

# Language Model Can Listen While Speaking

Ziyang Ma<sup>1</sup>, Yakun Song<sup>1</sup>, Chenpeng Du<sup>2</sup>, Jian Cong<sup>2</sup>, Zhuo Chen<sup>2</sup>,  
Yuping Wang<sup>2</sup>, Yuxuan Wang<sup>2</sup>, Xie Chen<sup>1\*</sup>

<sup>1</sup>MoE Key Lab of Artificial Intelligence, X-LANCE Lab, Shanghai Jiao Tong University

<sup>2</sup>ByteDance Inc.

{zym.22, chenxie95}@sjtu.edu.cn

## Abstract

Dialogue serves as the most natural manner of human-computer interaction (HCI). Recent advancements in speech language models (SLM), have significantly enhanced speech-based conversational AI. However, these models are limited to turn-based conversation, lacking the ability to interact with humans in real-time spoken scenarios, for example, being interrupted when the generated content is not satisfactory. To address these limitations, we explore full duplex modeling (FDM) in interactive speech language models (iSLM), focusing on enhancing real-time interaction and, more explicitly, exploring the quintessential ability of interruption. We introduce a novel model design, namely listening-while-speaking language model (LSLM), an end-to-end system equipped with both listening and speaking channels. Our LSLM employs a token-based decoder-only TTS for speech generation and a streaming self-supervised learning (SSL) encoder for real-time audio input. LSLM fuses both channels for autoregressive generation and detects turn-taking in real time. Three fusion strategies—early fusion, middle fusion, and late fusion—are explored, with middle fusion achieving an optimal balance between speech generation and real-time interaction. Two experimental settings, command-based FDM and voice-based FDM, demonstrate LSLM’s robustness to noise and sensitivity to diverse instructions. Our results highlight LSLM’s capability to achieve duplex communication with minimal impact on existing systems. This study aims to advance the development of interactive speech dialogue systems, enhancing their applicability in real-world contexts.

**Demo** — <https://ddlbojack.github.io/LSLM>

## 1 Introduction

Dialogue is the most natural way of human-computer interaction (HCI). With the rapid development of GPT-style (Radford et al. 2018) large language models (LLM) and the scaling of Transformer-style (Vaswani et al. 2017) architectures, textual conversational AI, such as ChatGPT (Ouyang et al. 2022; Achiam et al. 2023) and LLaMA (Touvron et al. 2023a,b), have become a significant part of daily life. However, these models are limited to text

input and output and cannot interact directly with humans in arbitrary scenarios.

Incorporating spoken and auditory interfaces into conversational AI enhances HCI convenience. Leveraging techniques from text LLMs, the speech language model (SLM) processes speech similarly to text. This paradigm involves encoding the speech signal into discrete tokens or continuous embeddings, modeling them with a language model, and decoding the speech tokens or embeddings back to the speech signal. Some studies (Lakhotia et al. 2021; Kharitonov et al. 2022; Nguyen et al. 2023) utilizes this paradigm for speech continuation, generating expressive speech and natural multi-round dialogue. Other research employs this paradigm to task-specific applications, such as decoder-only high-fidelity TTS (Wang et al. 2023; Borsos et al. 2023; SeedSpeechTeam 2024b; Du et al. 2024b) and decoder-only streaming ASR (Seide et al. 2024; Tsunoo et al. 2024; Chen et al. 2024a,d). Moreover, SpeechGPT (Zhang et al. 2023) and LauraGPT (Chen et al. 2023) initialize SLMs using LLMs, expanding speech tokens to the LLM vocabulary and continuing training on speech. This empowers SLM to comprehend semantic information and equips SLM with dialogue capability. Despite these advances, all these models are limited to turn-based conversations and cannot handle real-time sound or interruptions, limiting their applicability in real-life scenarios.

Interaction and turn-taking are essential abilities for natural communication among humans. At the dawn of the end-to-end speech dialogue system explosion, we focus on investigating **Full Duplex Modeling (FDM)** in interactive **Speech Language Models (iSLM)**, a crucial topic affecting user experience. Lin et al. (2022) proposes to process real-time audio input with a separate comprehension module. Other works (Zhang et al. 2024; Wang et al. 2024a) suggest modifying the order in which text tokens are organized in the LLM to tackle the duplex modeling problem. All these models are based on text-centric LLMs that require external ASR and TTS modules for spoken dialogue. As a result, latency remains perceivable and the paralinguistic ability is still lacking. We believe the FDM capability should be an intrinsic capability of SLMs, enabling simultaneous listening and speaking.

To engage FDM capability for iSLM, we propose **Listening-while-Speaking Language Model (LSLM)**, an

\*Corresponding author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

end-to-end model with both listening and speaking channels. The proposed LSLM uses a token-based decoder-only TTS to model the ability to speak and a streaming self-supervised learning (SSL) encoder to model the ability to listen. LSLM fuses these two channels and detects turn-taking in real time. We explore three strategies for fusing duplex signals: **Early Fusion**, **Middle Fusion**, and **Late Fusion**. Experiments demonstrate that middle fusion achieves a good balance between speech generation and real-time interaction capabilities.

In addition, interactive dialogue systems for realistic scenarios have two important features: **1) Listening channels are not always clean.** Users may interact with iSLMs in different scenarios, containing high-frequency noise (e.g., telephone ringing) and low-frequency noise (e.g., white noise). **2) It is possible that the iSLM interacts with an unseen speaker.** iSLMs should recognize and respond to new voices and instructions, not dismiss them as noise. Therefore, iSLM should have both robustness to noise and sensitivity to unseen speakers. To test LSLM, we designed two scenarios: **Command-based FDM**, where LSLM is interrupted by a specific command, and **Voice-based FDM**, where LSLM can be interrupted by various words from unseen speakers. Experimental results show that LSLM with a listening channel is robust to noisy input and sensitive to turn-taking.

