TASTE2: Text-Aligned Speech Modeling and Deployment toward Full-Duplex Voice Interaction

Yi-Chang Chen[*1], **Chun Wei Chen**[*23], **Dien-Ruei Wu**[*23], **Jie Lin**[3], **Hung-yi Lee**[3], **Da-Shan Shiu**[1]

[1]MediaTek Research, [2]Internship at MediaTek Research, [3]National Taiwan University

[*]Equal contribution

Full-duplex voice interaction requires more than converting one complete utterance into another. A system must process speech as it arrives, decide when to take or yield the floor, and stop when the user interrupts, while preserving pretrained linguistic competence and acoustic paralinguistic cues. We ask whether TASTE (Text-Aligned Speech Tokenization and Embedding) provides a viable modeling path toward this goal.

We present TASTE2, which transforms the original utterance-level TASTE method into an incremental dialogue stack. A shared text-token vocabulary removes word-level averaging across the Speech Tokenizer, Spoken LM, and Speech Detokenizer; modality-aligned dialogue training predicts one continuous audio latent for each text token without interleaving heterogeneous token streams; and an incremental Speech Detokenizer enables streaming synthesis through CosyVoice2. After speech and dialogue training, TASTE2 (Merge) reaches 56.3% on LLaMA-Questions against a 64.3% few-shot text-only reference (87.6% accuracy retention), and TASTE2 (Direct) reaches 53.0% (82.4% retention).

We also move beyond model-level experiments by building TASTE2 VoiceBot, a working research system that processes user speech incrementally, streams synthesized audio, and stops generation when the user barges in. On Full-Duplex-Bench v1.0, TASTE2 achieves the best value in four benchmark columns and ties the best in a fifth among the displayed systems, with strengths in pause handling, backchanneling, and interruption. The profile is uneven: natural-conversation conditions remain challenging, and deployed mean time to first audio remains 2.701 s on an NVIDIA RTX A6000 after TensorRT acceleration.

Finally, to our knowledge, we provide the first systematic characterization of explicit paralinguistic control in a TASTE-based model. Fast speaking rate serves as a cross-strategy proof of concept after dialogue SFT, while emotion control is strategy-dependent and the remaining attributes stay weak. Together, these results establish TASTE-based modeling as a practical route toward full-duplex systems while identifying natural-conversation robustness, speech-generation latency, and feature-general paralinguistic control as open challenges.

Contact: {yi-chang.chen, wilson-cw.chen, dienruei.wu, ds.shiu}@mtkresearch.com, {b11705048, hungyilee}@ntu.edu.tw

Contents

1 Introduction

Natural voice interaction extends beyond mapping an utterance to a response. A conversational system must interpret linguistic content and paralinguistic cues, detect when to take the floor, and stop or revise its response when the user interrupts. Full-duplex dialogue systems pursue these capabilities through continuous bidirectional processing, including turn taking, backchanneling, and barge-in handling (Défossez et al., 2024; Tang et al., 2024b; Wang et al., 2024; Lin et al., 2025b). Two tensions make this goal difficult.

First, adding speech representations can weaken the linguistic competence inherited from a pretrained language model (Cui et al., 2025; Défossez et al., 2024). Text pretraining establishes regularities among text tokens. Interleaving those tokens with acoustically oriented codes forces the model to represent linguistic content and acoustic realization in the same sequence, which can disrupt the patterns learned during pretraining. Existing systems mitigate this effect with inner-monologue decoding (Défossez et al., 2024), multitask training (Wang et al., 2024), or large-scale joint training, but preserving text-model competence remains an open problem. The cost scales with how much of the sequence acoustic information occupies: the parallel decoding streams of Mini-Omni (Xie and Wu, 2024) give up 70.3% of the text model's benchmark accuracy and the multiple codebook streams of Moshi (Défossez et al., 2024) give up 16.9%, whereas TASTE2, whose sequence stays as long as the text sequence, gives up 12.4% (§5.1).

Second, system integration trades modularity against information preservation. End-to-end models jointly optimize speech understanding, response generation, and synthesis (Zeng et al., 2024; Tang et al., 2024b), allowing paralinguistic information to flow across the model but coupling the components operationally. Modular ASR–LLM–TTS pipelines allow each component to be improved independently, yet their text-only interfaces discard prosody, emotion, and speaking style before these cues can influence the response (Wang et al., 2025b; Liu et al., 2025). A practical architecture should retain relevant acoustic information without giving up modular deployment.

TASTE (Text-Aligned Speech Tokenization and Embedding) (Tseng et al., 2025) offers a useful starting point. It aligns one speech representation with each text token, separating linguistic content from acoustic realization while avoiding the sequence-length mismatch common to joint text–speech modeling. Its original implementation, however, is not designed for incremental dialogue. The Whisper-based Speech Tokenizer and LLaMA tokenizer use different vocabularies, so TASTE averages representations over word-level boundaries. This operation can discard fine-grained information, requires language-dependent segmentation, and complicates dialogue templates. Moreover, its synthesis stack operates on complete utterances rather than incrementally. TASTE also did not evaluate whether its aligned representations support turn-taking, interruption handling, or explicit paralinguistic control.

We present TASTE2, a model family and deployment architecture that establish text-aligned speech modeling as a viable path toward full-duplex voice interaction. The central representation contains two aligned information streams: discrete *text tokens* carry linguistic content, and continuous *audio latents* carry acoustic information. TASTE2 preserves this distinction throughout tokenization, language modeling, and synthesis. We make four contributions:

- **An end-to-end adaptation of TASTE into an incremental dialogue stack.** A shared text-token vocabulary removes word-level averaging, modality-aligned dialogue training predicts one continuous audio latent per text token without interleaving heterogeneous token streams, and an incremental Speech Detokenizer enables streaming synthesis.
- **A working deployment.** TASTE2 VoiceBot processes user speech incrementally, decides when to take the floor, streams synthesized audio, and stops when the user barges in.
- **Semantic preservation and uneven full-duplex evidence.** TASTE2 (Merge) reaches 56.3% on LLaMA-Questions, retaining 87.6% of the few-shot text-only reference accuracy under zero-shot evaluation while keeping the sequence at text length; moreover, 3.3 of the 11.3 points the weaker strategy gives up are attributable to how instruction-following capability is introduced rather than to speech modeling. On Full-Duplex-Bench v1.0, TASTE2 is the only displayed system that both reacts to interruptions immediately (0.060 s) and stays coherent afterwards (GPT-4o 3.275). Natural-conversation robustness and speech-generation latency remain open challenges.

- **The first systematic characterization of explicit paralinguistic control in a TASTE-based model.** Fast speaking rate serves as a cross-strategy proof of concept after dialogue SFT, while emotion control is strategy-dependent and the remaining attributes stay weak.

Together, these contributions move TASTE from utterance-level spoken language modeling to an implemented dialogue system and provide evidence that its aligned representation is a practical foundation for further full-duplex development. They also delimit what remains: active overlap behaviors are not yet deployed, natural-conversation interaction is not uniformly strong, speech-generation latency remains substantial, and explicit paralinguistic control is not consistent across attributes.

2 Related Work

TASTE2 connects text-aligned speech modeling with incremental dialogue deployment. We review speech tokenization and joint text–speech modeling (Section 2.1), full-duplex architectures and turn taking (Section 2.2), and paralinguistic control and adaptation (Section 2.3).

