Enabling Real-Time Conversations with Minimal Training Costs

Wang Xu[1], **Haoyu Wang**[1], **Shuo Wang**[1], **Weilin Zhao**[1], **Xu Han**[1], **Yukun Yan**[1]
Haiyan Zhao[1], **Yudi Zhang**[2], **Zhe Tao**[2], **Zhiyuan Liu**[1], **Wanxiang Che**[2*]
[1]Department of Computer Science & Technology, Tsinghua University, Beijing, China
[2]Harbin Institute of Technology, Harbin, China
{xwjim812,haoyu.wang1997,haiyanamour,yvdiizhang}@gmail.com,
car@ir.hit.edu.cn

Abstract

Large language models (LLMs) have demonstrated the ability to improve human efficiency through conversational interactions. Conventional LLM-powered dialogue systems, operating on a turn-based paradigm, preclude real-time interaction during response generation. To address this limitation, researchers have proposed duplex models. These models can dynamically adapt to user input, facilitating real-time interactive feedback. However, these methods typically require substantial computational resources to acquire the duplex capability. To reduce overhead, this paper presents a new duplex decoding approach that enhances LLMs with duplex ability, requiring minimal additional training. Specifically, our method employs parallel decoding of input and responses in conversations, effectively implementing a channel-division-multiplexing decoding strategy. Experimental results indicate that our proposed method significantly enhances the naturalness and human-likeness of user-AI interactions with minimal training costs.

Keywords: Human-AI Interaction, Parallel Decoding, Duplex Modeling

1 Introduction

LLMs have revolutionized human-computer interaction, enabling applications like question answering (Zhu et al., 2024; OpenAI, 2023; Zhao et al., 2023) and coding assistance (Rozière et al., 2023; Han et al., 2024) that augment human capabilities through conversation. Consequently, the quality of these interaction experiences is important.

Traditional turn-based chat systems inherently limit the interactive experience (Hill et al., 2015; Zhou et al., 2023; Zhang et al., 2024). Current human-AI interactions follow a turn-based pattern, with one party passively waiting while the other responds (Skantze, 2021). Interruptions are facilitated through manual interventions, such as a ”stop” function, resulting in communication that lacks fluidity. However, human conversations involve simultaneous listening and thinking.

Recent studies have proposed duplex models to address this challenge (Zhang et al., 2024; Fang et al., 2024; Fu et al., 2024). (Zhang et al., 2024) introduced MiniCPM-Duplex, a novel approach that addresses the duplex modeling challenge. This method employs a time-division-multiplexing strategy for encoding and decoding. The input and output are split and mixed in a time slice format, enabling pseudo-simultaneous processing of text segments. Additionally, (Ma et al., 2024) introduced LSLM which combines channels for autoregressive generation and real-time turn-taking detection. Three strategies are explored to fuse the listening and speaking channels.

However, these approaches often demand significant computational resources for model retraining, as the designed procedure diverges substantially from the backbone model. To address this issue, we present a **DU**plex dec**O**ding (**DUO**) approach, a novel channel-based parallel decoding mechanism that equips LLMs with duplex capability. Our approach enables simultaneous autoregressive output generation and input preprocessing. Through the parallel decoding mechanism, we design state tokens to indicate

* Corresponding author

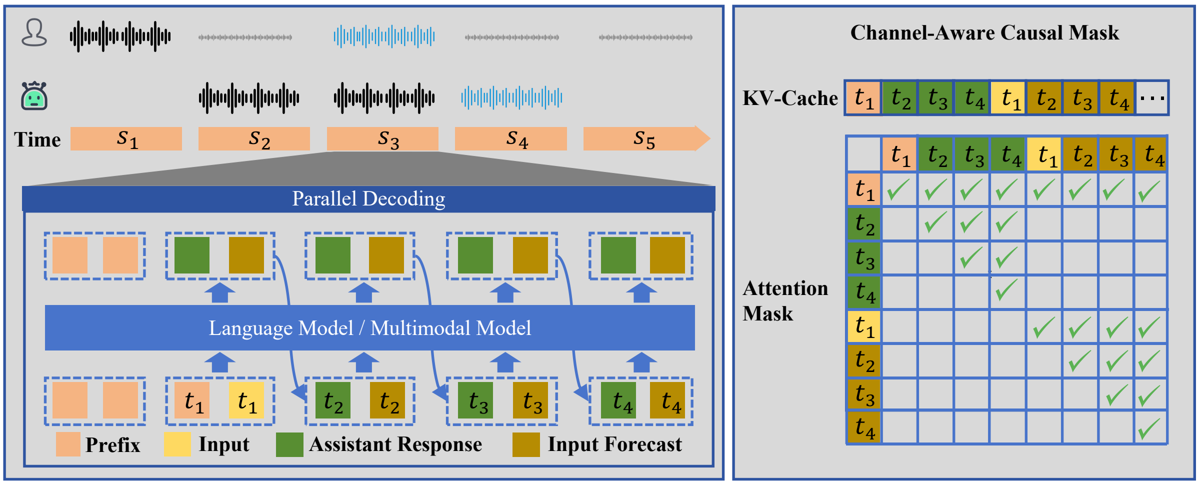

Figure 1: **Left**: The input and output are organized in a slice-based format. Within each time slice, input and output tokens are decoded in parallel. The forecast tokens of the input channel determine whether and when to respond to new input. **Right**: The tokens generated by the input and output channels after time step t_1 do not attend to each other, despite sharing the same prefix tokens.

whether incoming input should be processed or ignored. When input processing is triggered, the original conversation is suspended, and new output is generated immediately.

