# DUPLEXOMNI: REAL-TIME LISTENING, SEEING, THINKING, AND SPEAKING FOR FULL-DUPLEX INTERACTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Human interaction is continuous, multimodal, and full-duplex by nature. Although recent omni models have made substantial progress in unified speech, vision, and text modeling, combining seamless real-time interaction with complex reasoning and tool use remains challenging. We present DuplexOmni, a method for real-time multimodal full-duplex interaction. DuplexOmni separates model capability into an interaction layer and a thinking layer, which collaborate asynchronously in parallel. The interaction layer is implemented by the DuplexOmni model, an end-to-end system that processes streaming audio and video inputs while generating text and speech responses in real time. The thinking layer is a pluggable module that provides complex reasoning and tool-use capabilities. To support this method, we further develop a Writer-Director pipeline for constructing continuous-interaction training data. Experiments show that DuplexOmni achieves strong full-duplex interaction performance on public benchmarks while maintaining low response latency.

## 1 INTRODUCTION

Human interaction is naturally seamless, multimodal, and full-duplex. Even while speaking, people continue to listen to their partner and watch their surroundings, so they take in new input throughout their own turn. New information may change subsequent expression or lead to interaction events such as pauses. Therefore, this kind of interaction requires a model to express in real time during continuous perception and to keep updating its understanding during expression.

Recent omni models (OpenAI, 2024; Xu et al., 2025; Yao et al., 2024; Reid et al., 2024) have advanced unified modeling of speech, vision, and text. These models have demonstrated real-time multimodal understanding and generation. However, existing models still struggle to provide seamless user interaction. As shown in Figure 1, when interaction involves deep thinking or tool use, the model often pauses the ongoing dialogue. It then waits for reasoning or tool results before continuing its response. This causes clear interruptions. Some methods build lightweight real-time models by reducing model size and simplifying reasoning. However, this weakens or even removes the ability to handle complex tasks. The root cause is that

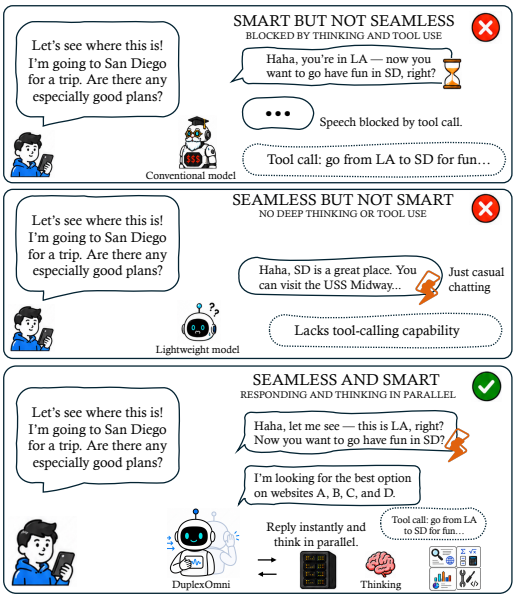


Figure 1: DuplexOmni keeps real-time interaction continuous by decoupling instant responses from asynchronous thinking and tool use.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

listening, speaking, reasoning, and acting are usually handled within a single serial pipeline. Reasoning and tool use can interrupt real-time interaction, while shortening the reasoning process limits the model's capabilities. Seamless multimodal interaction therefore requires independent execution spaces for real-time interaction and deep reasoning, as well as continuous collaboration throughout the dialogue.

To address this contradiction, we propose DuplexOmni, a method for real-time multimodal interaction that enables seeing, listening, speaking, thinking, and acting to proceed in parallel. Inspired by fast and slow thinking in human cognition, DuplexOmni divides model capabilities into a decoupled interaction layer and thinking layer, and lets them collaborate asynchronously. The interaction layer is handled by the DuplexOmni model, an end-to-end full-duplex model that continuously receives real-time speech, video, and dialogue history, while producing streaming text and speech. The thinking layer provides pluggable deep cognitive capability, and can be instantiated with large language models or tool agents according to the task. The interaction layer maintains low-latency perception and expression, and requests background thinking from the thinking layer when needed. Results returned by the thinking layer are progressively received, organized, and incorporated by the interaction layer into subsequent responses, keeping real-time interaction continuous while giving deep thinking independent computation time.

To realize DuplexOmni, we develop the model, collaboration mechanism, data construction, and experimental validation. First, we train the DuplexOmni model as the interaction layer, so that a single end-to-end model jointly handles all real-time multimodal inputs as well as its text and speech outputs. Second, we design asynchronous collaboration between the interaction layer and the thinking layer, allowing the interaction layer to invoke external thinking capability and integrate returned results during ongoing dialogue. Third, we build a Writer-Director data pipeline that converts turn-based dialogues into continuous-interaction data with explicit timing of turn-taking and thinking events. Experiments show that DuplexOmni achieves strong full-duplex interaction performance on multiple benchmarks while maintaining low response latency. The model weights and training data are publicly available, and the complete code is included in the supplementary material.

Our main contributions are as follows:

- We propose DuplexOmni, a continuous multimodal interaction method that decouples the interaction layer from the thinking layer, enabling low-latency interaction and pluggable deep thinking to collaborate asynchronously.
- We build and open-source a Writer-Director data pipeline for constructing continuous-interaction data, and use it to train the DuplexOmni model.
- We validate DuplexOmni on multiple public benchmarks, and have released the full training and inference process to support future research.

## 2 Related work

### 2.1 Omni Models

Recent years have seen substantial progress in omni models (Tongyi Fun Team et al., 2025; Chen et al., 2025b). By introducing end-to-end audio-visual understanding and generation, these models can process multiple forms of information in a way closer to human interaction. Models such as GPT-4o (OpenAI, 2024), Gemini (Gemini Team, 2023), Qwen2.5-Omni (Xu et al., 2025), Qwen3-Omni (Qwen Team, 2025), MiniCPM-o (Yao et al., 2024), Baichuan-Omni (Baichuan Inc., 2025), and Kimi-Audio (KimiTeam et al., 2025) have demonstrated strong cross-modal understanding and dialogue capabilities. The Qwen series unifies speech, vision, and text capabilities within an end-to-end omni architecture; MiniCPM-o emphasizes efficient multimodal interaction in on-device or lightweight settings. Despite significantly expanding the perceptual and generative boundaries of large models, most of these models still organize interaction in a request-driven manner: user input is collected, the model performs understanding and generation, and then a response is returned. This mode is suitable for single-turn or multi-turn task processing, but cannot handle user interaction in real time. Therefore, omni-modal models provide a foundation for unified multimodal capability, but do not directly address coordination in real-time continuous interaction.