2.1 Speech Tokenization and Spoken Language Models

2.1.1 From Cascades to Joint Modeling

Traditional ASR–LLM–TTS cascades allow each component to be optimized independently. Because only text crosses their interfaces, however, paralinguistic cues such as emotion, prosody, and speaking rate cannot influence the language model (Yue et al., 2024; Wang et al., 2025a). GSLM (Lakhotia et al., 2021; Lee et al., 2022) showed that language models can instead generate coherent speech directly from discrete self-supervised units, and AudioLM (Borsos et al., 2023) extended this to long-form audio continuation by cascading semantic and acoustic token streams. Subsequent work grafted speech onto pretrained text LLMs. SpeechGPT (Zhang et al., 2023) and AudioPaLM (Rubenstein et al., 2023) did so through cross-modal instruction tuning, while Qwen-Audio (Chu et al., 2023) and SALMONN (Tang et al., 2024a) used encoder-adapter interfaces. These efforts established joint text-speech modeling as the dominant paradigm, though whether the resulting models preserve the semantic competence of the underlying text model remains the central open question.

2.1.2 Speech Tokenization

The representation over which a spoken language model operates largely determines how much semantic competence survives. Acoustic tokenizers such as SoundStream (Zeghidour et al., 2022) and EnCodec (Défossez et al., 2023) use residual vector quantization for high-fidelity reconstruction, but their codes are optimized for signal fidelity rather than linguistic content, yielding token sequences that are long and only loosely related to text. Semantic tokenizers derived from self-supervised models such as HuBERT (Hsu et al., 2021) carry more linguistic information but discard the acoustic detail needed for expressive synthesis. Hybrid designs attempt to span both. SpeechTokenizer (Zhang et al., 2024) distills semantic information into the first RVQ layer, CosyVoice (Du et al., 2024a,b) supervises tokens with an ASR objective, and GLM-4-Voice (Zeng et al., 2024) derives an ultra-low-bitrate (175 bps) tokenizer from an ASR encoder. All of these, however, produce token streams at a rate decoupled from the text tokenizer, leaving a length mismatch that the language model must absorb.

2.1.3 Modality Integration Strategies

Given this mismatch, systems address it in different ways. Several mix text and speech tokens directly within a sequence. SpiritLM (Nguyen et al., 2025) interleaves at word level, Moshi (Défossez et al., 2024) predicts time-aligned text before audio through an inner monologue mechanism, Qwen2.5-Omni (Xu et al., 2025) routes the two through a Thinker-Talker split, and Mini-Omni (Xie and Wu, 2024) and Baichuan-Audio (Li et al., 2025a) emit text and audio in parallel decoding streams. Others instead absorb the resulting modality confusion with scale. GLM-4-Voice (Zeng et al., 2024) trains on 1 trillion tokens and Kimi-Audio (Team, 2025) trains on 13 million hours of audio. A further group sidesteps the problem by constraining what gets retrained. Freeze-Omni (Wang et al., 2024) keeps the LLM frozen and needs only 60K samples, and

LLaMA-Omni (Fang et al., 2025a,b) attaches a streaming speech decoder to a frozen backbone, remaining competitive with GLM-4-Voice on far less speech data.

2.1.4 Addressing Semantic Degradation

Think Before You Talk (Cui et al., 2025) diagnoses the failure directly. Mixing speech tokens into the context dilutes the linguistic patterns acquired during text pretraining, so its TurnGuide method emits turn-level text guidance before speech to compensate, though modality-interleaving within each turn still remains. TASTE (Tseng et al., 2025) addresses the mismatch at its source by using attention-based aggregation to construct one speech representation per text token. Its word-level averaging for vocabulary alignment nonetheless discards fine-grained acoustic detail and requires language-specific segmentation rules, which limits multilingual scaling. TASTE2 removes word-level averaging by sharing one text-token vocabulary across its components. Its modality-aligned objective predicts one continuous audio latent for each text token, so it requires no additional interleaving along the token axis.

2.2 Full-Duplex Dialogue Systems

Building a duplex system on top of such a model forces a second trade-off, orthogonal to the tokenization question. End-to-end joint training lets information flow freely between listening and speaking but couples every capability into one set of weights, whereas pipelines keep components separately optimizable at the cost of the boundary losses noted above. Recent surveys (Chen and Yu, 2025; Ji et al., 2024) organize the resulting design space along this axis, distinguishing engineered synchronization in modular systems from synchronization learned end-to-end.

2.2.1 End-to-End Architectures

Moshi (Défossez et al., 2024) was the first to demonstrate real-time full-duplex dialogue, modeling user and system streams in parallel at 200 ms practical latency, but its tight coupling makes individual components hard to improve in isolation. SyncLLM (Veluri et al., 2024) gives a standard autoregressive LLM a sense of wall-clock time by interleaving fixed-duration chunks of both speakers, predicting the user's in-flight chunk to mask network delay. OmniFlatten (Zhang et al., 2025a) flattens user and agent speech and text into a single sequence and progressively removes the text streams across training stages to cut latency. SALMONN-omni (Tang et al., 2024b) avoids discrete codecs entirely, operating on continuous embeddings so the model can listen to its own output for echo cancellation. NTPP (Zhang et al., 2025b) and SALM-Duplex (Hu et al., 2025) instead exploit dual-channel recordings, the former through speaker-independent Next-Token-Pair Prediction and the latter through channel fusion over a pretrained streaming encoder. These systems reach low latency, but each pays for coupled optimization with substantial speech-domain training data.

2.2.2 Turn-Taking and Interruption

Predicting when to speak has a long history independent of generative modeling. TurnGPT (Ekstedt and Skantze, 2020) predicts turn completion from lexical context, and Voice Activity Projection (Ekstedt and Skantze, 2022) learns near-future voice activity of both speakers directly from audio, supporting real-time backchannel and turn-shift decisions. Talking Turns (Arora et al., 2025) applies this lens to modern audio foundation models and finds them wanting, misjudging timing, interrupting aggressively, and rarely backchanneling, evidence that end-to-end training does not by itself confer competent turn management. Full-Duplex-Bench (Lin et al., 2025b) turns this diagnosis into a standard measurement, scoring pause handling, backchanneling, and interruption directly, its v1.5 extension (Lin et al., 2025a) adds overlap scenarios and prosodic adaptation, and FD-Bench (Peng et al., 2025) targets robustness under frequent disruption. TASTE2 VoiceBot takes the modular route. Because TASTE2 aligns audio latents with text tokens, components can exchange both forms of information and make turn-boundary decisions at text-token granularity. The deployed system processes user speech incrementally, begins ordinary responses after detected speech end, and supports user barge-in during playback.

2.3 Paralinguistic Control and Adaptation

2.3.1 Paralinguistic Control

One line of work gives the model an explicit style command and measures whether it complies. GLM-4-Voice (Zeng et al., 2024) exposes emotion, intonation, and speaking rate to natural-language instruction, so a user can ask the agent to slow down or to sound cheerful and the model only has to execute the request. VStyle (Song et al., 2025) issues the instruction as speech rather than text and reports that current models face clear limitations in controllable style adaptation. Three of its four categories, acoustic attributes, natural-language instruction, and role play, state the target explicitly, while the fourth asks for implicit empathy and so already crosses into inference from unstated cues. Style following of this kind establishes that a model can render a delivery on request. It says little about whether the model can choose one when the user names nothing.

2.3.2 Paralinguistic Adaptation

A second line removes the command, so the model must infer an appropriate style from the paralinguistic cues carried in the user's speech, including emotion but also prosody, speaking rate, and speaker attributes. ProsodyLM (Anonymous, 2025) shows that this perception can emerge from pretraining alone on word-level prosody tokens, with no instruction directing the model toward emotion or contrastive focus. TELEVAL (Li et al., 2025b) argues that this is the capability that matters most for a conversational agent, extracting cues from user speech and responding appropriately without being told to. Adaptation also affects response content and not only delivery, since a user who sounds distressed warrants different words as well as a gentler tone. EMO-Reasoning (Liu et al., 2025) covers the emotional case across turns, scoring the coherence of emotional transitions in extended dialogue, while ParaS2S (Wang et al., 2025b) scores the broader paralinguistic fit within a turn, judging whether a response matches its input in content and style together and optimizing that fit at the waveform level with reinforcement learning.

