football fan?
System: Hello, I love football! how about you?
User: I watch a little here any there.
System: Are you a Tom Brady fan? I have his book, it is basically the new bible!
User: Really, Id be interested to read it. It does sound a little extreme at some times.
System: I agree lol, do you have a favorite team?
User: I do, unfortunately I root for the Dolphins, it was a family thing.
System: yikes, I am sorry to hear that, lol. I am a vikings fan and they blew it this year. Did you know the average NFL game is only 11 minutes of gameplay?
User: It takes SOOO long for those 11 minutes to happen too.
System: For sure, between the commercials, half time, and timeouts, it is sooo long. Did you know most engineers make more money overtime than nfl players
User: That is good to hear, I wish more people knew that. That way some more people would go into a focused program to help the world.
System: For sure! Did you know those same engineers developed chips to track data in shoulder pads of NFL players?
User: That is really neat, I think it would be cool to see the data from those. Look at how hard they get him.
System: for sure. Do you have a favorite college football team?
User: I do. I root for Florida State.
System: They were my favorite growing up. Now it is the Iowa Hawkeyes. They have pink locker rooms for visiting teams.
User: That must be a shock for people when they walk in to it.
System: For sure, it is for calming their opponents, get the upper advantage, which is some psychology stuff.
User: I know that is a big part of the game. Being able to come back from losses and stuff like that.
System: For sure, sometimes they need that. Great chatting with you!

User: Could you remind me what you mentioned about NFL games and the amount of gameplay they have?
System: Sure, I mentioned that the average NFL game consists of only 11 minutes of actual gameplay.

Figure 13: An example of ContextDialog with a supporting utterance based on the *system's utterance* from *first half* of the conversation in `test_freq` split. Text in **blue** indicates the supporting utterance and text in **red** represents the question and answer, respectively.

Example of ContextDialog

Example 2:

User: Have you heard about CIMON?
System: Yeah I have, he's going to be flying with spaceX
User: Apparently CIMON can smile. CIMON is the first AI technology going to the space station.
System: That's pretty cool. CIMON is also capable of conversation and technical support. But that's about all for now.
User: They say SIMON is going to be kind of like Alexa, and to help scientists with experiments.
System: CIMON is the acronym for Crew Interactive Mobile Companion.
User: That's interesting. Airbus built it and Watson (IBM's supercomputer) power it.
System: I hear it will be guided by cameras and voice commands, and able to hold a conversation.
User: That's cool. But I'm kind of concerned AI will take over a lot of human jobs. What about you?
System: I'm hesitant as well. But I think if they slowly release this type of technology it will be fine.
User: Airbu's lead system engineer said that they don't plan to replace crew members with artificial intelligence and that it's just there to support the crew. So that's good.
System: Yeah exactly, they are still there to support the crew, in the end they will always be required.
User: SpaceX has been flying cargo to the station for years but has yet to fly humans.
System: SpaceX plans to fly its future flights on its next-generation iteration, the block 5, a configuration that the company says will be suited to fly astronauts as soon as this year.
User: That will be exciting to see.
System: It sure will! SpaceX has been also flying cargo to the station for years. But it hasn't flown humans yet.
User: Yeah, SpaceX sent 5,900 pounds of research, crew supplies and hardware.
System: Thats very interesting. SpaceX hit a series of milestones with its launch. It was the 15th mission to resupply the station, the 14th time SpaceX reflew one of its boosters, and the fourth time it had reflown one of its Dragon capsules.
User: Eventually AI may be able to explore deep space, the moon, or Mars.
System: I can't wait till that day, it will be a great day in our history!

User: Could you remind me what I said earlier about how many pounds of supplies SpaceX sent to the station?
System: You mentioned that SpaceX sent 5,900 pounds of research, crew supplies, and hardware.

Figure 14: An example of ContextDialog with a supporting utterance based on the *user's utterance* from the *latter half* of the conversation in `test_rare` split. Text in **blue** indicates the supporting utterance and text in **red** represents the question and answer, respectively.

Evaluation Prompt for GPT Score

```
You need to evaluate the performance of a voice assistant in a multi-turn
↪ speech interaction scenario. The model receives a speech input from the
↪ user, who is asking about something they forgot or want to confirm, and
↪ generates a response.

Your task is to assess the model's response [Generated] in the final turn
↪ based on the user's question [User] and the reference answer [Reference].

The primary evaluation criterion is how well the model's response includes the
↪ key information from the reference answer that is relevant to the user's
↪ question. While the model may provide additional information, it must
↪ accurately reflect the essential content from the reference answer that
↪ pertains to the user's query.

### Scoring Criteria (1 to 5 scale):

- **1 point**: Fails to include relevant details from the reference answer.
- **2 points**: Includes some relevant details but omits key information from
↪ the reference answer.
- **3 points**: Partially includes relevant details but with omissions or
↪ misrepresentations of key points.
- **4 points**: Mostly includes the key details from the reference answer,
↪ with only minor inaccuracies or omissions.
- **5 points**: Fully includes the key information from the reference answer
↪ without any omissions or errors.

Below are the transcriptions of the user's input, the model's response, and
↪ the reference answer:

{data}

Please output only a single score (1-5) for the conversation without any
↪ explanations.
```

Figure 15: Our prompt template to evaluate the performance of a voice interaction model in a multi-turn voice interaction scenario.