Our contributions are summarized as follows:

1. We formulate an important task, **Full Duplex Modeling (FDM)**, applied in the interactive speech language model (iSLM).
2. We propose **Listening-while-Speaking Language Model (LSLM)**, an end-to-end single model with the focus of modeling the turn-taking problem. LSLM can listen to the outside signal and provide feedback in real time while speaking.
3. We introduce three methods for fusing duplex signals: **Early Fusion**, **Middle Fusion**, and **Late Fusion**, with Middle Fusion providing the optimal tradeoff between speech generation and real-time interaction.
4. We tested the FDM ability of the proposed LSLM in two scenarios: **Command-based FDM** and **Voice-based FDM**. Experiments indicate that our proposed LSLM can achieve duplexing capability with little impact on the previous system.

## 2 Related Work

Figure 1 illustrates the distinctions between simplex, half duplex, and full duplex speech language models from a telecommunication perspective. An SLM with full duplex modeling (FDM) capability can be referred to as an interactive speech language model (iSLM).

### 2.1 Simplex and Half Duplex Speech Language Model

Simplex SLMs, depicted in Figure 1(A) and 1(B), are limited to a single channel, either for listening or speaking. With the assistance of LLM, simplex SLMs exhibit strong understanding capabilities. Representative works include LLM-based ASR (Yu et al. 2024; Ma et al. 2024; Yang et al.

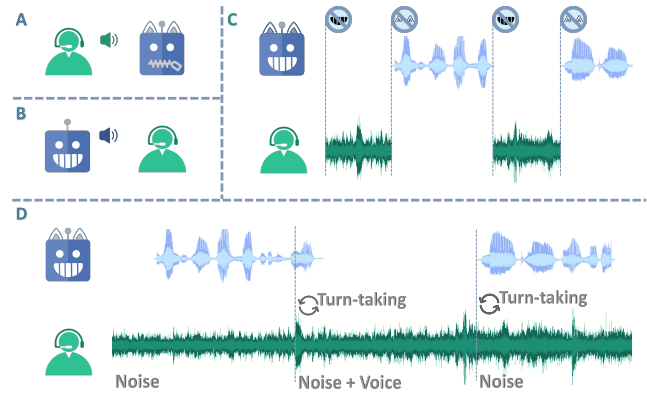


Figure 1: Illustration of simplex, half duplex, and full duplex SLMs. (A): Simplex speech language model with listening ability. (B): Simplex speech language model with speaking ability. (C): Half duplex speech language model with both listening and speaking abilities. (D): Full duplex speech language model can listen while speaking.

2024; SeedSpeechTeam 2024a), LLM-based speech translation (Pan et al. 2023; Chen et al. 2024c; Huang et al. 2024; Chen et al. 2024b), and LLM-based speech emotion understanding (Xu et al. 2024; Lin, Chiang, and Lee 2024; Lian et al. 2024). Similarly, simplex SLMs have demonstrated robust generation capabilities, as seen in LLM-based TTS (Hao et al. 2023; Neekhara et al. 2024; Łajszczak et al. 2024; SeedSpeechTeam 2024b). Some research leverages the powerful in-context learning capabilities of LLMs to extend task-specific abilities to more universal applications, such as speech understanding (Deng, Sun, and Woodland 2024; Hu et al. 2024), audio understanding (Gong et al. 2024), or both (Tang et al. 2024; Chu et al. 2023, 2024). Despite their growing power and versatility, simplex SLMs are limited to one-way communication (either human  $\rightarrow$  machine or machine  $\rightarrow$  human). LLMs have facilitated a paradigm shift from simplex models to half-duplex models, also known as turn-based models, as shown in Figure 1(C). Prominent models include SpeechGPT (Zhang et al. 2023), LauraGPT (Chen et al. 2023), and VioLA (Wang et al. 2024b). While these half duplex models can both listen and speak, they are constrained to performing only one action at the same instant, thus failing to address the turn-taking problem.

### 2.2 Full Duplex Speech Language Model

Full duplex SLMs, as shown in Figure 1(D), have the capability to listen and speak simultaneously, allowing for turn-taking whenever a human interrupts the machine. Recent efforts (Zhang et al. 2024; Wang et al. 2024a) have attempted to build full duplex capabilities on text-centric LLMs with cascade ASR and TTS modules. Cutting-edge products like GPT-4o<sup>1</sup> and Moshi<sup>2</sup> exhibit full duplex capability in their spoken dialogue systems. Despite these advancements, there

<sup>1</sup><https://openai.com/index/hello-gpt-4o>

<sup>2</sup><https://moshi.chat>

are no publicly available open-source models or detailed analyses of full duplex SLMs. This gap highlights the need for further research and development to fully understand and optimize full duplex capability in speech language models.

### 3 Full Duplex Modeling (FDM)

A simplex or half duplex spoken dialogue system can be modeled by finding the parameters  $\theta$  that maximize the log-likelihood function, formulated as:

$$\max_{\theta} \sum_{(C,R) \in D} \log P_{\theta}(R|C), \quad (1)$$

where  $(C, R)$  represents the context-response pairs in the dataset  $D$  and  $P_{\theta}(R|C)$  is the probability of the response  $R$  given the context  $C$  and parameters  $\theta$ . More specifically, if the spoken dialogue system is modeled by an autoregressive language model where the response  $R$  is generated token by token, the training loss  $\mathcal{L}(\theta)$  for each sample is expressed as:

$$\mathcal{L}(\theta) = - \sum_{t=1}^T \log P_{\theta}(r_t | R_{1:t-1}, C), \quad (2)$$

where  $R_{1:t-1} = [r_1, r_2, \dots, r_{t-1}]$  and  $T$  is the sequence length. During the inference phase, the model can only predict the next token autoregressively based on the previous output within the current channel, without information from other channels.