The empirical scope of this report is explicit paralinguistic control after dialogue fine-tuning, complemented by a pre-fine-tuning probe of whether a Spoken LM can continue a feature already present in a speech opening. We do not evaluate implicit paralinguistic adaptation, which remains future work.

3 TASTE2 Model and Training

TASTE2 builds on TASTE (Tseng et al., 2025) and modifies three aspects of its architecture for incremental dialogue: word-level averaging, vocabularies across components, and utterance-level synthesis (§3.1.1).

TASTE2 introduces a shared text-token space (§3.1.2) and incremental synthesis through CosyVoice2 (§3.1.3). Dialogue fine-tuning then teaches the TASTE2 Spoken LM turn-boundary and interruption patterns (§3.3.3).

Figure 1 shows the three model components. The Speech Tokenizer extracts one continuous audio latent per text token. The TASTE2 Spoken LM autoregressively predicts aligned text-token–audio-latent pairs. The Speech Detokenizer maps those pairs to S3 units, which the frozen CosyVoice2 flow-matching model and vocoder convert to audio. Training proceeds in three stages: representation learning, Spoken LM pretraining, and dialogue fine-tuning.

3.1 Model Architecture

3.1.1 Original TASTE Architecture

TASTE (Tseng et al., 2025) uses text-aligned speech tokenization for joint text-speech spoken language modeling. The framework comprises three components: a Speech Tokenizer consisting of an encoder, attention-based aggregator, and Residual Vector Quantizer; a Speech Detokenizer composed of an S3 unit decoder and vocoder; and a Text-Aligned Spoken Language Model (TASLM). An end-to-end cross-attention mechanism establishes a one-to-one correspondence between text and speech tokens at the token level. It uses soft alignment cues from the ASR encoder hidden states to aggregate multiple audio frames into text-aligned speech tokens. During training, word-level averaging connects the Speech Tokenizer and Speech Detokenizer

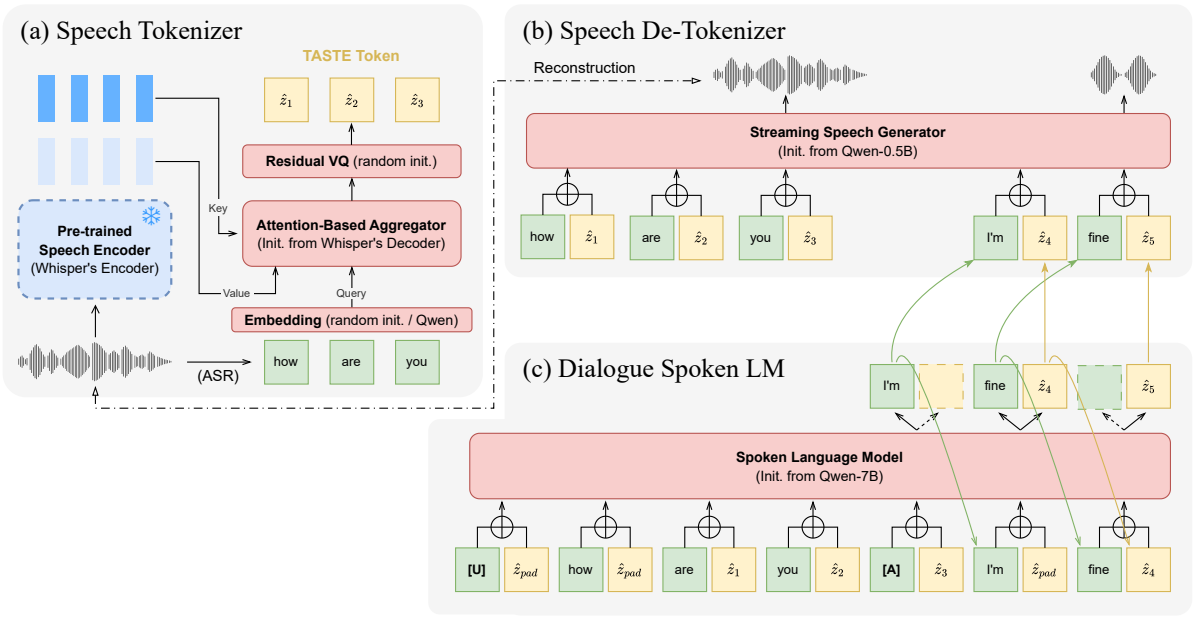


Figure 1 TASTE2 architecture. The Speech Tokenizer uses a frozen Distil-Whisper encoder to extract continuous audio latents aligned with text tokens. The TASTE2 Spoken LM autoregressively predicts text-token–audio-latent pairs. The Speech Detokenizer maps each pair to S3 units, which the frozen CosyVoice2 flow-matching model and vocoder synthesize incrementally.

vocabularies to the downstream Spoken LM vocabulary. TASLM uses a pretrained text language model (e.g., LLaMA) as its backbone and applies Low-Rank Adaptation (LoRA). It takes aligned text-token sequences as input and autoregressively generates text-speech token pairs.

TASTE performs averaging over word boundaries, so its Speech Tokenizer and Spoken LM operate with different text-token boundaries, and dialogue templates and token-level turn-boundary decisions require conversion between them. The original synthesis stack produces audio after receiving a complete utterance. TASTE2 changes the vocabulary, token alignment, and synthesis timing described below.

3.1.2 Aligned Token Space

TASTE uses different vocabularies in its Whisper-based Speech Tokenizer and LLaMA backbone, with word-level averaging between the two sequences.

TASTE2 instead uses the Spoken LM tokenizer throughout the model. The Speech Tokenizer, TASTE2 Spoken LM, and Speech Detokenizer therefore share the same text-token vocabulary and no longer require word-level averaging. Our implementation uses Qwen2.5-7B for the Spoken LM and adapts the Qwen2.5-0.5B text-to-S3 architecture from CosyVoice2 (Du et al., 2024b) as the Speech Detokenizer.

Our implementation modifies Distil-Whisper's architecture. The encoder remains frozen and retains its pretrained weights. The decoder's input embedding layer is replaced with the Spoken LM tokenization vocabulary and randomly initialized. The language modeling head is removed, and the decoder outputs continuous embeddings directly to the Residual Vector Quantizer (RVQ). This training configuration does not add a separate initialization scheme or warm-up strategy.

The Speech Tokenizer, Spoken LM, and Speech Detokenizer use the same LM tokenizer, and each text token corresponds to one audio latent. Each position in a standard text dialogue template therefore contains a text token and its aligned acoustic representation, so no additional interleaving along the token axis is required.

3.1.3 Support for Streaming Output

Incremental dialogue requires synthesis to begin before the complete response is available. TASTE2 reuses the CosyVoice2 streaming pipeline (Du et al., 2024b): a text-to-S3 model produces S3 units at 25 Hz, a chunk-aware causal flow-matching model produces mel spectrograms at 50 Hz, and a vocoder produces audio waveforms.

Only the Speech Detokenizer is modified and trained. It receives aligned text-token–audio-latent pairs from the TASTE2 Spoken LM and produces M S3 units for every N input pairs. The flow-matching model and vocoder are inherited from CosyVoice2 and remain frozen.