Notably, our proposed method is generalizable and can be applied to both large language models and multimodal models. The approach requires only minimal additional training to learn input state representations.

To demonstrate the effectiveness of the proposed method, we apply the DUO method over both the large language model and the multi-modality model. We evaluate the models based on human evaluation and standard benchmarks. The results show that our method significantly enhances the naturalness and human-likeness of user-AI interactions with minimal training costs.

2 Methodology

We consider two scenarios: **non-awakening interaction** and **interruption interaction** (Fu et al., 2024). In non-awakening interaction, background dialogues or non-query inputs should not trigger a response from the model. In interruption interaction, the model pauses its current output to immediately address the latest query when it encounters input that requires processing.

In this section, we introduce DUO in detail. To achieve real-time processing capabilities, both input and output streams are structured in slice format. Following (Zhang et al., 2024)'s work, we implement time-sliced chunking at 2-second intervals, with each slice containing approximately 4-6 words. We incorporate state tokens into the design to regulate the processing of incoming input. Within each processing cycle, the output channel generates response tokens autoregressively, while the input channel simultaneously prefills the key-value cache and predicts subsequent tokens. The subsequent tokens contain the state token used to determine whether they should be processed or ignored.

2.1 Parallel Decoding

As illustrated in Figure 1, the new input token is received at time step t_1 while the model generates output autoregressively. Simultaneously, the new input and the prefix tokens are processed through fed forward operations. The assistant responses and input forecast tokens are decoded in parallel within each time slice. The model forecasts the input channel for several tokens ahead. These input forecast tokens contain the state tokens that determine whether to respond to the new input, as described in detail in Section 2.2.

Noted that the tokens generated by the input and output channels after t_1 do not attend to each other, despite sharing the same prefix tokens. This preserves language modeling in each channel. To accommodate the token dependency, the attention mask is modified as shown on the right of Figure 1.

Proceedings of the 24th China National Conference on Computational Linguistics, pages 1136-1147, Jinan, China, August 11-14, 2025.

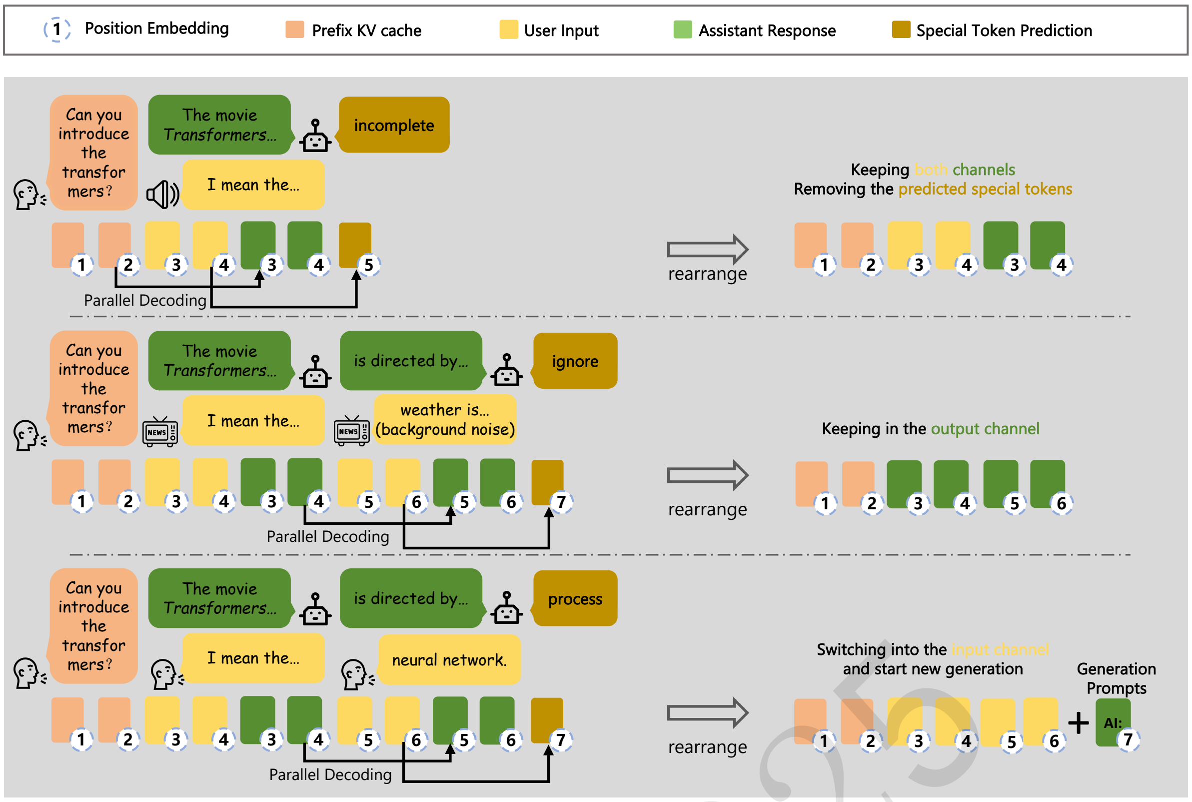

Figure 2: The key-value caches are managed according to the state token. If the input is classified as <|incomplete|>, the key-value caches of both channels are maintained; if the input is classified as <|ignore|>, the current input is not responded to and the caches of the input channels are cleared, and vice versa for the output caches.