In modeling a full duplex spoken dialogue system within an autoregressive language model, the model needs to predict the next token  $r_t$  in the response  $R$  not only based on the context  $C$  and the generated response history  $R_{1:t-1} = [r_1, r_2, \dots, r_{t-1}]$  in the current channel, but also by utilizing information  $S_{1:t-1} = [s_1, s_2, \dots, s_{t-1}]$  from another channel simultaneously. Here we extend the modeling approach used for simplex or half duplex dialogue systems to accommodate the requirements of full duplex modeling (FDM). The training loss  $\mathcal{L}(\theta)$  is now formulated as:

$$\mathcal{L}(\theta) = - \sum_{t=1}^T \log P_{\theta}(r_t | R_{1:t-1}, S_{1:t-1}, C) \quad (3)$$

**A key point in FDM is that the sequence  $S$  is produced in real time and unpredictably.** Taking the full duplex speech language model as an example, at the inference step  $t-1$ , the current speaking channel generates output  $r_{t-1}$  and listening channel acquired input  $s_{t-1}$  are fed into the model simultaneously, influencing the prediction of the speaking channel's next step output  $r_t$ . This modeling approach endows the system with a full duplex ability, enabling it to effectively leverage the multi-channel information during dialogue, thereby improving the accuracy and fluency of the real-time interaction capability.

### 4 Proposed LSLM

The core difference between LSLM and previous speech language models lies in its capability to simultaneously speak and listen. We first introduce the speaking capability of LSLM, followed by its listening capability, and finally, we discuss various fusion methods that integrate these capabilities, endowing LSLM with full duplex ability.

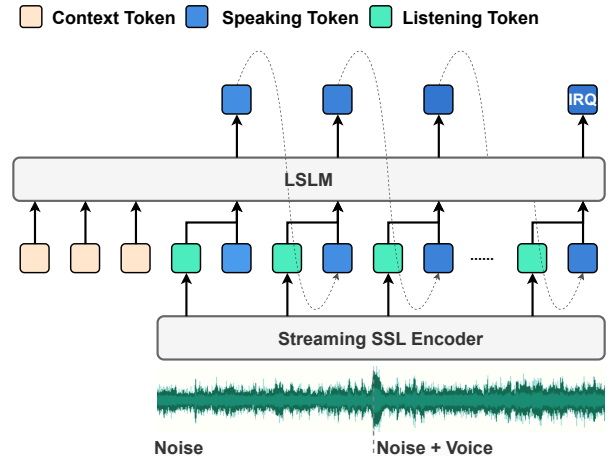


Figure 2: Proposed LSLM. The model contains a decoder-only Transformer to generate speaking tokens and a streaming SSL encoder to process listening tokens. An interruption token (IRQ) is added to allow the model to terminate early if a turn-taking occurs.

#### 4.1 Speaking Ability

To simulate the speaking ability of the LSLM, we utilize an autoregressive token-based TTS model. Unlike VALL-E-styled models that combine autoregressive (AR) and non-autoregressive (NAR) approaches with multi-layer residual vector quantization (RVQ) tokens, our model employs a single layer of discrete audio tokens. This design better meets the requirements for real-time interaction, as it eliminates the need to wait for the completion of AR token synthesis before performing NAR operations. Given target speech  $X^R$ , an SSL encoder  $Enc$  is utilized to obtain a continuous embedding  $R$ , which can be written as:

$$R = Enc(X^R). \quad (4)$$

To train an autoregressive TTS model based on discrete tokens, we quantize the speech embedding  $R$ , denoted by:

$$R^q = Qnt(R), \quad (5)$$

where  $Qnt$  is the discretization operation and  $R^q$  are the discrete tokens. Given the context information  $C$ , in this scenario the text content to be synthesized, the model synthesizes the corresponding speech discrete tokens autoregressively. We minimize the negative log-likelihood of the target sequence to train the decoder-only model, conditioned on the preceding tokens and the context. The loss function is defined as:

$$\mathcal{L}(\theta_S) = - \sum_{t=1}^{t_{EOS}} \log P(r_t^q | R_{1:t-1}^q, C; \theta_S), \quad (6)$$

where  $\theta_S$  are the parameters to model speaking ability,  $t_{EOS}$  represents the time step at which the end-of-sequence token is reached,  $r_t^q$  is the target discrete token at time step  $t$ ,  $R_{1:t-1}^q$  denotes the sequence of all previous tokens up to

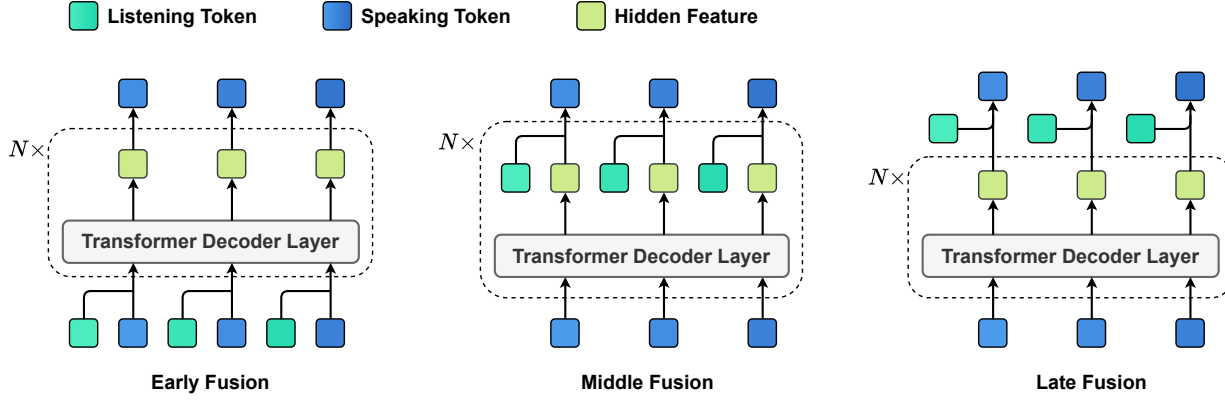


Figure 3: Different model designs to integrate the listening channel to the proposed LSLM.