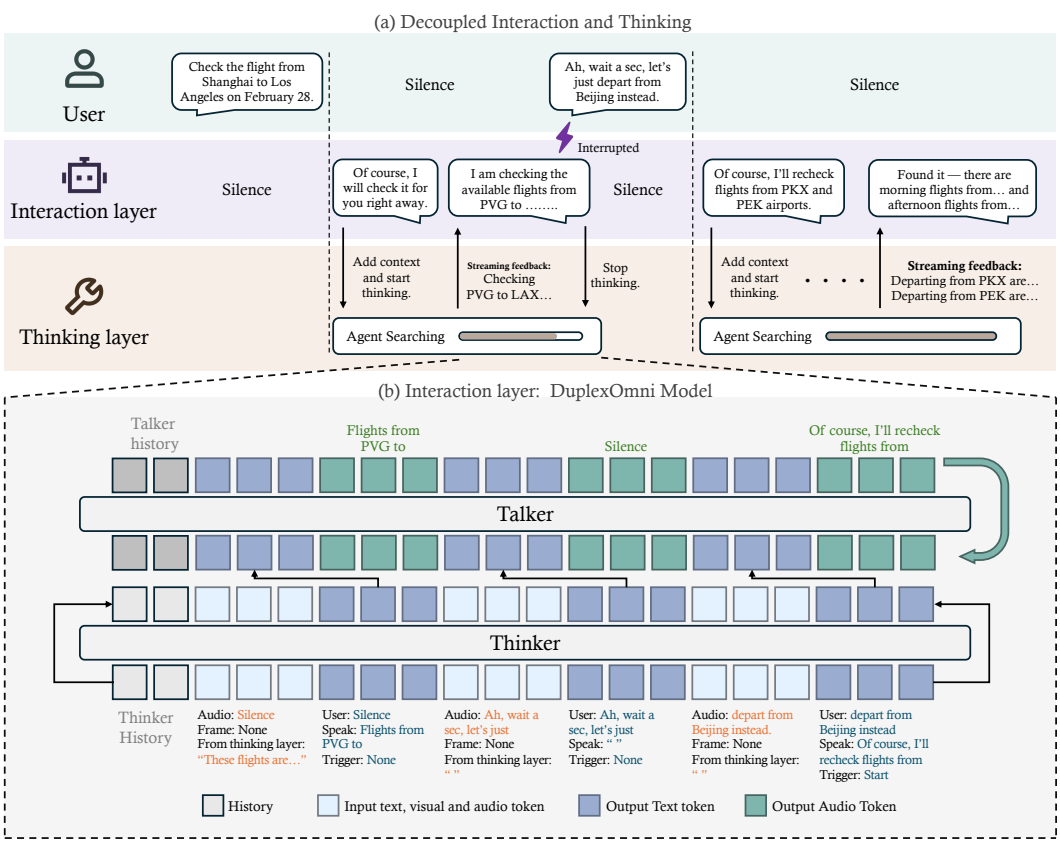


Figure 2: Overview of DuplexOmni. (a) The interaction layer conducts real-time dialogue while asynchronously collaborating with the thinking layer. (b) The DuplexOmni model implements the interaction layer with time-sliced Thinker-Talker inference over streaming inputs and thinking feedback. MTP and Code2Wav are omitted for clarity.

### 2.2 Duplex Interaction

Other studies focus on duplex interaction, where a model can receive user input and generate responses simultaneously in real time. Unlike traditional simplex models, duplex interaction (Ma et al., 2025; Nguyen et al., 2022; Wang et al., 2024; Chen et al., 2025a; Lu et al., 2025; Yang et al., 2026) requires the model to keep listening to user speech while speaking, and to handle natural interaction phenomena such as overlapping speech and interjections. Moshi (Défossez et al., 2024) achieves low-latency full-duplex speech dialogue by jointly modeling user speech, assistant text, and assistant speech. OmniFlatten (Zhang et al., 2025) flattens speech and text inputs and outputs into a unified GPT sequence to model seamless voice conversation in an end-to-end manner. MiniCPM-o (Yao et al., 2024) models time-aligned multimodal streams, enabling the model to see, listen, and speak on a unified timeline. Mini-Omni-Reasoner (Xie et al., 2025) and dGSLM (Nguyen et al., 2022) also advance streaming speech modeling and parallel listening and speaking (Yu et al., 2025; Zhang et al., 2023a; Held et al., 2025; Xiaomi, 2025) from different perspectives. These methods move speech interaction from strict turn-taking toward more natural continuous dialogue. However, when dialogue requires longer reasoning, interaction between the model and the user is still interrupted. Work on reasoning while listening or speaking (Shih et al., 2025; Chiang et al., 2025; Wu et al., 2025) is discussed in Appendix B.

## 3 Method

We propose DuplexOmni, a method for multimodal full-duplex interaction. As shown in Figure 2, DuplexOmni divides model capabilities into an interaction layer and a thinking layer, and connects

real-time interaction with deep thinking through asynchronous parallel collaboration. The interaction layer is implemented by the DuplexOmni model. It continuously takes in the user's audio-visual stream and the feedback from the thinking layer, and responds in real time with text and speech that reflect the current context and the inferred user intent. The thinking layer is a pluggable layer that can be instantiated with multimodal large language models (MLLMs) or tool agents to perform complex reasoning, tool use, and task planning. To train and deploy this form of interaction, we build a Writer-Director data pipeline that generates continuous-interaction data and simulates realistic dialogue scenarios for training the full-duplex interaction model. In the following sections, we detail the hierarchical structure of DuplexOmni, the Writer-Director data pipeline, and real-time inference for low-latency deployment.

### 3.1 DUPLEXOMNI

DuplexOmni adopts a two-layer structure with an interaction layer and a thinking layer. The interaction layer handles all user-facing behavior, from perceiving the audio-visual input and the returned thinking results to deciding when and how to respond, and it requests assistance from the thinking layer when needed. The thinking layer provides additional cognitive capability through a pluggable interface, running complex reasoning and tool use in the background. We first describe the decoupled structure and how the two layers collaborate, and then the DuplexOmni model that implements the interaction layer.

#### 3.1.1 DECOUPLED INTERACTION AND THINKING

Decoupled Interaction and Thinking centers on thinking requests initiated by the interaction layer and streaming results returned by the thinking layer. The interaction layer handles real-time dialogue, while the thinking layer performs background reasoning. When the current interaction requires external assistance, the interaction layer passes the user context to the thinking layer, including dialogue text, video information, and task state, and issues a thinking request. This request does not block real-time interaction. The interaction layer continues to receive user input, control the dialogue rhythm, and organize immediate responses.

After receiving a request, the thinking layer runs asynchronously with the interaction layer. It can be a strong LLM or a task-specific agent. During streaming output, the thinking layer wraps intermediate results with special control tokens and continuously returns them to the interaction layer. The interaction layer organizes these responses into text and speech replies suitable for the current context. When the interaction layer determines that conditions have changed or that the current dialogue no longer needs external assistance, it stops the streaming output of the thinking layer through special control tokens. This design enables DuplexOmni to continuously serve users in low-latency interaction while incorporating complex reasoning, tool use, or specialized agent capabilities on demand.

#### 3.1.2 DUPLEXOMNI MODEL

**Time-Sliced Full-Duplex Modeling** DuplexOmni model realizes full-duplex interaction through time-sliced autoregressive inference. Similar to Moshi (Défossez et al., 2024) and MiniCPM-o (Yao et al., 2024), we divide continuous interaction into fixed 480 ms slices. At slice $t$, the model consumes the dialogue history, the intermediate results returned by the thinking layer, and the inputs from slice $t-1$. For audio, this input is the 480 ms speech segment in that slice. For video, it is the visual frame sampled from the same slice.

Each slice produces three outputs: a thinking-control signal, an ASR transcription of the latest user input, and the Assistant's text and 480 ms speech response. This design allows the model to continue receiving new user inputs and new thinking results while generating responses, thereby achieving full-duplex interaction with a response granularity of 480 ms.

**Model Architecture** DuplexOmni model follows the Thinker-Talker speech generation structure in the Qwen-Omni (Xu et al., 2025; Qwen Team, 2025; 2026b) family. The Thinker is the internal MLLM backbone that processes the current context and generates Assistant text tokens. The Talker converts the generated linguistic states into streaming speech. For slice $t$, the Thinker produces an Assistant token sequence together with the aligned token embeddings and hidden states. We denote

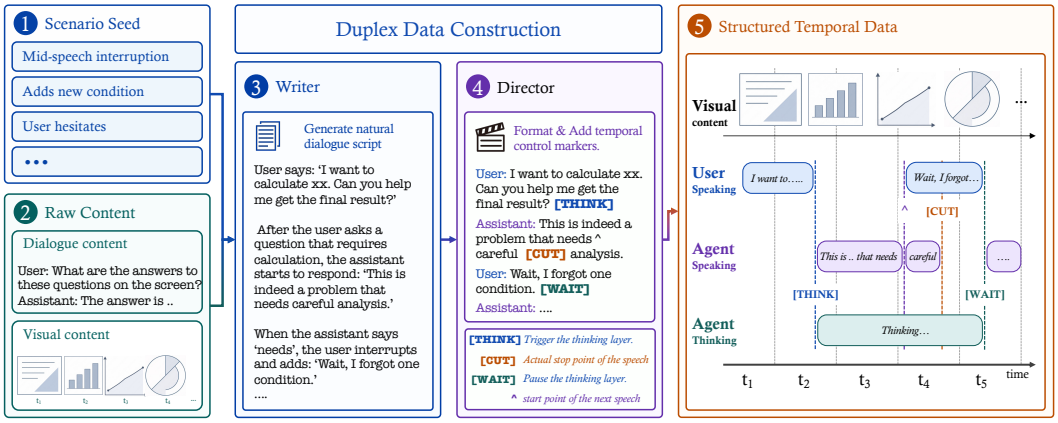


Figure 3: Data pipeline for DuplexOmni. Scenario seeds and raw content are converted into temporally annotated full-duplex dialogues; representative control tokens are shown, with the full token set detailed in Appendix F.