Streaming objective. In practice, the Speech Detokenizer produces $M = 15$ S3 units for every $N = 5$ input pairs. It learns the mapping from aligned text tokens and audio latents to the CosyVoice2 S3 space autoregressively, allowing each complete input chunk to be synthesized without waiting for the full utterance.

3.2 Data Processing

3.2.1 Data Collection

The three training stages use different data. Stages 1 and 2 use large speech corpora for representation learning and Spoken LM pretraining. Stage SFT uses conversational data containing turn transitions, interruptions, and paralinguistic instructions.

Following TASTE's data collection strategy, we use two datasets for pretraining. The **Emilia dataset** (He et al., 2024) provides 40,000 hours of English in-the-wild speech with pseudo-labeled transcriptions and includes background noise, reverberation, and multiple-speaker conditions. The **LibriTTS dataset** (Zen et al., 2019) provides 600 hours of reading-style speech with human-annotated transcriptions. Combined, these datasets total approximately 40,600 hours of training data.

Stage SFT dialogue fine-tuning uses two data sources. The **DeepDialogue dataset** comprises 37,000 multi-turn dialogues (223,673 turns, totaling 466 hours) spanning 41 domains (e.g., customer service, technical support, and casual conversation) with 20 emotion labels (e.g., happy, frustrated, and empathetic). These dialogues were generated by an ensemble of nine large language models (4B–72B parameters), followed by human annotation and quality filtering. Each dialogue was synthesized with a TTS system using the associated emotion label.

We also generated 106,507 **synthetic dialogues** (557,770 turns, totaling 1,027 hours) in two categories: general conversation data for full-duplex interaction modeling (48,000 dialogues with 388,021 turns over 778 hours) and paralinguistic control data (58,507 dialogues with 169,749 turns over 249 hours). The former includes interruption scenarios, and the latter includes explicit paralinguistic control annotations. The generation pipeline is detailed in the following subsection.

Table 1 summarizes the training data. Stages 1 and 2 use 40,600 hours of speech data. Stage SFT uses 143,507 dialogues comprising 781,443 turns and 1,493 hours of speech: 466 hours from DeepDialogue and 1,027 hours from synthetic dialogue data. The Stage SFT data include turn-taking, interruption, and paralinguistic-control annotations.

3.2.2 Synthetic Data Generation Strategy

Naturally occurring dialogue corpora rarely provide explicit interruption boundaries and controlled paralinguistic annotations. We therefore supplement DeepDialogue with 1,027 hours of synthetic data: 778 hours of general conversations with turn transitions and interruptions, and 249 hours of conversations with explicit paralinguistic controls.

General Conversation Data. The six-step pipeline (1) generates scenarios with GPT-4, (2) produces five 6–10-turn dialogues per scenario with Llama-3.3-70B, (3) uses GPT-4o to rewrite a subset with user interruptions, (4) inserts hesitations, fillers, and false starts, (5) filters dialogues with a GPT-4o judge, and (6) synthesizes the retained dialogues with CosyVoice2. We draw from 50 public reference speakers and use CosyVoice2's instruction-following mode to vary the output style. The resulting data contain interruptions at

Table 1 Training data used in Stages 1–2 and supervised fine-tuning (SFT). Dialogue and turn counts apply only to the conversational corpora used for SFT.

Stage	Dataset	Dialogues	Turns	Duration (h)
Stage 1 & 2	Emilia	–	–	40,000
	LibriTTS	–	–	600
	Subtotal	–	–	**40,600**
Stage SFT	DeepDialogue	37,000	223,673	466
	General Conversation	48,000	388,021	778
	Paralinguistic Control	58,507	169,749	249
	Subtotal	**143,507**	**781,443**	**1,493**

semantic boundaries, topic changes during Assistant responses, spontaneous-speech disfluencies, and different emotions for User and Assistant turns.

Paralinguistic Control Data. A four-step pipeline (1) generates 3–4-sentence scenarios with GPT-4, (2) alternates Llama-3.3-70B user turns with GPT-4o assistant turns to produce five approximately 10-turn dialogues per scenario, (3) filters the dialogues with a GPT-4o judge, and (4) synthesizes them with IndexTTS. In randomly selected turns, the user requests a change in speaking style and the assistant continues the conversation using the requested emotion. Synthesis uses the same pool of 50 reference speakers as the general conversation data and IndexTTS's emotion-vector mode to enforce the annotated emotion.

The data target explicit control commands and cross-turn emotion consistency.

GPT-4o scores each dialogue from 1 to 5 on six criteria: role and scenario adherence, language naturalness and variety, emotion/speed tag placement, formatting, coherence of inserted overlaps and disfluencies, and absence of extraneous stage directions. We sum the six ratings and retain the three highest-scoring dialogues for each scenario.

3.3 Training Methodology

TASTE2 training proceeds through three stages. **Stage 1** jointly trains the Speech Tokenizer and Speech Detokenizer to establish the audio-latent space. **Stage 2** adapts a pretrained text language model into the TASTE2 Spoken LM with LoRA, training it to predict aligned text-token–audio-latent pairs. **Stage SFT** fine-tunes the Spoken LM on dialogue data (§3.3.3).

The two training configurations differ in how instruction-following (IF) capability is introduced (§3.3.2). **Approach 1 (Base → Merge → SFT)** starts Stage 2 from Qwen2.5-7B and transfers IF capability through model merging before Stage SFT. **Approach 2 (Instruct → SFT)** starts Stage 2 directly from Qwen2.5-7B-Instruct and does not perform model merging.

3.3.1 Spoken Language Model Pretraining

Stages 1 and 2 follow the TASTE training procedure. Stage 1 jointly trains the Speech Tokenizer to extract speech representations from audio and the Speech Detokenizer to reconstruct audio from paired text tokens and speech representations. In Stage 2, the language model is initialized from Qwen2.5-7B for Approach 1 or Qwen2.5-7B-Instruct for Approach 2, and both configurations use the same LoRA-based fine-tuning objective to generate aligned text tokens and audio latents.

Residual Vector Quantization. During Spoken LM training, the audio-latent loss uses the final latent synthesized by the Residual Vector Quantizer (RVQ) as its target rather than a sequence of quantized indices. The RVQ is positioned between the Speech Tokenizer decoder and the audio-latent target. Section 5.2 reports the comparison between configurations with and without RVQ.

Modality-aligned training strategy. TASTE2 maintains text and audio as paired channels at the text-token rate. Each input position combines a text-token embedding with its aligned audio-latent embedding through a

weighted sum; the latent is not inserted as an additional sequence element. For autoregressive generation, the audio target is shifted by one position so that the text token is predicted before its corresponding audio latent. Text tokens and audio latents serve as the prediction targets for content and acoustic realization, respectively.

3.3.2 Instruction-Following Initialization Strategies

Approach 1 introduces IF capability after Stage 2 by transferring an instruction-tuning task vector to the pretrained Spoken LM backbone. This task-vector arithmetic (Yadav et al., 2023) requires no additional training data or compute at the merging step. The merged model then serves as the initialization checkpoint for Stage SFT.