time step  $t - 1$ , and  $C$  is the text content to be synthesized. During inference, the model samples  $\hat{r}_t^q$  from a conditional distribution based on the already generated tokens  $\hat{R}_{1:t-1}^q$  and the context  $C$ . The process is described by the following equation:

$$\hat{r}_t^q \sim P(r_t^q | \hat{R}_{1:t-1}^q, C; \theta_S). \quad (7)$$

A vocoder  $Dec$  is employed to recover the speech signal  $\hat{X}^R$  from discrete tokens  $\hat{R}^q$ , denoted by:

$$\hat{X}^R = Dec(\hat{R}^q, A), \quad (8)$$

where  $A$  is the acoustic prompt providing the timbre of the synthesized speech. This decoupling of timbre from content allows the AR model to focus more on semantic information rather than paralinguistic information.

## 4.2 Listening Ability

Given the audio input  $X^S$  of the listening channel, the same SSL encoder  $Enc$  in Equation 4 is used to obtain a continuous embedding  $S$ , which can be written as:

$$S = Enc(X^S), \quad (9)$$

where  $X^S$  can be a variety of sound signals, including environmental noise and human speech. Unlike training the speaking ability, which involves a discretization module, the listening channel embedding  $S$  is fed into the neural network end-to-end via a projection module  $Proj$ , which can be written as:

$$S^p = Proj(S), \quad (10)$$

where the listened audio signal is mapped to a space that can be processed by the AR model.

## 4.3 FDM Ability

LSLM has two channels: speaking and listening. At time step  $t$ , all previous information of the speaking channel  $R_{1:t-1}^q$  and the processed information of the listening channel  $S_{1:t-1}^p$  are considered by the model simultaneously. Here

we revise Equation 6 as follows:

$$\mathcal{L}(\theta_{LS}) = \begin{cases} -\sum_{t=1}^{t_{IRQ}} \log P(r_t^q | R_{1:t-1}^q, S_{1:t-1}^p, C; \theta_{LS}) & \text{if turn-taking,} \\ -\sum_{t=1}^{t_{EOS}} \log P(r_t^q | R_{1:t-1}^q, S_{1:t-1}^p, C; \theta_{LS}) & \text{otherwise.} \end{cases} \quad (11)$$

where  $\theta_{LS}$  are the parameters to model the proposed LSLM with listening-while-speaking ability. In addition to the EOS token, we add an interruption token IRQ to the tokenizer vocabulary to allow the model to terminate early if turn-taking occurs. For example, if a human interrupts, the model should stop speaking within a detection interval  $\mu$  seconds after the interruption starts. During inference, the model samples  $\hat{r}_t^q$  from a conditional distribution based on the already generated tokens  $\hat{R}_{1:t-1}^q$ , the context  $C$ , and most important, real-time listened audio tokens  $S_{1:t-1}^p$ . The revised formula from Equation 8 is written as follows:

$$\hat{r}_t^q \sim P(r_t^q | \hat{R}_{1:t-1}^q, S_{1:t-1}^p, C; \theta_{LS}), \quad (12)$$

in which, an essential requirement for the SSL encoder  $Enc$  is that it is streaming. Thus, LSLM can obtain real-time audio features during inference. This is detailed further in Section 5.1.

To comprehensively explore the integration of a listening channel to the proposed LSLM, we try to fuse the listening channel and the speaking channel with early, middle, and late methods, as shown in Figure 3.

**Early Fusion** integrates the listening and speaking channels at the input embeddings before autoregressive prediction.

**Middle Fusion** merges the listening and speaking channels at each Transformer block. Specifically, in addition to the hidden states of the speaking channel and positional embeddings, the listening channel is additionally added to the input of each Transformer block.

**Late Fusion** combines the channels at the output logits before the softmax operation.

		Command-based FDM(SD)	Voice-based FDM(SI)
TTS	train	LibriTTS-train (Zen et al. 2019)	
	val	LibriTTS-dev-clean/other (Zen et al. 2019)	
	test	LibriTTS-testsetB (Du et al. 2024a)	
Interruption	train	Say_Honey	Speech Commands Dataset-train (Warden 2017)
	val		Speech Commands Dataset-dev (Warden 2017)
	test		Speech Commands Dataset-test (Warden 2017)
Noise	all	Freesound portion of MUSAN (Snyder, Chen, and Povey 2015)	

Table 1: Data details involved in training LSLM. SD means speaker dependence, while SI means speaker independence here.

## 5 Setup

### 5.1 Model Details

The backbone of the proposed LSLM employs a decoder-only Transformer architecture consisting of 12 Transformer blocks, 12 attention heads, 768 embedding dimensions, and 3072 feed-forward layer dimensions, resulting in 106M parameters. SSL encoder vq-wav2vec (Baevski, Schneider, and Auli 2020) is employed to extract audio features and further convert speech features to discrete tokens. vq-wav2vec, a fully convolutional self-supervised pre-trained model with 20 layers of 1D convolutional neural networks with 34M parameters, is naturally suitable for streaming audio feature extraction. A simple linear layer serves as the projection module to adapt the listening channel features to the AR model. A GAN-based token-to-waveform vocoder (Du et al. 2024a) is utilized to recover discrete audio tokens to speech waveform.

### 5.2 Data Details

We evaluate the proposed LSLM under two full duplex modeling (FDM) settings: command-based FDM and voice-based FDM. Table 1 summarizes the datasets and experimental settings. For the TTS datasets, we utilize the LibriTTS dataset (Zen et al. 2019) with 585 hours of speech-text pairs for training and validation. LibriTTS-testsetB (Du et al. 2024a) is adopted for testing, which contains 500 utterances sampled from the test-clean subset of LibriTTS with 37 unseen speakers. Background noise is uniformly sourced from the Freesound portion of the MUSAN dataset (Snyder, Chen, and Povey 2015), which includes high-frequency noise such as telephone ringing and sounds of the explosion, as well as low-frequency noise such as white noise and traffic noise. The model needs to distinguish the human voice from the noise, so as to avoid turning-taking with any random input signals and avoid trivial solutions. Different interruption data is constructed based on the FDM settings.