them by $E_t$ and $H_t$, respectively. These two sequences are projected into the Talker conditioning space and added row-wise:

$$c_{t,\ell} = f_{\text{text}}(e_{t,\ell}) + f_{\text{hidden}}(h_{t,\ell}), \qquad \ell = 1, \ldots, m_t,$$

where $m_t$ is the number of Assistant tokens in the current slice. We denote the resulting conditioning sequence by

$$C_t = (c_{t,1}, \ldots, c_{t,m_t}).$$

The Talker does not generate speech from $C_t$ alone. It preserves both the historical conditioning sequences and the historical speech codec history. Let $Q_i$ denote the complete residual vector quantization (RVQ) codec sequence generated in slice $i$. If slice $i$ contains $L_i$ speech frames, then

$$Q_i = (q_{i,1}, \ldots, q_{i,L_i}),$$

where each frame is represented by

$$q_{i,j} = (q^0_{i,j}, q^1_{i,j}, \ldots, q^{K-1}_{i,j}),$$

where $K$ is the number of RVQ codebooks. For each frame, the Talker constructs a summed codec embedding

$$r_{i,j} = u_0(q^0_{i,j}) + \sum_{k=1}^{K-1} u_k(q^k_{i,j}),$$

where $u_k$ is the embedding table of the $k$-th codebook, and we denote the codec embedding sequence of slice $i$ by

$$R_i = (r_{i,1}, \ldots, r_{i,L_i}).$$

At the beginning of slice $t$, the embedding prefix of the Talker is

$$P_t = \left[(C_i, b_{\text{BOS}}, R_i, b_{\text{EOS}})_{i=1}^{t-1},\ C_t,\ b_{\text{BOS}}\right].$$

Here, $b_{\text{BOS}}$ and $b_{\text{EOS}}$ denote the embeddings of the codec BOS and EOS tokens. This expression matches the actual organization of the prompt. Each historical slice contributes one conditioning segment followed by one codec segment wrapped by a single codec BOS and EOS. The current slice contributes its conditioning sequence and one codec BOS before speech generation starts. Inside the current slice, the model maintains both a token channel and an embedding channel. Conditioning positions are filled with a dedicated pad id on the token channel, while historical codec spans keep their true codec tokens. On the embedding channel, the model writes the conditioning projections and the corresponding summed codec embeddings.

Given the current prefix, the Talker autoregressively predicts the layer-0 codec token of the current speech frame:

$$q^0_{t,j} = \mathrm{Talker}(P_t, q^0_{t,<j}).$$

For each non-EOS layer-0 token, a multi-token prediction (MTP) module (Qwen Team, 2025) predicts the residual codebooks conditioned on the current Talker hidden state and the predicted layer-0 token, producing the complete RVQ code $q_{t,j}$ and its summed embedding $r_{t,j}$. The newly generated layer-0 token and summed embedding are then appended to the current prefix and used for the next frame. Therefore, within a slice, the Talker repeatedly appends newly generated codec content, rather than repeatedly inserting new BOS or EOS markers.

Each slice generates six codec frames, corresponding to 480 ms of speech, and the resulting RVQ sequence is decoded into waveform samples by the Code2Wav decoder of Qwen3-Omni. After generation, the current conditioning sequence $C_t$ and codec sequence $Q_t$ are written into history and reused to construct the Talker prefix of the next slice. In this way, DuplexOmni model remains autoregressive at the token level, continuous across adjacent time slices, and full-duplex at the interaction level.

### 3.2 Data Construction

To train the full-duplex interaction ability of DuplexOmni, the training data must describe not only what is said but also when each interaction event happens, including interruptions and the integration of delayed thinking results. Existing multi-turn dialogue data is mostly turn-based. It only records user and assistant utterances, and lacks temporal information. Therefore, it cannot tell the model when to take or yield the turn, or when to start background thinking and use its results.

We address this by dividing data construction into two stages (Figure 3): *scenario and content construction* and *Writer-Director temporal annotation*. The first stage decides what the dialogue is about and how it happens. The second stage converts natural dialogue into structured samples with explicit temporal causality. Since content and temporal annotation are decoupled, new tasks only require new content data and scenario configurations. The downstream pipeline remains unchanged.

#### 3.2.1 Scenario and Content Construction

This stage builds content-scenario pairs for training. Each sample contains raw content and a scenario seed: the raw content (dialogue content and, for video-call samples, visual content) provides the task semantics, while the scenario seed specifies how the interaction unfolds, such as who initiates the dialogue, whether the user may revise or add information during the exchange, and how the assistant should respond when background reasoning or turn-taking events occur. We sample scenario combinations from configuration files and discard invalid or conflicting settings. The raw content is then rewritten into a speech-friendly form that avoids written-style phrasing and overly long sentences.

#### 3.2.2 Writer-Director Data Pipeline

This stage converts content-scenario pairs into structured temporal data for full-duplex training. The Writer generates a natural dialogue script that follows the scenario. The Director adds temporal control signals.

The Writer generates a natural language script from the raw content and the scenario seed. The script preserves the task semantics and reflects the target interaction pattern. For example, in a math reasoning task, the Writer may arrange the user to ask a question, add missing conditions midway, interrupt the assistant, or insert new information during assistant reasoning. For assistant-initiated scenarios, the Writer adjusts the opening and speaking order so that the dialogue naturally starts from the assistant.

The output of the Writer is still a human-readable linear script. For example:

> User says: 'I want to calculate xx. Can you help me get the final result?' After the user asks a question that requires calculation, the assistant starts to respond: 'This is indeed a problem that needs careful analysis.' When the assistant says 'needs', the user interrupts and adds: 'Wait, I forgot one condition.'

This script describes how the dialogue happens. It does not handle control tokens or the audio timeline.

The Director converts this script into a structured sample with temporal control signals. The full definitions of special text tags are given in Appendix F. For the above script, the annotated sample can be written as:

> User: I want to calculate xx. Can you help me get the final result? `[THINK]`
> Assistant: This is indeed a problem that needs^ careful `[CUT]` analysis.
> User: Wait, I forgot one condition. `[WAIT]`

Here, `[THINK]` triggers background reasoning. `^` marks the time point where the user speech stream starts. `[CUT]` marks the actual stopping point of the interrupted assistant speech. `[WAIT]` means that the user adds a new condition, so the ongoing background reasoning is suspended until a new `[THINK]` is issued. The Director does not change the dialogue content. It adds trainable temporal causality. Complete annotated examples are given in Appendix H.

Finally, we apply consistency checks to the Director output. We filter samples with conflicting control tokens, missing thinking triggers, returned information without causal source, or unreasonable interruption points. After cleaning, we synthesize speech with TTS and segment the audio into fixed-size chunks, so that the resulting data follows the same time-sliced format used by DuplexOmni model training and inference. Through this pipeline, DuplexOmni learns both the turn-taking behavior of full-duplex dialogue and when to request and use background thinking.

### 3.3 Real-Time Duplex Inference

To enable real-time duplex inference, DuplexOmni uses $\mathrm{RTF} < 1$ as the latency target for speech generation. RTF is the ratio between generation time and audio duration. A value below 1 means that each audio chunk can be generated before it finishes playing. DuplexOmni decouples Thinker-based text generation from Talker-based speech generation and runs them as an asynchronous pipeline. This avoids serial latency accumulation across continuous chunks. For the Talker path, we apply KV-cache-based incremental decoding and graph execution optimization to autoregressive codec decoding and MTP computation. These optimizations reduce redundant computation and scheduling overhead and support stable low-latency speech output.