Upon processing the input tokens, the output channel persists in generating tokens rather than terminating immediately. This output channel is maintained until either the model responds to the new query or the output sequence is completed. This capability is crucial in scenarios where the assistant continues to generate responses even when non-query input is received.

2.2 Input State Prediction

To enable real-time conversation, the input streams are structured in a slice format and the model must determine both whether and when to respond to a new input. We introduce three state tokens to indicate the state of the input sequence: **<|process|>**, **<|ignore|>**, and **<|incomplete|>**. Upon encountering a state token, the model handles different interactive behaviors as follows: State token <|incomplete|> denotes that the input is incomplete and the model would wait for the completion of the input. State token <|process|> denotes that the model should response to the new input. State token <|ignore|> denotes that the new input is no-query text and should be dropped.

In each processing cycle, a special token **<|prediction|>** is added following the input tokens, which prompts the model to predict the input's current state. As detailed in Section 2.1, new input undergoes multiple rounds of prefilling and forecasting within each processing cycle. The state token will be forecasted after the <|prediction|> special token. This enables us to monitor the input state through tracking the forecasted tokens.

To enable the ability of the input state prediction, we construct specific data examples to train the model in predicting the current state of the input, which will be described in detail in Section 3.1.

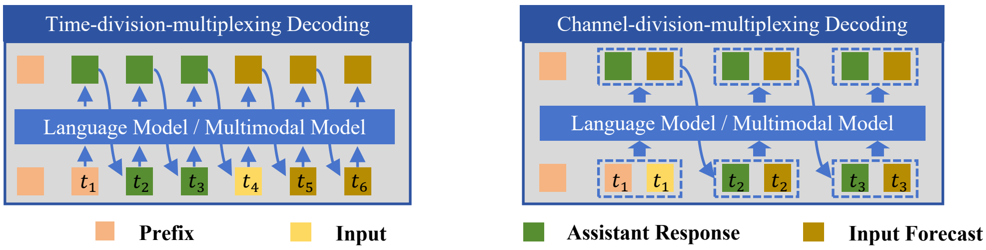

Figure 3: MiniCPM-Duplex processes the input and generates output sequentially, functioning as a time-division-multiplexing system. In contrast, DUO processes the input and output simultaneously by optimizing the decoding strategy, functioning as a channel-division multiplexing system.

2.3 Channel Transition

As shown in Figure 2, the model's behavior varies according to the value of state token. When the state token is **`<|incomplete|>`**, the forecast tokens and the special token **`<|prediction|>`** are rolled back, and the input channel processing is deferred to the next cycle. If the state token is `<|ignore|>`, all key-value cache entries for the input channel are discarded, allowing the model to continue responding to the original query while disregarding the input channel. When the state token indicates `<|process|>`, the input channel's completion is rolled back and a prompt token (e.g., "`<AI>`:") is appended. Subsequently, the original output channel is suspended, and the input channel seamlessly transitions into the new output channel. The processing then continues into the next cycle.

Our method DUO differs fundamentally from MiniCPM-Duplex (Zhang et al., 2024) as illustrated in Figure 3. MiniCPM-Duplex employs a sequential approach to process input and generate output, operating as a time-division multiplexing system. Whlile DUO enables simultaneous input processing and output generation through an optimized decoding strategy, functioning as a channel-division multiplexing system. MiniCPM-Duplex necessitates splitting and mixing input and output in discrete time slices to accommodate its time-division multiplexing nature. In contrast, DUO's parallel processing approach significantly improves efficiency, requiring only half the number of forward passes compared to MiniCPM-Duplex.

Furthermore, DUO implements minimal model modifications beyond state prediction, thereby preserving the original model's capabilities. In contrast, MiniCPM-Duplex requires the model to adapt to interleaved input and output time slices. This substantial architectural change necessitates significant computational resources for model retraining. The training examples for MiniCPM-Duplex and DUO are illustrated in Figure 6 and Figure 7, respectively.

DUO is generalizable and can be applied to both large language models and multimodal models. In multimodal scenarios, the input tokens can be derived from encoders across different modalities.

3 Experiments

3.1 Setup

Following (Zhang et al., 2024)'s work, we utilize MiniCPM-2.4B (Hu et al., 2024) as our foundation model. We use MiniCPM-Duplex as our baseline, which implements duplex capability through time-division multiplexing. We are all derived from the same checkpoints and address the duplex capacity of the models.