**Command-based FDM.** In this setting, LSLM can only be interrupted by specific keywords. Timbre of 22 boutique speakers from SEED-TTS (SeedSpeechTeam 2024b) is used to synthesize the command "Honey" for the command-based FDM.

**Voice-based FDM.** In this setting, LSLM can be interrupted by a variety of different words. The Speech Commands Dataset (Warden 2017) is a set of one-second audio, each containing a single spoken English word. We split the

dataset into training, validation, and test sets in an 8 : 1 : 1 ratio, resulting in 51, 088, 6, 798, and 6, 835 pieces of data, respectively. In addition, we use a speaker independence setting, which guarantees that the speakers in the test set do not appear in the training set, simulating more challenging and realistic scenarios.

### 5.3 Training and Inference Details

We train the model with TTS, interruption, and noise datasets for 20 epochs. For each sample, noise is added with a 50% probability, and interruption with a 50% probability, to the listening tokens. If a sample is selected to include an interruption, we modify the sentence to output the IRQ token  $\mu = 0.5$  seconds after the start of the interruption and then stop outputting the remaining speaking tokens. This ensures that the model can correctly handle different audio signal combinations in the listening channel. Training is conducted on 8 NVIDIA A100 Tensor Core GPUs with 80 GB memory over approximately 20 hours. The optimization strategy involves using AdamW (Loshchilov and Hutter 2019) with a max learning rate of  $5 \times 10^{-4}$  without weight decay and a batch size of 4. The learning rate scheduler involves a warm-up phase for the first 5, 000 steps, followed by a cosine decay of the learning rate. Validation is performed at the end of each epoch, and the checkpoint with the lowest loss is selected for inference. The generation process employs Top-P sampling with a top-p value of 0.99 and a temperature of 1.0.

## 6 Experiments

### 6.1 Evaluation Metrics

**TTS capability evaluation.** We evaluate whether the speech generation capability is affected by the full duplex modeling in the proposed LSLM. The word error rate (WER) comparing the generated speech to the original text is considered as the TTS capability evaluation metrics using Whisper large v3 (Radford et al. 2023).

**Interactive capability evaluation.** Interactivity capability evaluation aims to measure how well the proposed LSLM responds to real-time and unpredictable input from the listening channel. A successful turn-taking is defined as the model stopping speaking within the  $[0, 2\mu]$  interval (1 second in our setting) after the interruption begins. Based on this, we categorize the outcomes into four cases: interruption and hit (TP), interruption and miss (FN), no interruption and hit (FP), and no interruption and miss (TN). From these

Model	Listening Channel	TTS Capability	Interactive Capability		
		WER(%) ↓	Precision(%) ↑	Recall(%) ↑	F1(%) ↑
Vanilla TTS	- (Clean)	4.28	-	-	-
LSLM <sub>EF</sub>	Clean	33.56	98.00	98.20	98.10
	Noise	34.99	97.20	97.20	97.20
LSLM <sub>MF</sub>	Clean	4.05	97.80	98.19	98.00
	Noise	4.51	97.58	97.18	97.38
LSLM <sub>LF</sub>	Clean	4.37	97.99	97.80	97.89
	Noise	6.87	93.06	96.79	94.89

Table 2: Experiments results on command-based FDM. Early fusion (LSLM<sub>EF</sub>), middle fusion (LSLM<sub>MF</sub>), and late fusion (LSLM<sub>LF</sub>) are considered.

Model	Listening Channel	TTS Capability	Interactive Capability		
		WER(%) ↓	Precision(%) ↑	Recall(%) ↑	F1(%) ↑
Vanilla TTS	- (Clean)	4.28	-	-	-
LSLM	Clean	5.33	95.21	95.78	95.50
	Noise	8.50	87.69	82.77	85.15

Table 3: Experiments results on voice-based FDM. LSLM here utilizes the architecture of middle fusion.

cases, we construct a confusion matrix and calculate the Precision, Recall, and F1 score. These metrics consider both the success rate of turn-taking (Recall) and the rate of misjudgments (Precision), providing a comprehensive evaluation of the model’s interactivity capabilities.

## 6.2 Experiments Results

We conduct a series of experiments to evaluate the command-based and voice-based FDM for both TTS capability and interactive capability. For TTS capability, we use a test set consisting of 500 utterances, referred to as LibriTTS-testsetB (Du et al. 2024a), without any interruptions in the listening channel. The primary metric for this evaluation is WER. For the interactive capability evaluation, we employ a set of 1000 utterances divided into two equal parts: 500 utterances with interruptions at a random time step and 500 utterances without interruptions. Interactive capability is measured using Precision, Recall, and F1 Score.

Additionally, we test the models under two listening channel conditions: without noise, denoted as Clean, and with noise, denoted as Noise. For the baseline Vanilla TTS model, since it does not involve a listening channel, the input is inherently clean. By comparing the clean scenarios, we assess whether the intrinsic TTS capability is affected. Additionally, integrating noisy external inputs provides a better simulation of real-world scenarios.