## 4 Experiments

Table 1: Interaction-layer comparison with real-time full-duplex and speech-to-speech baselines. All models with real-time capabilities are evaluated in real-time mode, and Daily-Omni is evaluated under streaming mode.

| Model | DuplexBench ToR (%) | Big Bench Audio (%) | Daily-Omni (%) | LibriSpeech WER (%) | Latency (s) |
|---|---|---|---|---|---|
| DuplexOmni | **72.6** | **77.2** | 53.8 | 11.92 | 0.506 |
| MiniCPM-o 4.5 | 36.3 | 36.0 | 80.2 | – | **0.502** |
| Doubao | 27.8 | 47.9 | – | – | 1.82 |
| Qwen3-Omni-Realtime-Flash | 25.2 | 32.1 | 76.2 | 1.3 | 1.28 |
| Qwen3.5-Omni-Realtime-Flash | 26.4 | 35.0 | **81.8** | **1.1** | 1.25 |
| Gemini-3.1-Flash-Live | 24.1 | 57.9 | – | 4.1 | 2.57 |
| Gemini-3.1-Flash-Lite | – | 58.9 | 59.0 | 3.8 | – |

### 4.1 Settings

We compare DuplexOmni with recent real-time omni models and speech-to-speech systems, including MiniCPM-o, Doubao (via the Volcano Engine API), Qwen-Omni real-time variants, and Gemini-3.1-Flash-Live; we also report the default thinking layer, Gemini-3.1-Flash-Lite, used on its own. All models are evaluated under their streaming or real-time settings; proprietary models are evaluated through official APIs, and open-source models use public weights with officially recommended inference configurations. We evaluate models on Full DuplexBench (DuplexBench) (Lin

et al., 2025), Big Bench Audio (Srivastava et al., 2022; Suzgun et al., 2022), Daily-Omni (Zhou et al., 2025), and LibriSpeech (Panayotov et al., 2015) WER. Full DuplexBench measures full-duplex interaction quality, Big Bench Audio measures streaming audio understanding, and Daily-Omni measures general omni capability. For models with ASR capability, we report LibriSpeech as a reference for speech recognition quality. We also report latency as a measure of real-time response delay. Unless otherwise specified in ablation experiments, DuplexOmni uses the full configuration, where the DuplexOmni model serves as the interaction layer and Gemini-3.1-Flash-Lite (Reid et al., 2024) serves as the thinking layer.

### 4.2 DATA

The training data construction consists of two parts: scenario seeds and dialogue content. We build about 620K scenario seeds to define the distribution of continuous interaction settings. These seeds cover five types of interaction scenarios (Appendix I) and annotate attributes such as interaction type, required events, and user style, so that the resulting data spans full-duplex interactions with different densities, rhythms, and levels of complexity.

The dialogue content contains about 3.02M raw conversations, including 10K video-call conversations. The text-dialogue portion includes about 1.486M user-initiated conversations and 1.528M assistant-initiated conversations, sourced from UltraChat (Ding et al., 2023), WildChat (Zhao et al., 2024a), BELLE (Ji et al., 2023), COIG (Zhang et al., 2023b), no-robots (Rajani et al., 2023), and OASST2 (Köpf et al., 2023). The overall language distribution is mainly Chinese, about 70%, with about 30% English and a small amount of other languages. Data construction uses Qwen3.5-397B-A17B (Qwen Team, 2026c) and, for later data, DeepSeek-V4-Flash (DeepSeek-AI, 2026) as the generation and annotation models. Speech training data is synthesized with Qwen3-TTS (Qwen Team, 2026a) and converted into codec tokens with the Mimi (Défossez et al., 2024) encoder.

### 4.3 TRAINING

DuplexOmni treats time-sliced full-duplex interaction as continuous multi-turn dialogue. Initialized from Qwen3-Omni, it is trained with two-stage SFT: the first stage builds basic full-duplex listening and speaking ability on large-scale speech interaction data, while the second stage further improves complex interaction and video-call performance using high-quality interaction and video-call data.

In each stage, we alternately optimize the Thinker and the Talker. When training the Thinker, we freeze the Talker and compute only the cross-entropy loss of the Thinker. When training the Talker, we freeze the Thinker and compute only the cross-entropy loss of the Talker's mixture-of-experts (MoE) layers and the MTP module. The data ratio between the two optimization parts is 1:1. The learning rate is 1e-5 for the Thinker and 1e-4 for the Talker, with a batch size of 128. Training is conducted with Megatron-swift-3.12 (Zhao et al., 2024b) on 128 Nvidia H20 GPUs.

### 4.4 PERFORMANCE COMPARISON

Table 1 reports the main comparison between DuplexOmni and recent real-time omni models or speech-to-speech systems. The results show two main observations:

**Strong full-duplex interaction.** DuplexOmni achieves 72.6% ToR on Full DuplexBench, substantially outperforming all real-time baselines, while keeping a response latency of 0.506s. This shows that the DuplexOmni model can handle continuous input and output under strict real-time constraints.

**Reasoning without sacrificing interaction.** DuplexOmni achieves the best Big Bench Audio score of 77.2%, although its Daily-Omni score (53.8%) is lower than those of all other models with Daily-Omni results. This suggests that the layered design preserves real-time interaction quality while using the thinking layer to improve audio understanding and complex reasoning.

Table 2: Ablation study on DuplexOmni. Full DuplexBench measures interaction quality, while Big Bench Audio evaluates audio understanding under the streaming setting.

| Model | DuplexBench ToR (%) | Big Bench Audio (%) |
|---|---|---|
| DuplexOmni | 72.6 | 77.2 |
| Weak Thinking | 72.1 | 50.3 |
| w/o Thinking | 65.2 | 22.2 |
| w/o Thinking & ASR | 67.2 | 18.1 |
| Thinking Only | – | 58.9 |

Table 3: LibriSpeech WER (%) by utterance length (words). Short utterances are substantially harder for DuplexOmni under the full-duplex streaming setting.

| Length | Gemini Lite | Gemini Live | Duplex Omni |
|---|---|---|---|
| 1–5 | 4.9 | 7.8 | 25.1 |
| 6–10 | 5.3 | 4.4 | 15.4 |
| 11–15 | 3.4 | 3.3 | 12.0 |
| 16–20 | 2.9 | 2.2 | 10.6 |
| 21+ | 3.2 | 4.6 | 8.8 |
| Overall | 3.75 | 4.09 | 11.92 |

### 4.5 Ablation Study

Table 2 presents the ablation results of DuplexOmni. *Weak Thinking* replaces the default thinking layer with a weaker model, GPT-4o-mini. *w/o Thinking* removes the thinking layer and keeps only the interaction layer. *w/o Thinking & ASR* further removes speech-to-text output. *Thinking Only* directly uses the thinking layer to process audio input without the DuplexOmni interaction layer. Based on these settings, we obtain three observations.

**The full-duplex ability of the interaction layer is insensitive to the strength of the thinking layer.** Replacing the thinking layer with a weaker model only changes Full DuplexBench from 72.6% to 72.1%. This shows that full-duplex interaction quality is mainly determined by the DuplexOmni model, and is not sensitive to the strength of the thinking layer.

**The thinking layer determines the reasoning ceiling.** When the thinking layer is replaced by a weaker model, Big Bench Audio drops from 77.2% to 50.3%; removing the thinking layer further reduces it to 22.2%. Together with the first observation, this shows that the thinking layer is a pluggable component of DuplexOmni.

**The interaction layer improves the effectiveness of the thinking layer.** *Thinking Only* directly processes audio input with the thinking layer and reaches 58.9% on Big Bench Audio, while the full DuplexOmni reaches 77.2%. This indicates that the gain does not only come from the external strong model. Instead of raw audio, the thinking layer receives the context organized by the interaction layer, including dialogue text, video information, and task state (Section 3.1.1 and Appendix C), which makes the thinking process more effective.

### 4.6 Full-Duplex ASR Analysis

Table 3 shows that DuplexOmni performs worse on short utterances, with WER dropping from 25.1% for 1–5 words to 8.8% for 21+ words. This suggests that short, low-context speech fragments in full-duplex interaction remain a key challenge.