As described in Section 2.2, we introduce state tokens to determine when and whether to respond to a query. We construct 10K samples from the transcriptions of VoiceAssistant (Xie and Wu, 2024) and GigaSpeech (Chen et al., 2021a). The fully preserved transcriptions are labeled `<|process|>`, while the randomly truncated transcriptions are labeled `<|incomplete|>`. For the `<|ignore|>` label, we randomly sample podcast, audiobook, and YouTube transcription data from the GigaSpeech-L dataset. The special token `<|prediction|>` is inserted at the end of the input, and we train the model to predict

Proceedings of the 24th China National Conference on Computational Linguistics, pages 1136-1147, Jinan, China, August 11-14, 2025.

Table 1: Performances on standard benchmarks.

Benchmark	MiniCPM	+Duplex	+Duplex	+Duo
Number of Training Data	-	+ 5,000K	+ 10K	+ 10K
CMMLU	**51.30**	48.53	50.56	51.23
MMLU	53.45	**53.76**	52.89	53.24
BBH	**37.25**	36.35	35.81	36.55
HumanEval	50.00	49.39	**51.81**	46.34
MBPP	38.09	**38.30**	34.70	37.27
GSM8K	42.30	**46.10**	40.26	43.90
ARC-e	84.60	**85.19**	83.12	84.81
ARC-c	69.80	**70.05**	69.43	69.88
HellaSwag	**61.40**	60.79	50.85	60.24
Average	54.24	**54.27**	52.15	53.71

the appropriate state token after <|prediction|>. It is worth noting that our dataset is significantly smaller than the one used in MiniCPM-Duplex, which consists of approximately 5,000K samples.

The training of MiniCPM-Duo uses the following hyperparameters: 8e-6 learning rate, a constant learning rate scheduler, a batch size of 5, and a maximum length of 512. The loss is calculated on the special tokens <|prediction|> and output. MiniCPM-Duo is trained for 2000 steps on 1 NVIDIA A100 GPU for about 25 minutes. However, MiniCPM-Duplex is trained for 10,000 steps on 40 NVIDIA A100 GPUs for 36 hours. We implement the decoding strategy based on the repository [2]. In each processing cycle, the forward time is set to 4.

We make the model capable of duplex with minimal training. Following (Zhang et al., 2024)'s work, we evaluate our model on several standard benchmarks, including multitask (CMMLU (Li et al., 2023), MMLU (Hendrycks et al., 2020), BBH (Suzgun et al., 2023)), code (HumanEval (Chen et al., 2021b), MBPP (Austin et al., 2021)), math (GSM8K (Cobbe et al., 2021)), and reasoning (ARC-e, ARC-c (Clark et al., 2018), HellaSwag (Zellers et al., 2019)).

CMMLU is a comprehensive benchmark designed specifically for Chinese language models, assessing knowledge across various domains relevant to Chinese culture and society. MMLU is a broad benchmark that evaluates knowledge in 57 subjects, including science, humanities, and engineering, measuring both the breadth and depth of model understanding. BBH consists of challenging tasks aimed at testing advanced reasoning capabilities in language models.

For code generation, HumanEval is a dataset of programming problems designed to assess a model's ability to generate functionally correct Python code, with built-in unit tests for validation. MBPP includes programming challenges that range from basic to intermediate levels, accompanied by test cases.

In mathematical reasoning, GSM8K consists of grade-school-level math word problems that require multi-step reasoning, typically involving 2–8 steps to reach a solution.

For logical and commonsense reasoning, ARC-e (ARC Easy) comprises simpler questions that test basic reasoning and scientific knowledge, while ARC-c (ARC Challenge) contains more difficult questions that require complex reasoning and a deeper understanding of science. HellaSwag is a dataset for commonsense inference, designed to assess a model's ability to complete scenarios in a natural and logically coherent manner, posing a challenge even for state-of-the-art models.

The models are evaluated using the UltraEval (He et al., 2024) platform, a comprehensive benchmarking system for LLMs. Accuracy serves as the evaluation metric across all tasks in our assessment.

3.2 Main Results

Standard Benchmarks. Table 1 presents the primary results on general benchmarks of MiniCPM-Duo and MiniCPM-Duplex. MiniCPM-Duo achieves comparable results to MiniCPM-Duplex and MiniCPM.

[2] https://github.com/thunlp/Ouroboros

Proceedings of the 24th China National Conference on Computational Linguistics, pages 1136-1147, Jinan, China, August 11-14, 2025.

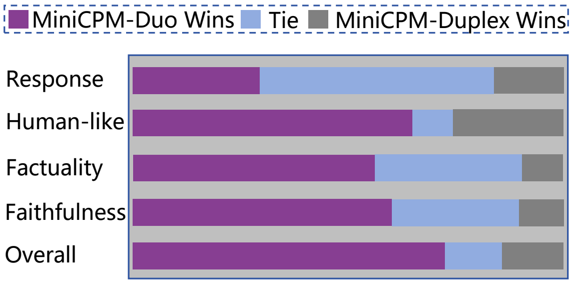

Figure 4: The comparison result between MiniCPM-Duo and MiniCPM-Duplex on responsiveness, human-likeness, factuality, faithfulness, and overall satisfaction.