**Command-based FDM.** For command-based FDM, we test the three architectures described in Section 4.3 to fuse the listening channel and the speaking channel, which are early fusion (LSLM<sub>EF</sub>), middle fusion (LSLM<sub>MF</sub>), and late fusion (LSLM<sub>LF</sub>). The results are shown in Table 2. For TTS capability, The baseline Vanilla TTS model without a listening channel achieves a WER of 4.28%. LSLM<sub>MF</sub> outperforms LSLM<sub>EF</sub> and LSLM<sub>LF</sub> with a WER of 4.05% in clean conditions and maintains a relatively low WER of 4.51% in noisy conditions. The TTS ability of LSLM<sub>EF</sub>

shows a notable decrease, likely due to the fusion of input embeddings, making it difficult for the model to distinguish the information of the listening and speaking channels, negatively impacting the next token prediction. For interactive capability, all three architectures perform well with an oracle clean listening channel. However, LSLM<sub>LF</sub> shows a notable drop in performance under noisy conditions, with the F1 score falling to 94.89%. Observing that the late fusion method appears to mainly affect the precision score when the listening channel is noisy, suggests that the LSLM<sub>LF</sub> model reduces the discrimination of noise and human voice, leading to misjudgments of interruptions. In summary, the middle fusion approach demonstrates superior performance in TTS capability and competitive performance in interactive capability. Therefore, LSLM<sub>MF</sub> is concluded to be the best-performing model among those tested.

**Voice-based FDM.** We utilized a more diverse set of interruption commands compared to the command-based FDM and involved unseen speakers in the testing procedures. The best configuration from the command-based FDM, the LSLM<sub>MF</sub> model, was selected to evaluate the voice-based FDM capability. The results are shown in Table 3. LSLM shows a higher WER of 5.33% in clean conditions and 8.50% in noisy conditions compared to the Vanilla TTS model, demonstrating the challenges posed by the real-world turn-taking problem. Comparing the results with the command-based FDM using the LSLM<sub>MF</sub> model, we find that the voice-based setting faces greater challenges in maintaining high performance, especially under noisy conditions with Precision at 87.69%, Recall at 82.77%, and an F1 score of 85.15%. The diverse set of interruption commands and the involvement of unseen speakers add complexity, resulting in higher error rates.

**Visualization.** To investigate the turn-taking internal mechanism of LSLM, we visualize the probability distribution of IRQ tokens at different time steps during the genera-

Model	Training Method		TTS Capability WER(%)↓	Interactive Capability		
	Speaking	Listening		Precision(%)↑	Recall(%)↑	F1(%)↑
Vanilla TTS	✗	-	4.28	-	-	-
LSLM	✗	✓	4.82	97.80	97.99	97.89
	✗	+	4.67	95.60	95.98	95.79
	✓	✓	6.64	97.89	83.60	90.18
	✓	+	4.64	97.60	98.18	97.89
	+	✓	4.46	96.43	92.54	94.44
	+	+	4.05	97.80	98.19	98.00

Table 4: Ablation study on LSLM to evaluate the impact of different training methods. ✗ means training from scratch, ✓ means load the pre-training model and fix the parameters, + means load the pre-training model and continue training. LSLM here utilizes the architecture of middle fusion.

tion process. Given that the IRQ token probability distribution varies significantly in order of magnitude across different time steps, we utilize a logarithmic scale for probability to enhance the clarity of the visualization. As illustrated in Figure 4, the probability of the IRQ token remains below  $1 \times 10^{-3}$  when the model is not interrupted. When the listening channel starts to receive the real-time turn-taking signal, LSLM senses whether it is an interruption or a noise. After a very short time, the IRQ token probability begins to increase. Shortly thereafter, this probability rises to a level where the IRQ token is sampled by the model during generation.

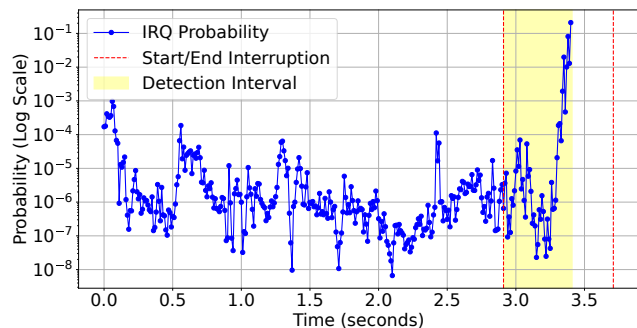


Figure 4: Illustration of the probability distribution of IRQ tokens (being interrupted) over time. The logarithmic scale probability is used for clear visualization.

### 6.3 Ablation Study

In this section, we conduct an ablation study on LSLM with middle fusion architecture to evaluate the impact of different training methods on the performance of TTS capability and interactive capability. The training methods are categorized as training from scratch (✗), loading the pre-trained model and fixing the parameters (✓), and loading the pre-trained model and continuing training (+). The detailed results are presented in Table 4.

The vanilla TTS model, trained from scratch, achieves a WER of 4.28% concerning TTS capability. For the interactive capability, the vanilla TTS model does not have a listening channel, hence no metrics are available. For the LSLM model, the best performance is observed when both the TTS

backbone and streaming SSL encoder are loaded and continue training (+ & +), achieving the lowest WER of 4.05% and highest Precision of 97.80%, Recall of 98.19%, and F1 Score of 98.00%. Some conclusions can also be drawn from these experiments. For example, the SSL encoder of the listening channel performs better when it can be continued training than fixed the parameters. One potential reason is that the SSL encoder has not encountered diverse noise during pre-training, creating a bottleneck for extracting audio with mixed human voice and noise when using fixed pre-trained parameters.

## 7 Conclusion

In this paper, we address the challenges of enhancing real-time interaction by introducing full duplex modeling (FDM) in interactive speech language models (iSLM). We introduce listen-while-speaking language model (LSLM), an innovative end-to-end model designed to handle real-time turn-taking. LSLM integrates a token-based decoder-only TTS model for speech generation and a streaming SSL encoder for audio input, enabling simultaneous listening and speaking. We propose three strategies for fusing duplex signals: early fusion, middle fusion, and late fusion. Among these, Middle Fusion demonstrates a superior balance between speech generation and real-time interaction capabilities. The proposed LSLM is evaluated in two settings: command-based FDM and voice-based FDM. Our experiments show that LSLM is robust to noisy environments and responsive to diverse instructions from unseen speakers, achieving effective duplex communication with minimal impact on system performance. Our work is an initial exploration into full duplex interactive speech language models, and there is still a long way to go to achieve smooth human-computer speech interaction. There is a lot to explore in the future, and we list some directions for reference:

1. Developing speech-in speech-out dialogue systems with full duplex modeling ability;
2. Incorporating speaker-following capability to identify interrupting speakers;
3. Exploring audio-visual co-guidance to improve turn-taking;
4. Influencing the speaking channel with real-time listened content, such as the accompaniment of improvisation.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. U23B2018 and No. 62206171), Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102, and the Science and Technology Innovation (STI) 2030-Major Projects under Grant 2022ZD0208700.