## 5 Conclusion

We present DuplexOmni, a real-time multimodal full-duplex interaction method that decouples low-latency interaction from asynchronous thinking and tool use. By combining the DuplexOmni interaction model with a pluggable thinking layer and a Writer-Director data pipeline, DuplexOmni supports continuous listening, seeing, speaking, and reasoning. Experiments show that it achieves strong full-duplex interaction performance while maintaining low response latency. Limitations are discussed in Appendix A.

### AI USE STATEMENT

In this work, generative AI tools were used to generate and annotate synthetic training data as described in Section 4.2, and to assist with language polishing and LaTeX formatting. The authors reviewed all AI-assisted work and take responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

Full-duplex voice agents that can speak proactively, including assistant-initiated calls such as Example 4 in Appendix H, carry dual-use risks such as impersonation and fraudulent calls. DuplexOmni is designed to limit these risks. The Talker uses a single fixed synthetic voice that is not designed to imitate any real person, and its interface accepts no reference audio, speaker embeddings, voice-cloning controls, or prompt-based voice switching, so the model cannot directly clone an uploaded voice. Changing the voice would require large amounts of target speech data and retraining the Talker. We recommend that deployments disclose to users that they are interacting with an AI system, especially for assistant-initiated calls. The text dialogues used for training are derived from publicly available datasets (Section 4.2).

### REPRODUCIBILITY STATEMENT

We have released the model weights, training data, and training and inference code, and an anonymized copy of the code is included in the supplementary material. The model is described in Section 3.1.2, its training in Section 4.3 and Appendix D, and the interaction–thinking protocol in Appendix C. Data construction is described in Section 3.2 and Appendix G, with the control tokens, annotated examples, and data statistics in Appendices F, H, and I. The evaluation settings are given in Section 4.1 and Appendix E. The thinking layers in our experiments are proprietary models accessed through APIs; other LLMs or agents can be used through the same text interface (Appendix C).

## REFERENCES

Baichuan Inc. Baichuan-Omni-1.5 technical report. *CoRR*, abs/2501.15368, 2025. doi: 10.48550/ARXIV.2501.15368. URL https://doi.org/10.48550/arXiv.2501.15368.

Chuangtao Chen, Grace Li Zhang, Xunzhao Yin, Cheng Zhuo, Ulf Schlichtmann, and Bing Li. LiveMind: Low-latency large language models with simultaneous inference. *CoRR*, abs/2406.14319, 2024. doi: 10.48550/ARXIV.2406.14319. URL https://doi.org/10.48550/arXiv.2406.14319.

Junjie Chen, Yao Hu, Junjie Li, Kangyue Li, Kun Liu, Wenpeng Li, Xu Li, Ziyuan Li, Feiyu Shen, Xu Tang, Manzhen Wei, Yichen Wu, Fenglong Xie, Kaituo Xu, and Kun Xie. FireRedChat: A pluggable, full-duplex voice interaction system with cascaded and semi-cascaded implementations. *CoRR*, abs/2509.06502, 2025a. doi: 10.48550/ARXIV.2509.06502. URL https://doi.org/10.48550/arXiv.2509.06502.

Wenxi Chen, Ziyang Ma, Ruiqi Yan, Yuzhe Liang, Xiquan Li, Ruiyang Xu, Zhikang Niu, Yanqiao Zhu, Yifan Yang, Zhanxun Liu, Kai Yu, Yuxuan Hu, Jinyu Li, Yan Lu, Shujie Liu, and Xie Chen. SLAM-Omni: Timbre-controllable voice interaction system with single-stage training. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pp. 2262–2282. Association for Computational Linguistics, 2025b. URL https://aclanthology.org/2025.findings-acl.115/.

Cheng-Han Chiang, Xiaofei Wang, Linjie Li, Chung-Ching Lin, Kevin Lin, Shujie Liu, Zhendong Wang, Zhengyuan Yang, Hung-yi Lee, and Lijuan Wang. STITCH: simultaneous thinking and talking with chunked reasoning for spoken language models. *CoRR*, abs/2507.15375, 2025. doi: 10.48550/ARXIV.2507.15375. URL https://doi.org/10.48550/arXiv.2507.15375.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *CoRR*, abs/2410.00037, 2024. doi: 10.48550/ARXIV.2410.00037. URL https://doi.org/10.48550/arXiv.2410.00037.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Gemini Team. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023. doi: 10.48550/ARXIV.2312.11805. URL https://doi.org/10.48550/arXiv.2312.11805.

William Barr Held, Yanzhe Zhang, Minzhi Li, Weiyan Shi, Michael J. Ryan, and Diyi Yang. Distilling an end-to-end voice assistant without instruction training data. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 7876–7891. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.388/.

Yunjie Ji, Yan Gong, Yong Deng, Yiping Peng, Qiang Niu, Baochang Ma, and Xiangang Li. Towards better instruction following language models for Chinese: Investigating the impact of training data and evaluation. *CoRR*, abs/2304.07854, 2023. doi: 10.48550/ARXIV.2304.07854. URL https://doi.org/10.48550/arXiv.2304.07854.

KimiTeam, Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, Zhengtao Wang, Chu Wei, Yifei Xin, Xinran Xu, Jianwei Yu, Yutao Zhang, Xinyu Zhou, Y. Charles, Jun Chen, Yanru Chen, Yulun Du, Weiran He, Zhenxing Hu, Guokun Lai, Qingcheng Li, Yangyang Liu, Weidong Sun, Jianzhou Wang, Yuzhi Wang, Yuefeng Wu, Yuxin Wu, Dongchao Yang, Hao Yang, Ying Yang, Zhilin Yang, Aoxiong Yin, Ruibin Yuan, Yutong Zhang, and Zaida Zhou. Kimi-Audio technical report. *CoRR*, abs/2504.18425, 2025. doi: 10.48550/ARXIV.2504.18425. URL https://doi.org/10.48550/arXiv.2504.18425.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. OpenAssistant conversations – democratizing large language model alignment, 2023. URL https://arxiv.org/abs/2304.07327.

Shufan Li and Aditya Grover. PredGen: Accelerated inference of large language models through input-time speculation for real-time speech interaction. *CoRR*, abs/2506.15556, 2025. doi: 10.48550/ARXIV.2506.15556. URL https://doi.org/10.48550/arXiv.2506.15556.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H. Liu, and Hung-Yi Lee. Full-Duplex-Bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2025, Honolulu, HI, USA, December 6-10, 2025*, pp. 1–8. IEEE, 2025. doi: 10.1109/ASRU65441.2025.11433838. URL https://doi.org/10.1109/ASRU65441.2025.11433838.

Yudong Lu, Yazhe Niu, Shuai Hu, and Haolin Wang. CleanS2S: Single-file framework for proactive speech-to-speech interaction. *CoRR*, abs/2506.01268, 2025. doi: 10.48550/ARXIV.2506.01268. URL https://doi.org/10.48550/arXiv.2506.01268.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. Language model can listen while speaking. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances*

*in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 24831–24839. AAAI Press, 2025. doi: 10.1609/AAAI.V39I23.34665. URL https://doi.org/10.1609/aaai.v39i23.34665.

Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoit Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux. Generative spoken dialogue language modeling, 2022. URL https://arxiv.org/abs/2203.16502.

OpenAI. GPT-4o system card. *CoRR*, abs/2410.21276, 2024. doi: 10.48550/ARXIV.2410.21276. URL https://doi.org/10.48550/arXiv.2410.21276.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pp. 5206–5210. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178964. URL https://doi.org/10.1109/ICASSP.2015.7178964.

Qwen Team. Qwen3-Omni technical report. *CoRR*, abs/2509.17765, 2025. doi: 10.48550/ARXIV.2509.17765. URL https://doi.org/10.48550/arXiv.2509.17765.

Qwen Team. Qwen3-TTS technical report. *CoRR*, abs/2601.15621, 2026a. doi: 10.48550/ARXIV.2601.15621. URL https://doi.org/10.48550/arXiv.2601.15621.

Qwen Team. Qwen3.5-Omni technical report. *CoRR*, abs/2604.15804, 2026b. doi: 10.48550/ARXIV.2604.15804. URL https://doi.org/10.48550/arXiv.2604.15804.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026c. URL https://qwen.ai/blog?id=qwen3.5.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. No Robots. https://huggingface.co/datasets/HuggingFaceH4/no_robots, 2023.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530, 2024. doi: 10.48550/ARXIV.2403.05530. URL https://doi.org/10.48550/arXiv.2403.05530.