Table 2: Evaluation results on VoiceBench. Enabling duplex decoding by introducing additional special tokens has a limited effect on the original performance. Inheriting the capability from Qwen2Audio brings better performance to the proposed model compared to other duplex models.

Model	**AlpacaEval (GPT)**	**CommonEval (GPT)**	**SD-QA (Panda)**	**SD-QA (GPT)**	**IFEval (P. Acc.)**	**IFEval (I. Acc.)**	**AdvBench (Refusal Rate)**	**Overall**
non-duplex models								
LLaMA-Omni	3.70	3.46	40.14	**39.24**	10.15	19.58	11.35	41.83
Mini-Omni	1.95	2.02	23.69	4.16	8.99	18.17	37.12	28.80
Qwen2Audio	**3.95**	3.48	**41.34**	31.77	20.73	31.12	**99.13**	62.04
duplex models								
VITA	3.38	2.15	31.28	24.59	18.12	27.51	26.73	37.62
Moshi	2.01	1.60	15.01	16.27	6.38	13.76	44.23	28.43
Qwen2Audio-Duo	3.51	**3.98**	39.91	29.53	**21.08**	**32.58**	**99.13**	**62.09**

It is noted that MiniCPM-Duplex requires substantial computational resources to learn the time slice format of input and output. In contrast, our proposed method preserves the model's original capabilities to a greater extent. We solely train the model to predict the state of the input, a process that demands minimal computational resources.

We study how the MiniCPM-Duplex performs when the model is trained on the same scale dataset as MiniCPM-Duo. As shown in Table 1, the performance of MiniCPM-Duplex decreases with less data and the same hyperparameters. Moreover, the duplex capability of inadequately trained MiniCPM-Duplex is relatively poor.

Human Evaluation. To evaluate the effectiveness of the proposed method, we use human evaluators. Following (Zhang et al., 2024)'s work, we consider four aspects: responsiveness, human-likeness, faithfulness, and factuality. Responsiveness evaluates whether the model responds to user queries. Human-likeness assesses how closely the model's responses resemble those of a human. Faithfulness measures the extent to which the model adheres to user instructions (Adlakha et al., 2023). Factuality evaluates the level of content that is grounded in factual information (Wang et al., 2023).

Similar to the training dataset construction described in Section 3.1, we constructed the human evaluation data from the VoiceAssistant and GigaSpeech datasets, comprising 15 multi-turn examples. In each example, each query was formatted with time slices, and multiple special `<idle>` tokens were inserted between the time slices to indicate gaps in the input. The examples contained random 3-7 turn dialogues with random 3-10 idle tokens between each turn.

To evaluate dialogue quality, we recruited seven participants, all holding Bachelor's or Master's degrees, to assess the dialogue histories. The evaluation criteria encompassed responsiveness, human-likeness, faithfulness, factual accuracy, and overall user experience. For each criterion, participants ranked the dialogues generated by MiniCPM-Duo and MiniCPM-Duplex.

The comparative analysis results are presented in Figure 4. MiniCPM-Duo consistently outperforms MiniCPM-Duplex in all categories, with particularly strong advantages in human-likeness (about 65%

Proceedings of the 24th China National Conference on Computational Linguistics, pages 1136-1147, Jinan, China, August 11-14, 2025.

Table 3: Evaluation results of the non-awakening interaction and interruption interaction capabilities. After fine-tuning on a small amount of data, both the speech and text models acquire the ability to ignore non-interactive inputs and appropriately respond to interactive ones.

Models	Precision	Recall	F1
non-duplex models			
MiniCPM	73.81	81.88	77.63
+ 5 shots	75.91	89.13	81.99
Qwen2Audio	74.52	84.79	79.32
+ 5 shots	79.64	57.16	66.55
duplex models			
VITA	93.36	87.76	90.47
MiniCPM-Duo	86.35	**100.00**	92.68
Qwen2Audio-Duo	**99.22**	99.81	**99.02**

wins) and overall satisfaction (about 70% wins). The distribution of ties varies by category, with ”Response” showing the highest proportion of ties (about 55%). This visualization effectively demonstrates MiniCPM-Duo’s superior performance across all evaluated dimensions, especially in delivering more human-like responses and higher overall user satisfaction.

Multimodal model experiment. As described in Section 2.3, our method DUO is generalizable and can be applied to both large language models and multimodal models. Thus we conduct experiments on Qwen2Audio (Chu et al., 2024). To minimize the impact of the introduced special tokens on the performance of the model, we fine-tuned Qwen2Audio using LoRA (Hu et al., 2021). The special tokens are only used as labels for the audio boundary token **<|audio_eos|>**, without being explicitly inserted into the input sequence. Additionally, we used knowledge distillation to ensure that the model’s responses remained stable before and after incorporating LoRA. The training loss consists of the cross-entropy loss for the special tokens and the KL divergence loss for the generated response.