Task-vector merging. We extract the dense language model backbone from the pretrained Spoken LM checkpoint by isolating the 339 language model weight tensors (i.e., the `model.*` and `lm_head.*` parameters) from the full Spoken LM state dict. Let W_{SLM} denote these extracted backbone weights, W_{base} the weights of Qwen2.5-7B, and W_{instruct} those of Qwen2.5-7B-Instruct. We construct a sparsified IF task vector using TIES merging (Yadav et al., 2023): the raw task vector $\tau = W_{\text{instruct}} - W_{\text{base}}$ is sparsified by retaining only the top p parameters by absolute magnitude and zeroing out the rest,

$$\hat{\tau} = \text{TopK}_p(\tau), \quad p = 0.3$$

and the merged backbone is computed as:

$$W_{\text{merged}} = W_{\text{SLM}} + \lambda \cdot \hat{\tau}, \quad \lambda = 0.5$$

The merged weights are injected back into the Spoken LM checkpoint structure, replacing the original backbone weights. This merged checkpoint is then used as the starting point for Stage SFT (§3.3.3). In Approach 2, the Spoken LM is initialized from Qwen2.5-7B-Instruct at Stage 2 and proceeds to the same Stage SFT directly, without task-vector arithmetic. Apart from their Stage 2 initialization and the merging step, both approaches use the same speech-learning objectives, dialogue data, and Stage SFT procedure.

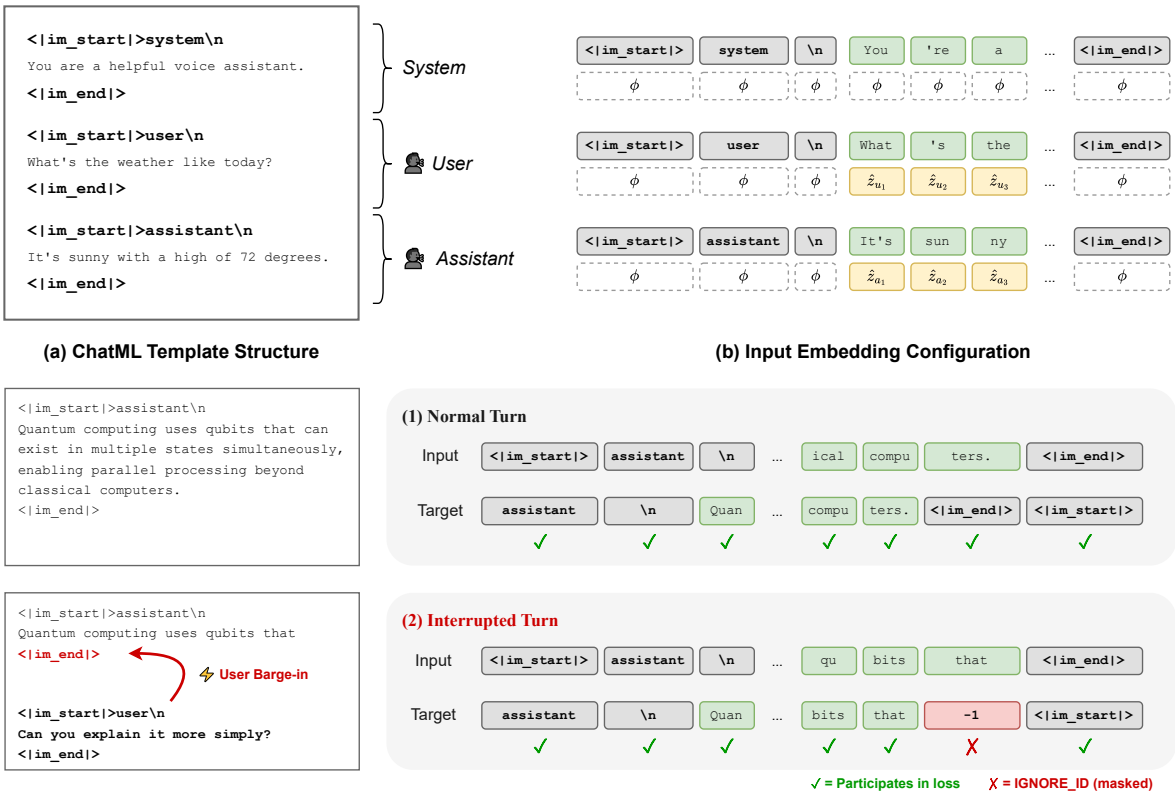
3.3.3 Dialogue Supervised Fine-Tuning

For both approaches, Stage SFT extends single-utterance generation to dialogue. A structured template and the existing autoregressive objective train response generation, turn-boundary estimation, and interruption-aware stopping without adding a separate turn classifier.

Template Design and Input Configuration. Figure 2 illustrates our dialogue template design, input embedding configuration, and interruption-aware training strategy. We adopt a ChatML-style dialogue template using special tokens <|im_start|> and <|im_end|> to delimit conversational segments. Each dialogue comprises three segment types: *System* (instruction prompt), *User* (user utterances), and *Assistant* (model responses). Segments are arranged sequentially without intervening newlines, with each segment following the pattern <|im_start|>*role*\n*content*<|im_end|> where *role* $\in$ {system, user, assistant}. During training, input embedding configuration varies by segment type: *System segments* contain text tokens with zero-filled audio latents because system prompts are non-voiced instructional content; *User segments* pair text tokens from ASR transcription with aligned audio latents extracted by the TASTE2 Speech Tokenizer; and *Assistant segments* pair text tokens with audio latents from ground-truth responses. The training objective applies unified next-token prediction across all segments and predicts both the next text token and its aligned audio latent.

Supervised Fine-Tuning Paradigm. Stage SFT uses structured dialogue templates and unified autoregressive prediction. Special tokens (<|im_start|>user, <|im_start|>assistant, etc.) mark the segment roles, and each dialogue sequence contains System, User, and Assistant segments. For the sequence [system] $\rightarrow$ [user] $\rightarrow$ <|im_end|> $\rightarrow$ <|im_start|>assistant, the training objective estimates $P(\text{assistant response} \mid \text{system prompt}, \text{user input})$. User segments contain both text tokens and audio latents.

Stage 2 and Stage SFT use the same unified autoregressive objective. In interruption examples, Stage SFT masks the terminal token of the interrupted Assistant segment. The loss is computed over both User and Assistant segments, and the resulting <|im_end|> distribution is used by the response-generation gate at inference time.



(c) Interruption-Aware <|im_end|> Token Masking (Text Tokens)

Figure 2 Dialogue template and interruption-aware training strategy. **(a)** ChatML segments for System, User, and Assistant, delimited by <|im_start|> and <|im_end|>. **(b)** One-to-one pairing of text tokens (green) and audio latents (yellow); special tokens use zero-filled audio embeddings (white, dashed). **(c)** Complete turns compute loss over all tokens, whereas interrupted Assistant turns set the loss weight of the terminal <|im_end|> token to 0 (marked ×).

Interruption-Aware Loss Design. Full-duplex dialogue training contains normal turn-taking and interruption examples. In normal examples, the Assistant response ends with <|im_end|> before the next User turn. In interruption examples, the Assistant utterance is truncated at the User barge-in position, and the loss weight of the terminal <|im_end|> token is set to 0.

The loss mask applies only to the terminal <|im_end|> token of an interrupted Assistant segment; all other content tokens in the segment retain their loss. Uninterrupted dialogues compute loss over all tokens, including <|im_end|>. At inference time, a user-speech-onset event stops generation.

Model-Gated Turn Decisions. A completed User segment is followed by <|im_end|> and <|im_start|>assistant, whereas an incomplete User segment is followed by another content token. In the deployed system, a VAD speech-end event triggers the turn gate. The gate checks whether <|im_end|> appears in the top-k predictions or exceeds a probability threshold (e.g., $p(\texttt{<|im\_end|>}) > 0.3$). If the gate condition is met, the model enters an Assistant segment; otherwise, it waits for additional user speech (§4.3.1).