Yi-Jen Shih, Desh Raj, Chunyang Wu, Wei Zhou, SK Bong, Yashesh Gaur, Jay Mahadeokar, Ozlem Kalinli, and Mike Seltzer. Can speech LLMs think while listening? *CoRR*, abs/2510.07497, 2025. doi: 10.48550/ARXIV.2510.07497. URL https://doi.org/10.48550/arXiv.2510.07497.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Junlong Tong, Yingqi Fan, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. StreamingThinker: Large language models can think while reading. *CoRR*, abs/2510.17238, 2025. doi: 10.48550/ARXIV.2510.17238. URL https://doi.org/10.48550/arXiv.2510.17238.

Tongyi Fun Team, Qian Chen, Luyao Cheng, Chong Deng, Xiangang Li, Jiaqing Liu, Chao-Hong Tan, Wen Wang, Junhao Xu, Jieping Ye, Qinglin Zhang, Qiquan Zhang, and Jingren Zhou. Fun-Audio-Chat technical report. *CoRR*, abs/2512.20156, 2025. doi: 10.48550/ARXIV.2512.20156. URL https://doi.org/10.48550/arXiv.2512.20156.

Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Yuanjun Xiong, and Wei Xia. A full-duplex speech dialogue scheme based on large language models. *CoRR*, abs/2405.19487, 2024. doi: 10.48550/ARXIV.2405.19487. URL https://doi.org/10.48550/arXiv.2405.19487.

Siyin Wang, Wenyi Yu, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Lu Lu, and Chao Zhang. End-to-end listen, look, speak and act. *CoRR*, abs/2510.16756, 2025. doi: 10.48550/ARXIV.2510.16756. URL https://doi.org/10.48550/arXiv.2510.16756.

Donghang Wu, Haoyang Zhang, Jun Chen, Xiangyu Tony Zhang, Hexin Liu, Eng Siong Chng, Fei Tian, Xuerui Yang, Xiangyu Zhang, Daxin Jiang, and Gang Yu. Mind-Paced Speaking: A dual-brain approach to real-time reasoning in spoken language models. *CoRR*, abs/2510.09592, 2025. doi: 10.48550/ARXIV.2510.09592. URL https://doi.org/10.48550/arXiv.2510.09592.

LLM-Core Xiaomi. MiMo-Audio: Audio language models are few-shot learners. *CoRR*, abs/2512.23808, 2025. doi: 10.48550/ARXIV.2512.23808. URL https://doi.org/10.48550/arXiv.2512.23808.

Zhifei Xie, Ziyang Ma, Zihang Liu, Kaiyu Pang, Hongyu Li, Jialin Zhang, Yue Liao, Deheng Ye, Chunyan Miao, and Shuicheng Yan. Mini-Omni-Reasoner: Token-level thinking-in-speaking in large speech models. *CoRR*, abs/2508.15827, 2025. doi: 10.48550/ARXIV.2508.15827. URL https://doi.org/10.48550/arXiv.2508.15827.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni technical report. *CoRR*, abs/2503.20215, 2025. doi: 10.48550/ARXIV.2503.20215. URL https://doi.org/10.48550/arXiv.2503.20215.

Jianing Yang, Yusuke Fujita, and Yui Sudo. DuplexCascade: Full-duplex speech-to-speech dialogue with VAD-free cascaded ASR-LLM-TTS pipeline and micro-turn optimization. *CoRR*, abs/2603.09180, 2026. doi: 10.48550/ARXIV.2603.09180. URL https://doi.org/10.48550/arXiv.2603.09180.

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

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. SALMONN-omni: A standalone speech LLM without codec injection for full-duplex conversation. *CoRR*, abs/2505.17060, 2025. doi: 10.48550/ARXIV.2505.17060. URL https://doi.org/10.48550/arXiv.2505.17060.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pp. 15757–15773. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.FINDINGS-EMNLP.1055. URL https://doi.org/10.18653/v1/2023.findings-emnlp.1055.

Ge Zhang, Yemin Shi, Ruibo Liu, Ruibin Yuan, Yizhi Li, Siwei Dong, Yu Shu, Zhaoqun Li, Zekun Wang, Chenghua Lin, Wenhao Huang, and Jie Fu. Chinese open instruction generalist: A preliminary release, 2023b. URL https://arxiv.org/abs/2304.07987.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chao-Hong Tan, Zhihao Du, and Shiliang Zhang. OmniFlatten: An end-to-end GPT model for seamless voice conversation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 14570–14580. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.709/.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT interaction logs in the wild, 2024a. URL https://arxiv.org/abs/2405.01470.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Hong Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A scalable lightweight infrastructure for fine-tuning, 2024b. URL https://arxiv.org/abs/2408.05517.

Ziwei Zhou, Rui Wang, Zuxuan Wu, and Yu-Gang Jiang. Daily-Omni: Towards audio-visual reasoning with temporal alignment across modalities. *CoRR*, abs/2505.17862, 2025. doi: 10.48550/ARXIV.2505.17862. URL https://doi.org/10.48550/arXiv.2505.17862.

## A Limitations

DuplexOmni still has several limitations. First, its video capability remains limited because the amount of video-call and visually grounded interaction data is relatively small. Second, its English speech ability is weaker than desired, partly due to the training data being dominated by Chinese speech. Third, both thinking layers in our experiments are proprietary models accessed through APIs, and the Weak Thinking ablation covers only one weaker model, so the behavior with open-source thinking models of different scales remains to be evaluated. Fourth, ASR errors are high on short utterances (Table 3): within a 480 ms slice, the model must commit to the first transcription tokens with little acoustic or semantic context. Delaying the first ASR token by 4–8 slices could provide more context, but it would desynchronize the transcription from the audio and delay `[THINK]`, which depends on the streaming transcription. Fifth, we do not report human evaluations of interaction naturalness; our evaluation relies on public benchmarks and qualitative examples. We leave stronger video modeling, more balanced multilingual speech training, open-source thinking layers, and human evaluation to future work.

## B Additional Related Work

Several recent works reduce the latency cost of reasoning in spoken interaction.

**Thinking while listening.** Shih et al. (2025) fine-tune multi-stream speech LLMs with text chain-of-thought and let the model start reasoning before the user query ends, using an entropy-based question-completeness signal to decide when to start. LiveMind (Chen et al., 2024) performs inference on incomplete streaming prompts and allows a large model to reason while a smaller model produces the output. PredGen (Li & Grover, 2025) speculatively generates candidate responses while the user is still speaking, so that a cascaded voice assistant can start speech synthesis earlier. StreamingThinker (Tong et al., 2025) lets text LLMs reason in the order of the incoming input, using streaming attention masks and separate KV caches for the input and the reasoning.

**Thinking while speaking.** STITCH (Chiang et al., 2025) alternates unspoken reasoning chunks and spoken response chunks in a spoken language model, generating the next reasoning chunk while the current audio chunk is played. Mind-Paced Speaking (Wu et al., 2025) uses a dual-brain design in which a Formulation Brain performs reasoning that paces a separate Articulation Brain for speech generation. Mini-Omni-Reasoner (Xie et al., 2025) interleaves reasoning tokens with response tokens at the token level.

**Full-duplex multimodal models.** ELLSA (Wang et al., 2025) is an end-to-end full-duplex model that perceives and generates across vision, text, speech, and action within a single architecture.

**Relation to DuplexOmni.** These methods mainly reduce the latency of reasoning for a single query or a single spoken response. Mind-Paced Speaking also separates reasoning from speech generation, but it focuses on reasoning while a spoken response is produced. DuplexOmni instead targets full-duplex interaction, in which the user may interrupt, add conditions, or stay silent while reasoning is still running. It places reasoning in a pluggable thinking layer, which can be any LLM or tool agent, and lets the interaction layer decide when to start (`[THINK]`), suspend (`[WAIT]`), and verbalize (<...>) thinking results. Early reasoning and chunked reasoning are complementary to this design and could be used inside the thinking layer to further reduce its latency.

## C Interaction–Thinking Protocol

This section describes how the interaction layer and the thinking layer exchange information at inference time (Section 3.1.1).