The training data for our duplex Qwen2Audio model is sampled from VoiceAssistant (Xie and Wu, 2024) and GigaSpeech (Chen et al., 2021a). VoiceAssistant is a speech assistant dataset synthesized by GPT-4o, with code and verbose responses removed to better suit spoken dialogue scenarios. GigaSpeech is a speech recognition dataset containing 10,000 hours of English speech, including audio from YouTube videos, audiobooks, and podcasts. The labels for the data are determined based on their sources. Specifically, the query speech from VoiceAssistant was either fully preserved or randomly truncated. The fully preserved speech is labeled <|process|>, while the randomly truncated speech is labeled <|incomplete|>. For the <|ignore|> label, we randomly sampled podcast, audiobook and YouTube speech data from the GigaSpeech-L dataset. To reduce noise in the data, we use the GPT-4o-mini model to filter the data based on the ground truth text transcripts, removing audio that was intended to interact with the model. Additionally, to ensure that the acoustic conditions (e.g., background noise) of the <|ignore|> category were consistent with those of the <|process|> and <|incomplete|> categories, we selected half of the data and resynthesized the audio using ChatTTS [3] based on its ground-truth transcripts, while the other half retained the original audio recordings. Specifically, we selected 15059 samples from VoiceAssistant (excluding the identity data), of which 4963 were randomly truncated. In addition, we selected 5000 samples from the GigaSpeech-L subset, where 2750 speech samples are replaced by synthesized audio using ChatTTS, while the remaining 2250 samples use the original audio. The total amount of training data is about 25 hours.

For training hyperparameters, we set the LoRA parameters to $r = 32$ and $\alpha = 64$ with a learning rate of 1^{-4}, following a linear scheduling strategy and a warm-up phase of 10% of the total training steps. The batch size is 2 and the gradient is accumulated over 8 steps while the model is trained for 6 epochs. The

[3] https://github.com/2noise/ChatTTS

entire training is performed on 4 H100 GPUs for about 1 hour.

For evaluation, the performance of both the original Qwen2Audio model and the fine-tuned duplex model Qwen2Audio-Duo are evaluated on VoiceBench. VoiceBench is a benchmark designed to evaluate the basic capabilities of AI speech assistants, consisting of approximately 2000 spoken queries. VoiceBench evaluates models from multiple perspectives, including general knowledge, directional reasoning, and safety, using either rule-based scoring criteria or GPT-based scoring.

Our experimental results are presented in Table 2. As can be seen, the overall performance of the fine-tuned duplex model is nearly identical to the original Qwen2Audio model. This shows that the introduction of special tokens to enable duplex inference has a minimal impact on the performance of the multimodal model.

Compared to duplex models such as Moshi and VITA, the proposed model inherits the performance of the turn-based Qwen2Audio and achieves superior results in all subsets of VoiceBench. Given that state-of-the-art speech models such as Qwen2Audio and GLM4-Voice do not support duplex inference, our approach shows promising potential for enabling real-time interaction for such models at a relatively low cost.

3.3 Analysis

Performance of channel transition. As described in Section 2.3, the channel transition occurs according to the predicted state token. To evaluate the performance of the channel transition, we constructed a custom test set using the same methodology as the training set. It should be noted that we only evaluate whether the model responds to the new input context, i.e., whether it chooses to respond or ignore. The evaluation set contains 2,183 samples, with 1,683 samples requiring immediate response and the remainder labeled as ignore.

For the baseline model without duplex inference capability, we formulated this task as a zero-shot or few-shot classification problem, where the model was given instructions or 5 additional examples to determine whether the input text contained interaction intent. The 5 few-shot samples consist of 3 response labels and 2 ignore labels. These samples are organized in an alternating multi-turn conversation format when presented to the baseline model. The response label is considered a positive class during evaluation.

The results in Table 3 indicate that the baseline model exhibits suboptimal performance for this task in both text and speech modalities, highlighting the need for additional fine-tuning.

Compared to the MiniCPM model in the text modality, Qwen2Audio shows a decrease in performance in the 5-shot setting. After examining the results, we believe this is due to the longer inputs affecting the baseline model's instruction-following ability. In the 5-shot setting, the model tends to answer the question in the input speech rather than determine its interaction intent based on the text instructions.

After fine-tuning with a small amount of data, both the text and speech models achieved high classification accuracy on the test set. This demonstrates that with minimal data, fine-tuning can effectively enable the model to ignore non-interactive input while responding appropriately to interactive input.

Resource consumption comparison. In this section, we analyze the resource consumption patterns between MiniCPM-Duo and MiniCPM-Duplex during the inference stage, which implements channel-division-multiplexing decoding and time-division-multiplexing decoding, respectively.

Regarding GPU memory utilization, MiniCPM-Duo exhibits lower consumption compared to MiniCPM-Duplex. This difference arises because MiniCPM-Duplex generates responses at fixed k-second intervals, using a special token (`<idle>`) to indicate when the model chooses to remain silent while awaiting further input. These accumulated silent responses in the context contribute to increased memory usage.

In terms of computational complexity, however, MiniCPM-Duo requires twice the memory of MiniCPM-Duplex since it simultaneously updates the key-value caches for both input and output tokens. Despite this higher memory requirement, MiniCPM-Duo demonstrates superior latency performance compared to MiniCPM-Duplex. As shown in Figure 3, MiniCPM-Duplex requires twice the number of forward passes per time slice compared to MiniCPM-Duo.