Training Configuration. During Stage SFT, the Speech Tokenizer, Speech Detokenizer, CosyVoice2 flow-matching model, and vocoder remain frozen. The Spoken LM continues from the Stage 2 checkpoint and is fine-tuned with LoRA. Training data comprise three categories: normal multi-turn dialogues from DeepDialogue and synthetic general conversation data, synthetic interruption scenarios, and dialogues from the synthetic paralinguistic control data. Together they contain 1,493 hours of dialogue-annotated speech (466 hours from DeepDialogue and 1,027 hours from synthetic data).

4 TASTE2 VoiceBot: Incremental Deployment Architecture

TASTE2 VoiceBot deploys the model from §3 as an incremental pipeline. It processes user speech as it arrives, decides when to take the floor, streams its response, and stops when the user barges in. This section describes the services that realize these behaviors and the deployment decisions they require.

4.1 Overall System Architecture

TASTE2 VoiceBot employs a WebSocket-based microservice architecture comprising four main services (illustrated in Figure 3). Each service handles a distinct stage of the conversation pipeline, communicating asynchronously through WebSocket connections.

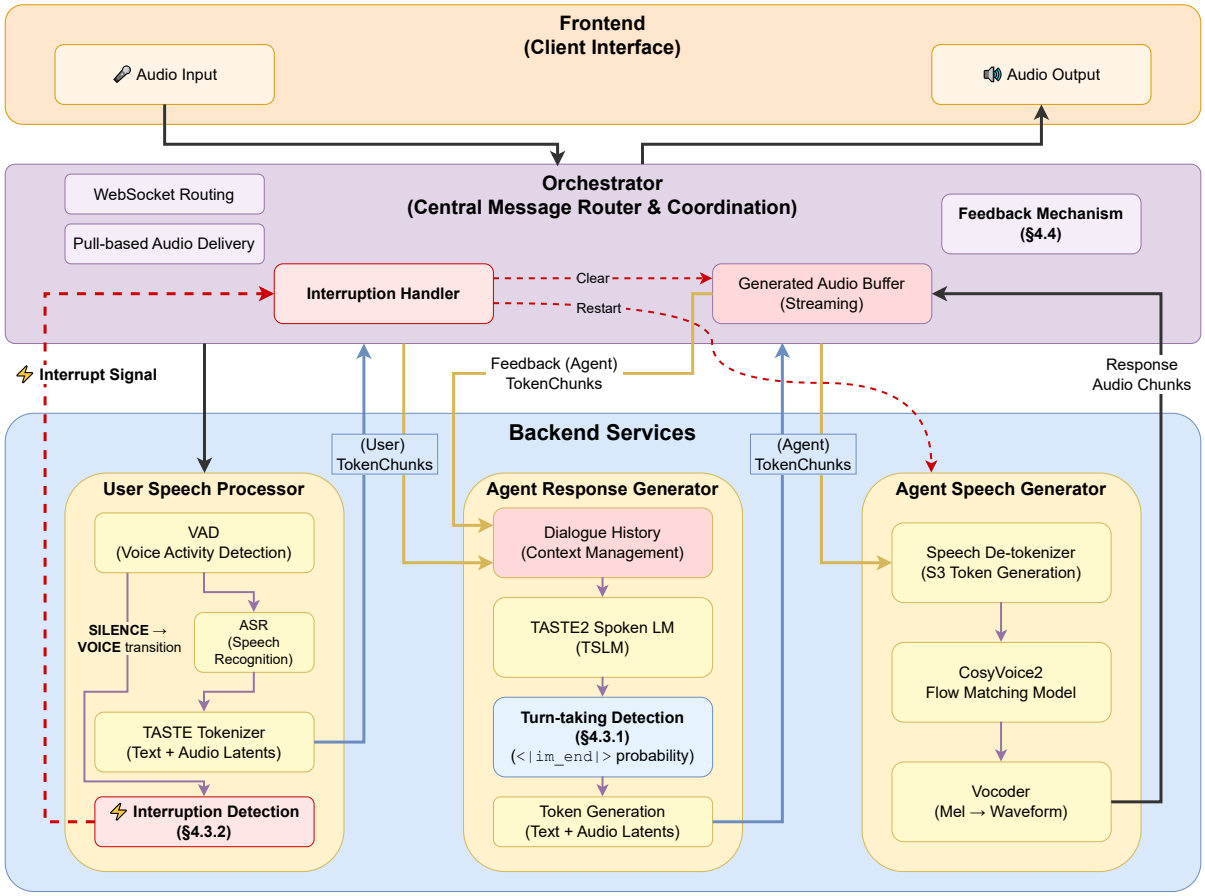


Figure 3 TASTE2 VoiceBot system architecture. Four WebSocket services form the dialogue pipeline: the **User Speech Processor** performs VAD, ASR, TASTE tokenization, and interruption detection (§4.3.2); the **Agent Response Generator** maintains dialogue context and applies model-gated turn decisions (§4.3.1); the **Agent Speech Generator** streams detokenized and vocoded speech; and the **Orchestrator** coordinates delivery, interruption handling, and playback-synchronized feedback (§4.4).

The **User Speech Processor** converts raw audio into aligned text tokens and audio latents. Voice activity detection (VAD) identifies speech and silence, automatic speech recognition (ASR) provides the transcription, and the TASTE2 Speech Tokenizer extracts the audio latents. A VAD transition from silence to voice during system playback emits a barge-in event before ASR completes.

The **Agent Response Generator** maintains dialogue context and runs the TASTE2 Spoken LM. It receives `TokenChunk` messages containing aligned text tokens and audio latents. After a VAD speech-end signal, it uses the probability of `<|im_end|>` to decide whether the user has completed a semantic turn (§4.3.1); only

then does it autoregressively predict response pairs. Feedback from the Orchestrator adds only delivered response pairs to the dialogue history.

The **Agent Speech Generator** maps response pairs to S3 units with the Speech Detokenizer, then uses the CosyVoice2 flow-matching model and vocoder to produce audio. It synthesizes complete input chunks incrementally and stops when it receives an interruption event (§4.2).

Orchestrator coordinates all services and manages client interactions. The service acts as a central message router, orchestrating communication between User Speech Processor, Agent Response Generator, and Agent Speech Generator. Crucially, the Orchestrator implements pull-based audio delivery and synchronized feedback mechanism (§4.4)—buffering audio from Agent Speech Generator, delivering chunks only upon client request, and feeding back corresponding agent tokens to Agent Response Generator only after delivery. This design ensures dialogue context remains synchronized with the user's actual auditory experience, even during interruptions.

4.2 Streaming Processing Pipeline

Full-duplex voice conversation is sensitive to accumulated response latency. Traditional pipelines process ASR, language model inference, and speech synthesis sequentially, with each stage waiting for the previous one to complete. TASTE2 therefore adopts a latency-oriented incremental pipeline.

TASTE2 addresses this challenge through a three-service concurrent streaming architecture. The pipeline comprises three services: (1) **User Speech Processor**: performs VAD, ASR, and TASTE tokenization, processing audio chunks incrementally and producing partial token representations; (2) **Agent Response Generator**: receives streaming user token chunks. Although the architecture can consume token chunks as they arrive, the deployed configuration invokes the turn-decision gate only when VAD detects speech end (voice → silence transition). The Spoken LM then either starts an Assistant segment or remains silent, avoiding repeated generation attempts while the user speaks; and (3) **Agent Speech Generator**: uses TASTE2's streaming output synthesis (§3.1.3) to convert agent tokens into S3 units and then produce audio waveforms incrementally through CosyVoice2's chunk-aware flow matching model and vocoder. The services operate independently and pass partial results downstream as they become available.