**Context for the thinking layer.** The server keeps a text record of the ongoing dialogue. It collects the incremental user transcription produced by the interaction layer and the assistant text that has actually been synthesized into speech, merges consecutive fragments from the same speaker, and serializes them as turns prefixed with `[User]:` and `[Agent]:`. Raw codec tokens are not passed to the thinking layer. In video mode, up to 50 recent frames are also provided.

**Request lifecycle.** When the interaction layer emits `[THINK]`, the current dialogue record is appended to the thinking-layer history and a new request starts. If a request is still running, it is canceled and restarted with the updated context. When the interaction layer emits `[WAIT]`, the running request is stopped until the next `[THINK]`.

**Returning results.** The thinking layer streams its answer as short fragments. The fragments are queued and fed to the interaction model through the "from thinking layer" input of the following slices (Figure 2b), and the interaction layer decides when and how to verbalize them. The current implementation merges fragments with deterministic rules and does not use a learned compressor.

**Replacing the thinking layer.** Because the protocol only exchanges text, plus frames in video mode, the thinking layer can be any LLM or tool agent that accepts this input and returns text. Our experiments use Gemini-3.1-Flash-Lite as the default thinking layer and GPT-4o-mini for the Weak Thinking ablation, both accessed through official APIs.

## D Talker Training Details

Training the Talker requires target RVQ codec tokens, but the audio-to-codec encoder of Qwen3-Omni is not publicly available. We therefore generate the targets with the Mimi encoder (Défossez et al., 2024), using 16 RVQ layers ($K = 16$) at 12.5 Hz, which yields six codec frames per 480 ms slice. In a cross-decoding check, decoding Mimi-encoded tokens with the Code2Wav decoder of Qwen3-Omni produced speech that was largely consistent with the source audio in content and perceptual quality, which supports using these tokens as Talker targets. Training uses teacher forcing: the layer-0 tokens supervise the autoregressive prediction of the Talker, and the remaining 15 layers supervise the MTP module. In addition to the Talker's MoE layers and the MTP module, we train the projections $f_{\text{text}}$ and $f_{\text{hidden}}$ that map Thinker outputs into the Talker conditioning space (Section 3.1.2).

## E Evaluation Details

**Systems.** Proprietary systems, including Doubao on Volcano Engine, the Qwen-Omni real-time variants, and the Gemini models, are evaluated through their official APIs. Because these services

are updated over time, their results reflect the versions available at the time of evaluation. MiniCPM-o 4.5 is evaluated with its public weights and the officially recommended inference configuration.

**Latency.** For all systems, latency is measured end to end, from the moment the user stops speaking to the moment the client receives the first audio packet of the response. It therefore includes network transmission and server-side processing.

**Daily-Omni.** Daily-Omni pairs a complete video with a text question and is designed for offline evaluation. To evaluate it in streaming mode, we convert each question to speech and insert it into the video stream. When a video ends before the model can finish its spoken answer, the video is replayed so that the model has enough time to respond.

## F Full Control Token Set

Table 4 lists the complete set of Director annotation tokens inserted during the Writer-Director pipeline. These tokens encode the temporal relationship between user speech, assistant speech, and background thinking in each training sample. Throughout the appendix, S2 refers to the thinking layer.

Table 4: Complete set of Director annotation tokens. Ghost text (after `[CUT]`) preserves what the assistant intended but did not utter.

| Token | Turn | Semantics | Placement |
|---|---|---|---|
| `[THINK]` | User / Assistant | Triggers S2 to begin background reasoning. | End of the utterance after which external reasoning is needed. |
| `<...>` | Assistant | Injects a complete S2 result fragment before the corresponding fact. | Before the relevant assistant sentence; never as the first token after a user turn. |
| ^ | Assistant | Marks the character offset where the next speaker begins. | Inside the assistant text at the overlap onset; the assistant cannot trigger the user or itself. |
| `[CUT]` | Assistant | Marks where assistant audio stops; following text is ghost text. | 2–4 Chinese characters or at least one complete English word after ^; omitted for pure backchannel overlap. |
| `[WAIT]` | User | Suspends the ongoing S2 reasoning process. | End of the interrupting user turn; a new `[THINK]` is required to resume. |
| `[PEND`$N$`S]` | User | Encodes $N$ seconds of shared silence ($N = 1$–$35$). | Start of the user turn that resumes after the silence; forbidden in assistant turns. |

The six tokens capture six distinct temporal phenomena in full-duplex interaction: (1) **thinking onset** (`[THINK]`), triggering asynchronous background reasoning at a precise moment in the dialogue; (2) **thinking delivery** (`<...>`), injecting reasoning results into the assistant stream with explicit causal ordering; (3) **speech overlap onset** (^), recording the exact character-level time point at which a second speaker begins; (4) **speech cutoff** (`[CUT]`), recording where the first speaker stops while preserving ghost text for loss computation; (5) **thinking suspension** (`[WAIT]`), suspending the ongoing reasoning when user intent changes; and (6) **shared silence** (`[PEND`$N$`S]`), encoding mutual pauses with second-level granularity.

These tokens are stripped from the text before TTS synthesis and force-alignment; the resulting character-level timestamps are used to construct the time-sliced training samples consumed by the DuplexOmni model. Appendix G describes this conversion in detail.

## G Data Construction Details

**Writer and Director.** Both the Writer and the Director are implemented by prompting LLMs (Section 4.2). The Writer produces a complete dialogue script without control tokens, including the full intended sentence even when that sentence is later interrupted. The Director only inserts temporal and thinking-layer control tokens and does not change the dialogue content.

**Quality control.** Quality control combines structured scenario seeds, a classification of whether each request can be answered directly or needs the thinking layer, format cleaning, hallucination detection, up to five rounds of repair and re-checking, and deterministic validation. For a genuine interruption, `^` marks the onset of the user speech and `[CUT]` follows 2–4 Chinese characters or at least one complete English word later; overlaps and backchannels without a physical cutoff use only `^`. The validator rejects adjacent `^` and `[CUT]`, cutoffs inside an English word, `[CUT]` without a preceding `^`, and cutoffs at the end of a sentence.

**Video data.** For video-call samples, Qwen3.5-397B-A17B first reads each video and segments it into continuous, non-overlapping events, recording their time intervals, keyframes, descriptions of the visible content, and the time windows in which questions about them are valid. The Writer then creates spoken dialogues that only refer to events already visible at each moment, and the Director adds interruption, silence, and thinking controls. Cleaning removes answers that leak future events or are not grounded in the video. Streaming video and video-call data account for about 1/30 of the data in the second training stage.

**From annotations to time-sliced samples.** After the control tokens are removed, the full text of each turn is synthesized with TTS and aligned to the audio at the character level by forced alignment. `^` and `[CUT]` are then mapped to audio timestamps, and when the two channels are mixed, only the assistant audio before `[CUT]` is kept. `[PEND`$N$`S]` becomes silent 480 ms chunks spanning about $N$ seconds, and `[THINK]`, `[WAIT]`, and thinking-layer results become chunk-level control and input fields. Each final sample therefore contains aligned user and assistant audio, the user transcription, the assistant text, and the interaction and thinking controls for every 480 ms slice.

## H Examples of Writer-Director Outputs

The following four examples are taken from actual Writer-Director pipeline outputs. Each example is a complete annotated dialogue produced by the Director stage for one scenario seed. Control tokens appear inline in `monospace`; `<...>` carries the S2 reasoning fragment that precedes each factual assistant utterance.

**Example 1: Interruption with condition change.** The user requests a story (`[THINK]`) but interrupts the assistant before it can finish asking a clarifying question. `^` marks the moment the user begins speaking during the assistant's turn; `[CUT]` marks where the assistant's audio actually stops (ghost text follows). `[WAIT]` resets the ongoing S2 reasoning; a new `[THINK]` later restarts it when the user adds further detail. The conversation also contains a one-second shared silence (`[PEND1S]`) while the user thinks.