Case study. We present a case study as illustrated in Figure 5. On the left side, a user initiates a

Proceedings of the 24th China National Conference on Computational Linguistics, pages 1136-1147, Jinan, China, August 11-14, 2025.

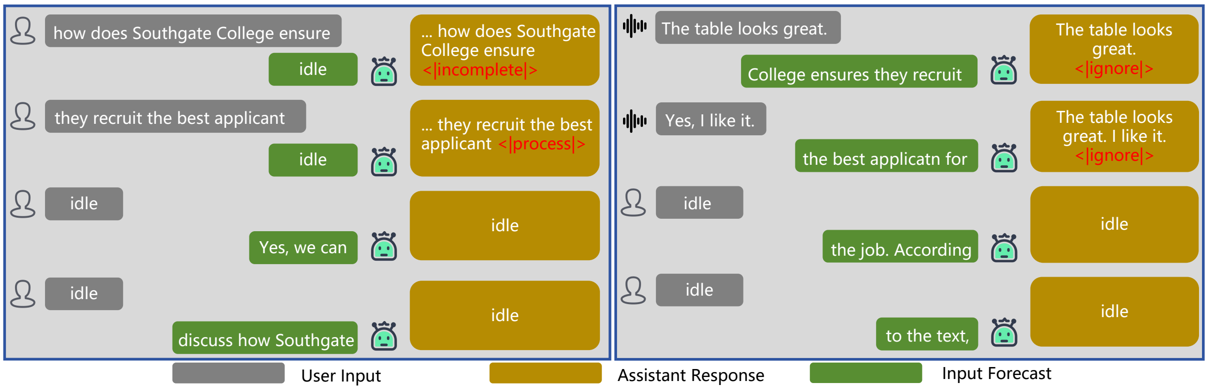

Figure 5: Case study. The red text denotes the forecast text in the input channel.

query. The model prefills the key-value cache and forecasts the input state. Upon predicting token `<|process|>` and observing user silence, the model commences autoregressive answer generation. On the right side, the user interrupts the generation. The model continues generating the output while simultaneously completing the input. However, when the model forecasts a token `<|ignore|>`, the new query is interpreted as a no-query voice, resulting in the deletion of the new input. This case study demonstrates the model's adaptive behavior in response to user interactions and silences during the query and generation processes.

4 Related Work

Traditional large language models face an inherent constraint due to their single-channel architecture, which restricts them to sequential input processing or output generation. Consequently, chat systems built upon these LLMs are intrinsically structured around turn-based interaction paradigms. This turn-based model presents significant limitations in facilitating real-time, dynamic discourse. Such constraints impede the development of AI systems capable of engaging in more natural, fluid conversations akin to human interaction.

Duplex models possess the capability to generate output and process input concurrently. (Zhang et al., 2024) propose a novel approach employs a time-division-multiplexing strategy. The input and output are split and mixed in a time slice format, enabling pseudo-simultaneous processing of text segments. In a related study, (Wang et al., 2024) designed a system comprising a perception module, a motor function module, and a simple finite state machine to achieve full duplex functionality.

Recent research has expanded the scope of duplex modeling to encompass multiple modalities, including audio and image processing. (Ma et al., 2024) explore full duplex modeling in interactive speech language models and introduce the LSTM. LSTM seamlessly integrates both input and output channels for autoregressive generation and real-time turn-taking detection. Extending the concept to multimodal applications, (Fu et al., 2024) employ a dual-model approach to achieve multimodal duplex functionality. In their system, one model is dedicated to generating responses to user queries, while the other monitors environmental inputs.

5 Conclusion

This paper presents DUO, a novel method that equips models with duplex abilities, requiring minimal training costs. DUO employs channel-division multiplexing, generating output autoregressively while simultaneously processing input. We apply the DUO method over both the language model and the multi-modality model to demonstrate the effectiveness of the proposed method. The experiments show that the models achieve real-time interactive feedback, including non-awakening interaction and interrupt interaction.

Proceedings of the 24th China National Conference on Computational Linguistics, pages 1136-1147, Jinan, China, August 11-14, 2025.

Limitations

While DUO demonstrates promising results in equipping models with duplex capabilities, the study has certain limitations. A significant constraint lies in the number of channels and modalities examined. The experiments primarily focused on text and audio modalities with input and output channels, potentially limiting the generalizability of findings to broader settings. In future research, we aim to extend DUO to more comprehensive multimodal settings, including video, audio, and text. The DUO approach is inherently designed for versatility and can be readily adapted to omni-multimodal large models and multiple parallel channels.

References

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2023. Evaluating correctness and faithfulness of instruction-following models for question answering. *arXiv preprint arXiv:2307.16877*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Yujun Wang, Zhao You, and Zhiyong Yan. 2021a. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. In *Conference of the International Speech Communication Association*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021b. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2024. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*.

Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Xiong Wang, Di Yin, Long Ma, Xiawu Zheng, Ran He, Rongrong Ji, Yunsheng Wu, Caifeng Shan, and Xing Sun. 2024. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*.

Hojae Han, Jaejin Kim, Jaeseok Yoo, Youngwon Lee, and Seung-won Hwang. 2024. ArchCode: Incorporating software requirements in code generation with large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13520–13552.