This streaming capability is realized through a decoupled worker architecture. The User Speech Processor employs a three-layer parallel processing architecture: (1) **VAD main loop** continuously runs voice activity detection on 100ms chunks; (2) **ASR worker** (running in a thread pool) asynchronously processes accumulated audio chunks, decoupled from the VAD main loop via `audio_chunk_queue`; and (3) **TASTE Tokenization worker** (in a separate thread pool) performs TASTE tokenization, receiving results from the ASR worker via `asr_result_queue`. This queue-based communication prevents blocking between stages—each stage processes immediately when input arrives, without waiting for downstream consumption. Similarly, the Agent Speech Generator employs a double-buffering architecture: a receiver thread continuously receives S3 units from the SLM (decoupling SLM inference), while the main thread executes GPU audio generation (token2wav). This design ensures GPU operations do not block the asynchronous event loop, maintaining streaming characteristics throughout the pipeline.

TASTE2 VoiceBot adapts the originally utterance-level Speech Tokenizer for incremental operation. Its causal decoder predicts the audio latent at position i from only the first i text tokens and the corresponding audio. VAD groups 100-ms audio chunks into 600-ms batches, after which the Speech Tokenizer extracts aligned text tokens and audio latents and sends them downstream in `TokenChunk` messages. TASTE2 VoiceBot can therefore accumulate dialogue context while the user speaks, although ordinary response generation remains gated by detected speech end.

4.3 Turn and Interruption Mechanisms

4.3.1 VAD-Triggered, Model-Gated Turn Decisions

TASTE2 VoiceBot separates the acoustic trigger from the semantic turn decision. A VAD voice-to-silence transition triggers evaluation, but does not by itself start an Assistant response. The Agent Response

Generator then applies the turn-boundary distribution learned by the TASTE2 Spoken LM as a semantic gate, without adding a separate turn-taking classifier.

During training (§3.3.3), the model learns implicit turn-end probability patterns through exposure to User segments. When a user finishes speaking, the dialogue sequence transitions to <|im_end|> followed by <|im_start|>assistant—the model learns to assign high probability to <|im_end|> after sentence-final tokens. Conversely, when a user is mid-utterance, the model assigns low probability to <|im_end|> and high probability to continuation tokens. Through this pattern, <|im_end|> probability implicitly encodes turn-end likelihood.

After the VAD trigger, the Agent Response Generator evaluates <|im_end|> after the accumulated user tokens. The system supports two gate conditions: (1) **Top-k detection**—checking whether <|im_end|> appears in the top-k predictions (e.g., top-800); and (2) **Probability threshold**—checking whether <|im_end|> probability exceeds a threshold (e.g., $p(\texttt{<|im\_end|>}) > 0.3$). If either condition identifies a turn end, the model enters the Assistant segment; otherwise it remains silent and awaits more user speech. Because the gate is evaluated only after VAD detects speech end, the deployed configuration never speaks during user speech; backchanneling is therefore measured as a model behavior in §5.3 rather than deployed as an active system behavior.

Punctuation marks (e.g., '.', ',', '¡) pose challenges for turn-taking detection—they may signal sentence endings or mid-sentence pauses. TASTE2 employs a lookahead mechanism to resolve this ambiguity: when the predicted token is punctuation, the system uses KV cache to predict the next token and applies <|im_end|> detection to the second token. This lookahead enables the system to distinguish sentence-final punctuation (e.g., '.' followed by high <|im_end|> probability) from mid-sentence punctuation (e.g., ',' followed by low <|im_end|> probability), improving turn-boundary detection accuracy in natural conversation.

4.3.2 Interruption Handling

TASTE2 VoiceBot handles user barge-in through early VAD detection and coordinated cleanup.

Interruption trigger: When the User Speech Processor's VAD detects a SILENCE → VOICE transition during agent speech, the system immediately emits an interruption event. This early signal (before ASR processing) enables rapid response to barge-in, minimizing latency. The interruption event propagates through a global `interruption_events` dictionary, using `asyncio.Event` for asynchronous notification across services.

Four-step cleanup flow: Once interruption is detected, the Orchestrator executes system cleanup: (1) **Close Agent Speech Generator WebSocket**—immediately halting audio generation, preventing further resource consumption; (2) **Clear audio buffer**—discarding all unplayed audio chunks, preventing stale content delivery to the client; (3) **Reset interruption event flag**—preparing the event state for the next potential interruption; and (4) **Reconnect Agent Speech Generator**—establishing a new WebSocket connection for the next turn. This cleanup flow ensures the system rapidly recovers from interruption, ready to process the user's new input.

4.4 Audio Playback and Feedback Mechanism

TASTE2 VoiceBot must keep its dialogue history consistent with audio playback. Dual-channel models such as Moshi (Défossez et al., 2024), NTPP (Zhang et al., 2025b), and SALM-Duplex (Hu et al., 2025) represent user and agent speech separately. TASTE2 VoiceBot instead tracks a single sequence of aligned text tokens and audio latents, while synthesis and playback proceed asynchronously.

Text-token–audio-latent pairs do not map individually to fixed-duration waveform segments. The Speech Detokenizer produces S3 units at 25 Hz for groups of input pairs, and playback boundaries therefore need not coincide with individual text-token boundaries. TASTE2 VoiceBot assigns a minimum number of input pairs to each audio chunk, establishing a coarse correspondence that can be tracked during delivery.

This architectural characteristic creates practical synchronization problems. Speech generation speed typically exceeds playback rate, enabling the system to rapidly produce complete responses. However, users may interrupt before playback completes—if all generated tokens are immediately fed back to dialogue context, the

system's internal state will contain agent utterances users never heard. This leads to dialogue incoherence—the agent may reference content users never heard, breaking conversational coherence.

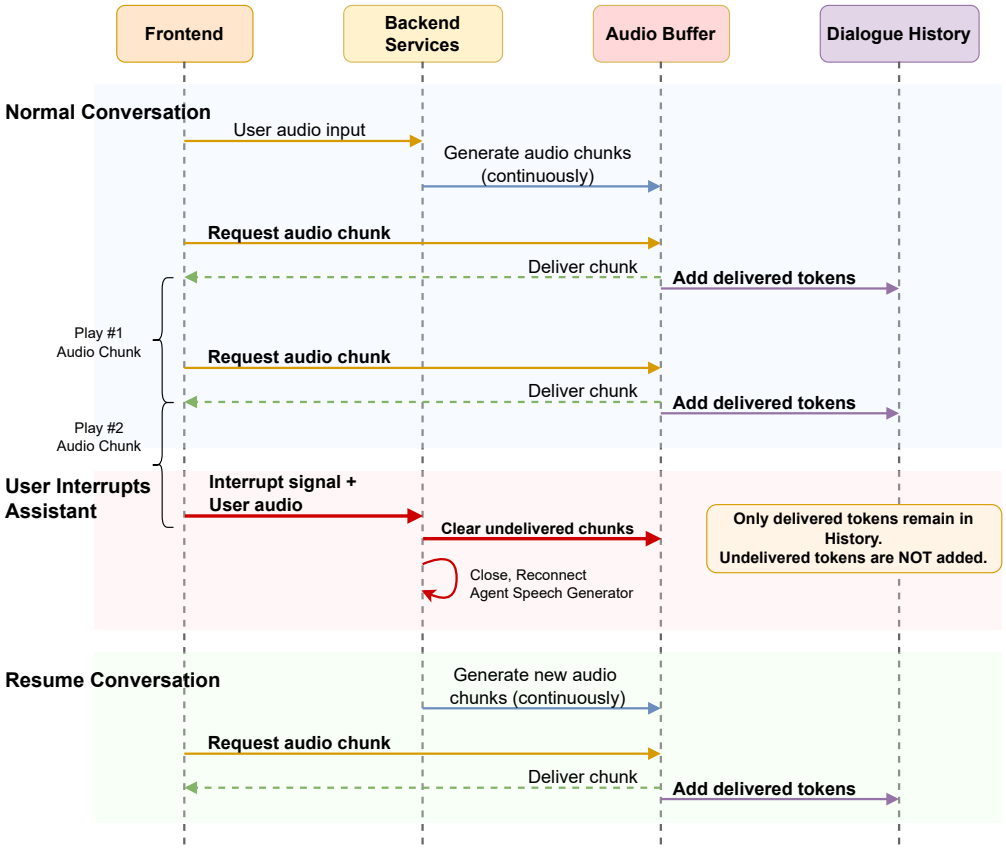


Figure 4 Pull-based audio delivery and synchronized feedback. The frontend requests chunks from the Audio Buffer, and the corresponding tokens enter Dialogue History only after delivery. Upon user interruption, the Orchestrator clears undelivered chunks and closes the Agent Speech Generator connection; tokens for those chunks are excluded from the dialogue context.