> **User:** Um, I wanna hear a story. `[THINK]`
> **Assistant:** No problem. What kind of ^topic`[CUT]` did you have in mind?
> **User:** Warm and caring—family helping strangers. No need to ask so many questions. `[WAIT]`
> **Assistant:** I need to confirm the tone, or the output might be off. `[THINK]`
> **User:** Fine, just confirm already.
> **Assistant:** OK, wait a sec.
> **Assistant:** `<The story's tone is warm and healing, initially set on a street.>`Warm and comforting tone. Start on a street.
> **User:** Wait, the street's too cold. Let's change to a restaurant.
> **Assistant:** Got it. Switching to restaurant scene.
> **User:** `[PEND1S]` He needs to suffer more, or it won't feel real. `[THINK]`

**Assistant:** Let me check the previous settings to avoid conflicts.

**Assistant:** <The man is unemployed and down on his luck; his family treats him to a meal at a restaurant.>Fixed. A man, jobless and down on his luck, is invited to dinner by his family. Is that okay?

**Example 2: Backchannel without speech cutoff.** The user acknowledges mid-stream while the assistant is describing a product feature. ^ records the overlap onset, but no [CUT] follows because the assistant's speech continues uninterrupted—this is a non-disruptive backchannel. Multiple <...> fragments arrive in sequence, pacing the assistant's factual delivery.

**User:** Hi, if the site doesn't work on mobile, will SEO be affected? [THINK]

**Assistant:** I'll check that.

**Assistant:** <Mobile-friendliness is a key ranking factor; non-mobile-friendly sites will have reduced search weight.>It really matters. Most traffic now comes from phones, so search engines favor mobile-friendly sites.

**User:** Can you help me check my website?

**Assistant:** Sure, send me the URL and I'll start the check.

**User:** It's www.example.com. [THINK]

**Assistant:** Got it, connecting to the server for^ verification.

**User:** Yeah.

**Assistant:** May take a moment to load the page snapshot.

**Assistant:** <The site is not fetching real-time data; likely a network or server response issue.>Sorry, can't fetch real-time data from this site—probably a network issue. Let me try again... Still can't connect. Server response seems to be the problem.

**User:** Why won't it connect? Just tell me how to fix it.

**Assistant:** Got it. Use responsive design and compress images for faster mobile loading.

**Example 3: Delayed thinking feedback with shared silence.** The user shifts the question mid-task (a new [THINK]) and twice falls silent before continuing ([PEND2S], [PEND3S]). S2 returns two short fragments in sequence. When the user corrects a factual error, a new [THINK] re-queries S2, and the assistant acknowledges the mistake.

**User:** Hey, you busy? Got a quick question—heard of a book called *Young Soul Rebels*? [THINK]

**Assistant:** I'm free. About this book, I'll answer in two parts: main content first, then social context. Just a moment while I check the database.

**User:** [PEND2S] Wait, skip the summary. I just want to know why Northern Soul rose—especially the drug culture part. Is that really detailed in the book? [THINK]

**Assistant:** Got it. I'll recheck the key sections, focusing on drug culture. It'll take a moment.

**User:** [PEND3S] Fine, hurry up. Nothing vague this time.

**Assistant:** Found it.

**Assistant:** <The book mentions that dancers in the 1970s used amphetamines to dance all night, prompting police raids on nightclubs.>The book mentions dancers used amphetamines to dance all night in the '70s, leading police to raid clubs.

**Assistant:** <The Twisted Wheel club in Manchester was raided.>For example, Manchester's Twisted Wheel Club got raided.

**User:** Twisted Wheel? I thought the Wigan Casino was more famous. Did you mix them up? [THINK]

**Assistant:** Sorry, I was going on an initial search. Since you mentioned Wigan Casino, let me double-check.

**Assistant:** <The Wigan Casino was frequently raided due to overcrowding and drug use.>You're right—the Wigan Casino has been raided multiple times over crowd and drug issues. The earlier info was inaccurate.

**Example 4: Assistant-initiated outbound dialogue.** The assistant opens the conversation proactively. The user's first turn includes a two-second silence (`[PEND2S]`) reflecting a natural phone pick-up delay, followed by a one-second pause later in the call (`[PEND1S]`). Two `[THINK]` tokens trigger S2, first for the school's resources and then, near the end, for real-time availability; each result is delivered as a <. . .> fragment after a brief holding phrase.

**Assistant:** Hi, this is London Floral School. We noticed you're interested in flower design—just a quick intro to our flexible courses, from beginner to expert.

**User:** `[PEND2S]` Hey, I was just looking at flower shop pics. Does your school have programs for different experience levels?

**Assistant:** Our classes are fully flexible—book anytime. Whether you're a total beginner or looking to improve, we've got the right course. With 16 years of teaching experience, many graduates have opened their own flower shops.

**User:** Sounds good. Any special resources during the program? `[THINK]`

**Assistant:** Let me check.

**Assistant:** `<Fresh flower wholesale market on site; students practice with flowers imported directly from Dutch growers and auctions.>`Our biggest advantage is a live wholesale market on campus. Students pick fresh flowers for practice without going out early to buy them—all sourced directly from Dutch growers and auctions.

**User:** That's really convenient. I'm especially interested in wedding floral design. Do you teach actual wedding setups?

**Assistant:** Sure, our wedding floristry class is super popular. We cover bridal bouquets, table centerpieces, and trendy wedding arches—you'll learn structure and secure arrangement techniques.

**User:** `[PEND1S]` The flower arch sounds great, but I'm pretty busy. How long does this course take?

**Assistant:** Don't worry—we know time is precious. The wedding floristry course is modular; learn on weekends or evenings at your own pace.

**User:** Good. I'm only free weekends next month. Can you check if there are classes then? I'd like to decide soon.

**Assistant:** Sure, just a moment while I check next month's weekend schedule. `[THINK]`

**Assistant:** Sorry to keep you waiting. `<Weekend class sessions on the 12th and 19th next month still have availability.>`Got it—there are weekend classes on the 12th and 19th next month, and spots are still open. I can hold one for you and send the detailed outline to your phone shortly.

# I Scenario Distribution and Data Statistics

## I.1 Interaction Pattern Distribution

Each training sample is annotated by the Director with one or more temporal control tokens. Because a single dialogue commonly exhibits multiple interaction phenomena—for example, a conversation may contain both an interruption and a shared silence—the categories below are *not* mutually exclusive. Table 5 reports the proportion of training samples that contain each pattern.

## I.2 Scenario Seed Distribution

Beyond token-level statistics, each sample is seeded with a primary interaction type that determines the overall dialogue structure. Table 6 shows the distribution of primary interaction types across the scenario seeds.

## I.3 Additional Data Statistics

Table 7 summarizes key statistics of the annotated training corpus.

Table 5: Proportion of training samples containing each interaction pattern. Categories are non-exclusive; 90.7% of samples exhibit two or more patterns simultaneously.

| **Interaction Pattern** | **Occurrence (%)** |
|---|---|
| Delayed reasoning (`[THINK]` + S2 fragments) | 94.3 |
| Shared silence (`[PENDNS]`) | 68.2 |
| Assistant-initiated turn | 50.0 |
| Overlapping speech (`^` + `[CUT]`) | 49.8 |
| Interruption with reset (`[CUT]` + `[WAIT]`) | 41.9 |
| Pure backchannel (`^` without `[CUT]`) | 3.1 |
| Samples containing $\geq$2 patterns | 90.7 |

Table 6: Distribution of primary interaction types in DuplexOmni scenario seeds. The distribution is intentionally diversified to expose the model to different temporal structures and improve robustness under continuous real-time interaction.

| **Scenario Type** | **Ratio (%)** |
|---|---|
| Step-by-step guidance | 32.1 |
| Standard turn-taking | 31.3 |
| Condition revision | 18.4 |
| Troubleshooting guidance | 13.9 |
| Extended listening | 4.4 |

Table 7: Key statistics of the DuplexOmni training corpus after Writer-Director annotation.

| **Statistic** | **Value** |
|---|---|
| Total scenario seeds | $\sim$620K |
| Total raw conversations | $\sim$3.02M |
| Average turns per sample | 10.4 |
| Turn range | 4–33 |
| S2 participation: high | 23.6% |
| S2 participation: medium | 46.0% |
| S2 participation: low | 24.9% |
| S2 participation: off | 5.5% |