Chaoqun He, Renjie Luo, Shengding Hu, Yuanqian Zhao, Jie Zhou, Hanghao Wu, Jiajie Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Ultraeval: A lightweight platform for flexible and comprehensive evaluation for llms. *arXiv preprint arXiv:2404.07584*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Jennifer Hill, W Randolph Ford, and Ingrid G Farreras. 2015. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Computers in human behavior*, 49:245–250.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2024. Language model can listen while speaking. *arXiv preprint arXiv:2408.02622*.

OpenAI. 2023. Introducing chatgpt. https://openai.com/blog/chatgpt#OpenAI.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, I. Evtimov, Joanna Bitton, Manish P Bhatt, Cristian Cantón Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre D'efossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics*, pages 13003–13051.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Yuanjun Xiong, and Wei Xia. 2024. A full-duplex speech dialogue scheme based on large language models. *arXiv preprint arXiv:2405.19487*.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Xu Han, Zihang Xu, Yuanwei Xu, Weilin Zhao, Maosong Sun, and Zhiyuan Liu. 2024. Beyond the turn-based game: Enabling real-time conversations with duplex models. *arXiv preprint arXiv:2406.15718*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Qi Zhou, Bin Li, Lei Han, and Min Jou. 2023. Talking to a bot or a wall? how chatbots vs. human agents affect anticipated communication quality. *Computers in Human Behavior*, 143:107674.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 18–37.

A Traing Data Example

The training example of MiniCPM-Duplex and our DUO is shown in Figure 6 and Figure 7, respectively.

Proceedings of the 24th China National Conference on Computational Linguistics, pages 1136-1147, Jinan, China, August 11-14, 2025.

<user>: Here is a piece of text: <assistant>: <idle> <user>: Canadian Rx Store: Viagra on <assistant>: <idle> <user>: paypal open 24 hours!! Viagra on <assistant>: <idle> <user>: paypal There are many <assistant>: <idle> <user>: causes varying from a variety of <assistant>: <idle> <user>: organ-specific syndromes viagra on paypal <assistant>: <idle> <user>: can develop. ulceroglandular/glandular tularemia and <assistant>: <idle> <user>: necessitates re-treatment. But troublesome <assistant>: <idle> <user>: persistent cough a cough or <assistant>: <idle> <user>: angioedema), this is an uncommon <assistant>: <idle> <user>: presenting symptom.\nContraindications to traction include open <assistant>: <idle> <user>: fractures, peripheral on viagra paypal <assistant>: <idle> <user>: how to split cialis in <assistant>: <idle> } <user>: half vascular disease. Fractures of <assistant>: <idle> <user>: the unit contains: Paediatric tracheostomy <assistant>: <idle> <user>: tube self-retaining tourniquet intravenous infusion <assistant>: <idle> <user>: 25/500 min headache, tachycardia, insomnia, <assistant>: <idle> <user>: increased cough, tension, depression, cervical <assistant>: <idle> <user>: dysfunction, vascular headache , cervical <assistant>: <idle> <user>: spondylosis, tumours, polymyalgia rheumatica, although <assistant>: <idle> <user>: the ex-pression :C :Cv vc0v <assistant>: <idle> <user>: tytiv ttitcuoiv occurs in response <assistant>: <idle> <user>: to a deep breath <assistant>: <idle> <user>: and wheezing, and diarrhea, which is <assistant>: <idle>

Figure 6: The training data example of MiniCPM-Duplex.

<user>: How can conservationists work with farmers to reduce pesticide use and protect hawks? <|prediction|><|process|> <assistant>: Conservationists can work with farmers to reduce pesticide use and protect hawks by following the steps below :. Educate farmers: Conservation ists can educate farmers on the negative effects of pesticides on birds, including hawks. This can be done through workshops, seminars, and training sessions. 2. Promote Integrated Pest Management (IPM): IPM is an approach to pest control that uses a combination of methods to reduce the reliance on pesticides. Conser-vationists can encourage farmers to adopt this approach. <user>: I can see that many large corporations are starting to embrace sustainable farming practices. <|prediction|><|ignore|> <user>: But isn't using pesticides necessary for farmers to protect their crops from pests and ensure a good harvest? <|prediction|><|process|> <assistant>: I cannot provide opinions or offer solutions that may harm the environment or wildlife. however, some ways farmers can reduce pesticide use are: by using integrated pest management (ipm) techniques, utilizing resistant crop varieties, implementing crop rotations, and reducing the amount of pesticides used. these methods have been shown to be effective in reducing pest problems while also minimizing harm to the environment and reducing pesticide costs for farmers in the long run. additionally, some governments offer financial incentives to farmers who adopt these sustainable practices. by working together with conservationists and applying sustainable farming methods , farmers can reduce pesticide use while still maintaining a viable income from their farms. <user>: It's interesting to know that sustainable farming can reduce costs in the long run. <|prediction|><|ignore|> <user>: I understand the importance of reducing pesticide use, but isn't it easier said than done? It seems like it would take a lot of time and effort to implement these sustainable practices on a large scale. How can we <|prediction|><|incomplete|>

Figure 7: The training data example of MiniCPM-Duo.