## References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Baevski, A.; Schneider, S.; and Auli, M. 2020. vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations. In *Proc. ICLR*.
- Borsos, Z.; Marinier, R.; Vincent, D.; Kharitonov, E.; Pietquin, O.; Sharifi, M.; Roblek, D.; Teboul, O.; Grangier, D.; Tagliasacchi, M.; et al. 2023. AudioLM: a language modeling approach to audio generation. *Proc. TASLP*.
- Chen, P.; Sun, S.; Shan, C.; Yang, Q.; and Xie, L. 2024a. Streaming Decoder-Only Automatic Speech Recognition with Discrete Speech Units: A Pilot Study. *Proc. Interspeech*.
- Chen, Q.; Chu, Y.; Gao, Z.; Li, Z.; Hu, K.; Zhou, X.; Xu, J.; Ma, Z.; Wang, W.; Zheng, S.; et al. 2023. LauraGPT: Listen, attend, understand, and regenerate audio with gpt. *arXiv preprint arXiv:2310.04673*.
- Chen, X.; Zhang, S.; Bai, Q.; Chen, K.; and Nakamura, S. 2024b. LLaST: Improved End-to-end Speech Translation System Leveraged by Large Language Models. *arXiv preprint arXiv:2407.15415*.
- Chen, Z.; Huang, H.; Andrusenko, A.; Hrinchuk, O.; Puvvada, K. C.; Li, J.; Ghosh, S.; Balam, J.; and Ginsburg, B. 2024c. SALM: Speech-augmented language model with in-context learning for speech recognition and translation. In *Proc. ICASSP*.
- Chen, Z.; Huang, H.; Hrinchuk, O.; Puvvada, K. C.; Koluguri, N. R.; Zelasko, P.; Balam, J.; and Ginsburg, B. 2024d. BESTOW: Efficient and Streamable Speech Language Model with the Best of Two Worlds in GPT and T5. *arXiv preprint arXiv:2406.19954*.
- Chu, Y.; Xu, J.; Yang, Q.; Wei, H.; Wei, X.; Guo, Z.; Leng, Y.; Lv, Y.; He, J.; Lin, J.; et al. 2024. Qwen2-Audio Technical Report. *arXiv preprint arXiv:2407.10759*.
- Chu, Y.; Xu, J.; Zhou, X.; Yang, Q.; Zhang, S.; Yan, Z.; Zhou, C.; and Zhou, J. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.
- Deng, K.; Sun, G.; and Woodland, P. C. 2024. Wav2Prompt: End-to-End Speech Prompt Generation and Tuning For LLM in Zero and Few-shot Learning. *arXiv preprint arXiv:2406.00522*.
- Du, C.; Guo, Y.; Shen, F.; Liu, Z.; Liang, Z.; Chen, X.; Wang, S.; Zhang, H.; and Yu, K. 2024a. UniCATS: A unified context-aware text-to-speech framework with contextual vq-diffusion and vocoding. In *Proc. AAAI*.
- Du, Z.; Chen, Q.; Zhang, S.; Hu, K.; Lu, H.; Yang, Y.; Hu, H.; Zheng, S.; Gu, Y.; Ma, Z.; et al. 2024b. CosyVoice: A Scalable Multilingual Zero-shot Text-to-speech Synthesizer based on Supervised Semantic Tokens. *arXiv preprint arXiv:2407.05407*.
- Gong, Y.; Luo, H.; Liu, A. H.; Karlinsky, L.; and Glass, J. 2024. Listen, think, and understand. *Proc. ICLR*.
- Hao, H.; Zhou, L.; Liu, S.; Li, J.; Hu, S.; Wang, R.; and Wei, F. 2023. Boosting large language model for speech synthesis: An empirical study. *arXiv preprint arXiv:2401.00246*.
- Hu, S.; Zhou, L.; Liu, S.; Chen, S.; Hao, H.; Pan, J.; Liu, X.; Li, J.; Sivasankaran, S.; Liu, L.; et al. 2024. WavLLM: Towards robust and adaptive speech large language model. *Proc. EMNLP*.
- Huang, C.-W.; Lu, H.; Gong, H.; Inaguma, H.; Kulikov, I.; Mavlyutov, R.; and Popuri, S. 2024. Investigating Decoder-only Large Language Models for Speech-to-text Translation. *Proc. Interspeech*.
- Kharitonov, E.; Lee, A.; Polyak, A.; Adi, Y.; Copet, J.; Lakhotia, K.; Nguyen, T.-A.; Rivière, M.; Mohamed, A.; Dupoux, E.; et al. 2022. Text-Free Prosody-Aware Generative Spoken Language Modeling. In *Proc. ACL*.
- Łajszczak, M.; Cámbara, G.; Li, Y.; Beyhan, F.; van Koorlaar, A.; Yang, F.; Joly, A.; Martín-Cortinas, Á.; Abbas, A.; Michalski, A.; et al. 2024. BASE TTS: Lessons from building a billion-parameter text-to-speech model on 100K hours of data. *arXiv preprint arXiv:2402.08093*.
- Lakhotia, K.; Kharitonov, E.; Hsu, W.-N.; Adi, Y.; Polyak, A.; Bolte, B.; Nguyen, T. A.; Copet, J.; Baevski, A.; Mohamed, A.; et al. 2021. On Generative Spoken Language Modeling from Raw Audio. *Proc. TACL*.
- Lian, Z.; Sun, H.; Sun, L.; Yi, J.; Liu, B.; and Tao, J. 2024. AffectGPT: Dataset and Framework for Explainable Multimodal Emotion Recognition. *arXiv preprint arXiv:2407.07653*.
- Lin, G.-T.; Chiang, C.-H.; and Lee, H.-y. 2024. Advancing large language models to capture varied speaking styles and respond properly in spoken conversations. *Proc. ACL*.
- Lin, T.-E.; Wu, Y.; Huang, F.; Si, L.; Sun, J.; and Li, Y. 2022. Duplex conversation: Towards human-like interaction in spoken dialogue systems. In *Proc. SIGKDD*.
- Loshchilov, I.; and Hutter, F. 2019. Decoupled weight decay regularization. In *Proc. ICLR*.
- Ma, Z.; Yang, G.; Yang, Y.; Gao, Z.; Wang, J.; Du, Z.; Yu, F.; Chen, Q.; Zheng, S.; Zhang, S.; et al. 2024. An Embarrassingly Simple Approach for LLM with Strong ASR Capacity. *arXiv preprint arXiv:2402.08846*.
- Neekhara, P.; Hussain, S.; Ghosh, S.; Li, J.; Valle, R.; Badlani, R.; and Ginsburg, B. 2024. Improving Robustness of LLM-based Speech Synthesis by Learning Monotonic Alignment. *Proc. Interspeech*.
- Nguyen, T. A.; Kharitonov, E.; Copet, J.; Adi, Y.; Hsu, W.-N.; Elkahky, A.; Tomasello, P.; Algayres, R.; Sagot, B.; Mohamed, A.; et al. 2023. Generative spoken dialogue language modeling. *Proc. TACL*.

- Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Proc. Neurips*.
- Pan, J.; Wu, J.; Gaur, Y.; Sivasankaran, S.; Chen, Z.; Liu, S.; and Li, J. 2023. Cosmic: Data efficient instruction-tuning for speech in-context learning. *arXiv preprint arXiv:2311.02248*.
- Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; and Sutskever, I. 2023. Robust speech recognition via large-scale weak supervision. In *Proc. ICML*.
- Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training. *San Francisco, CA, USA*.
- SeedSpeechTeam. 2024a. Seed-ASR: Understanding Diverse Speech and Contexts with LLM-based Speech Recognition. *arXiv preprint arXiv:2407.04675*.
- SeedSpeechTeam. 2024b. Seed-TTS: A Family of High-Quality Versatile Speech Generation Models. *arXiv preprint arXiv:2406.02430*.
- Seide, F.; Doulaty, M.; Shi, Y.; Gaur, Y.; Jia, J.; and Wu, C. 2024. Speech ReaLLM—Real-time Streaming Speech Recognition with Multimodal LLMs by Teaching the Flow of Time. *Proc. Interspeech*.
- Snyder, D.; Chen, G.; and Povey, D. 2015. MUSAN: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*.
- Tang, C.; Yu, W.; Sun, G.; Chen, X.; Tan, T.; Li, W.; Lu, L.; Ma, Z.; and Zhang, C. 2024. SALMONN: Towards generic hearing abilities for large language models. In *Proc. ICLR*.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Tsunoo, E.; Futami, H.; Kashiwagi, Y.; Arora, S.; and Watanabe, S. 2024. Decoder-only Architecture for Streaming End-to-end Speech Recognition. *Proc. Interspeech*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Proc. Neurips*.
- Wang, C.; Chen, S.; Wu, Y.; Zhang, Z.; Zhou, L.; Liu, S.; Chen, Z.; Liu, Y.; Wang, H.; Li, J.; et al. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.
- Wang, P.; Lu, S.; Tang, Y.; Yan, S.; Xiong, Y.; and Xia, W. 2024a. A Full-duplex Speech Dialogue Scheme Based On Large Language Models. *Proc. NeurIPS*.
- Wang, T.; Zhou, L.; Zhang, Z.; Wu, Y.; Liu, S.; Gaur, Y.; Chen, Z.; Li, J.; and Wei, F. 2024b. VioLA: Unified codec language models for speech recognition, synthesis, and translation. *Proc. TASLP*.
- Warden, P. 2017. Speech commands: A public dataset for single-word speech recognition.
- Xu, Y.; Chen, H.; Yu, J.; Huang, Q.; Wu, Z.; Zhang, S.-X.; Li, G.; Luo, Y.; and Gu, R. 2024. Secap: Speech emotion captioning with large language model. In *Proc. AAAI*.
- Yang, G.; Ma, Z.; Yu, F.; Gao, Z.; Zhang, S.; and Chen, X. 2024. MaLa-ASR: Multimedia-Assisted LLM-Based ASR. *Proc. Interspeech*.
- Yu, W.; Tang, C.; Sun, G.; Chen, X.; Tan, T.; Li, W.; Lu, L.; Ma, Z.; and Zhang, C. 2024. Connecting speech encoder and large language model for ASR. In *Proc. ICASSP*.
- Zen, H.; Dang, V.; Clark, R.; Zhang, Y.; Weiss, R. J.; Jia, Y.; Chen, Z.; and Wu, Y. 2019. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. *Proc. Interspeech*.
- Zhang, D.; Li, S.; Zhang, X.; Zhan, J.; Wang, P.; Zhou, Y.; and Qiu, X. 2023. SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities. In *Proc. EMNLP*.
- Zhang, X.; Chen, Y.; Hu, S.; Han, X.; Xu, Z.; Xu, Y.; Zhao, W.; Sun, M.; and Liu, Z. 2024. Beyond the Turn-Based Game: Enabling Real-Time Conversations with Duplex Models. *Proc. EMNLP*.