TASTE2 addresses this challenge through pull-based delivery with synchronized feedback (Figure 4). The mechanism comprises three design principles:

Pull-based audio delivery: The client sends a `type=request` message when ready for the next chunk (driven by playback). The Orchestrator serves from the audio buffer only upon request, tracking which chunks have been delivered to the client. This pull-based design ensures audio delivery synchronizes with the client's playback rate, preventing buffer overflow and enabling accurate delivery tracking.

Synchronized token feedback: Agent Speech Generator bundles `token_chunks` with corresponding `audio_data` in `AgentResponse` messages. The Orchestrator forwards agent tokens back to Agent Response Generator *only after delivering to the client*. This ensures dialogue context contains only what users actually received.

Interruption-safe design: When barge-in occurs, the Orchestrator clears undelivered audio buffer. Undelivered tokens never reach Agent Response Generator, maintaining context consistent with the user's actual experience. This design achieves a critical invariant: dialogue history precisely reflects the user's auditory experience, even during interruptions.

This synchronization mechanism yields three key advantages. First, **context accuracy**—dialogue history matches the user's auditory experience, preventing the agent from referencing unheard content. Second,

interruption coherence—the agent knows what was actually communicated, enabling post-interruption responses to continue appropriately. Third, **multi-turn consistency**—subsequent responses reference only heard content, maintaining coherence across the entire conversation.

5 Evaluation

We evaluate semantic preservation, full-duplex interaction patterns, deployed-system latency, and explicit paralinguistic control. Model evaluations cover the two implementation paths from §3.3: **TASTE2 (Merge)** for Approach 1 and **TASTE2 (Direct)** for Approach 2. They share training data, objectives, and dialogue fine-tuning, differing only in how instruction-following capability is introduced.

5.1 Semantic Preservation Analysis

We first measure how much language-understanding accuracy remains after adapting a text language model into the TASTE2 Spoken LM and characterize that retention under both implementation paths.

Benchmark Setup. We adopt LLaMA-Questions (LLaMA-Q.) (Nachmani et al., 2024) as our Spoken QA benchmark, a question answering benchmark specifically designed for evaluating speech-text joint models. Following the evaluation protocol of the TASTE paper (Tseng et al., 2025) and Moshi (Défossez et al., 2024), we employ answer containment accuracy as the evaluation metric—checking whether the generated response contains the correct answer. To quantify semantic degradation, we compute the degradation metric:

$$\text{Degradation (\%)} = \frac{\text{LLM}_{\text{Accuracy}} - \text{SpokenLM}_{\text{Accuracy}}}{\text{LLM}_{\text{Accuracy}}} \times 100 \tag{1}$$

where negative degradation indicates an improvement over the text reference; lower values indicate better preservation.

Experimental Configuration. We evaluate both TASTE2 (Merge) and TASTE2 (Direct) on 300 questions from LLaMA-Questions after Stage SFT. Each spoken question is provided as audio, and the model jointly generates text tokens and audio latents. Accuracy is computed from the generated text response using normalized, case-insensitive answer containment. The Spoken LM evaluation is zero-shot (no in-context examples at inference), and responses are generated with greedy decoding. The text-only reference is the Qwen2.5-7B base model under few-shot prompting, which reaches 64.3% accuracy (Table 2).

Table 2 Semantic degradation on the LLaMA-Questions benchmark. T and S denote text and speech, respectively. Rows marked † use few-shot prompting for both the spoken system and its text baseline. TASTE2 is evaluated zero-shot after dialogue SFT against a few-shot Qwen2.5-7B base text reference, so its degradation is not directly comparable to the † results.

System	Mode	LLaMA-Q. Accuracy	Degradation
Mini-Omni 0.5B(T→T)	T	39.0%	-
Mini-Omni 0.5B	T+S	11.6%	70.3%
Helium 7B (text)	T	75.0%	-
Moshi 7B	T+S	62.3%	16.9%
LLaMA3.1-8B-Instruct	T	71.7%	-
Llama-Omni 8B	T+S	67.3%	6.1%
LLaMA3.2-1B†	T	51.0%	-
TASLM 1B†	T+S	57.6%	**-12.9%** (improvement)
Qwen2.5-7B base†	T	64.3%	-
TASTE2 (Merge)	T+S	56.3%	12.4%
TASTE2 (Direct)	T+S	53.0%	17.6%

Results and Analysis. TASTE2 (Merge) preserves more semantic accuracy than TASTE2 (Direct). Two implications of this gap matter, both subject to the caveat that the other rows of Table 2 come from published setups with different backbones and prompting protocols.

The first concerns what the language model is asked to carry. The systems in Table 2 differ in how much acoustic responsibility sits inside the language model itself. Mini-Omni decodes text and audio in parallel streams, while Moshi predicts time-aligned text before audio across multiple codebook streams. The TASTE2 Spoken LM also emits acoustic information at every step, but its sequence is exactly as long as the text sequence, since one continuous latent travels alongside each text token instead of occupying positions of its own. Under that constraint the degradation is 12.4%. Llama-Omni retains more, yet its language model emits text only and delegates acoustics to a downstream module, so its semantic accuracy is protected by construction rather than preserved under acoustic load. Text-aligned modeling thus buys per-token acoustic generation without the sequence-length inflation that accompanies interleaved designs, and without the degradation such designs report.

The second reading locates part of the remaining loss outside speech modeling. Merge and Direct share tokenizer, speech objectives, dialogue data, and Stage SFT; they differ only in how instruction-following capability is introduced. That difference accounts for 29% of Direct's loss against the reference. A substantial share of the measured degradation is therefore attributable to instruction tuning rather than to representing speech, and is recoverable by changing the merge recipe alone.

5.2 Residual VQ Ablation

The Spoken LM predicts continuous audio latents rather than RVQ indices, making the need for quantization unclear. We therefore test whether RVQ acts as a useful training constraint.

Experimental Setup. We compare two training configurations: (1) **Without RVQ**—completely removing RVQ and directly training continuous latents; (2) **With RVQ (TASTE2)**—retaining RVQ as a discrete bottleneck. Both configurations use identical training data, hyperparameters, and model architecture (except for RVQ presence). The training proceeds in two stages: Stage 1 separately trains each configuration's tokenizer and detokenizer (4 epochs); Stage 2 uses the respective tokenizers trained from Stage 1 to train the spoken language model (50k steps). We evaluate: Stage 1 speech reconstruction accuracy (S3 accuracy); Stage 2 (1) latent space convergence measured by TASTE latent L1 distance and (2) language modeling performance measured by text accuracy.

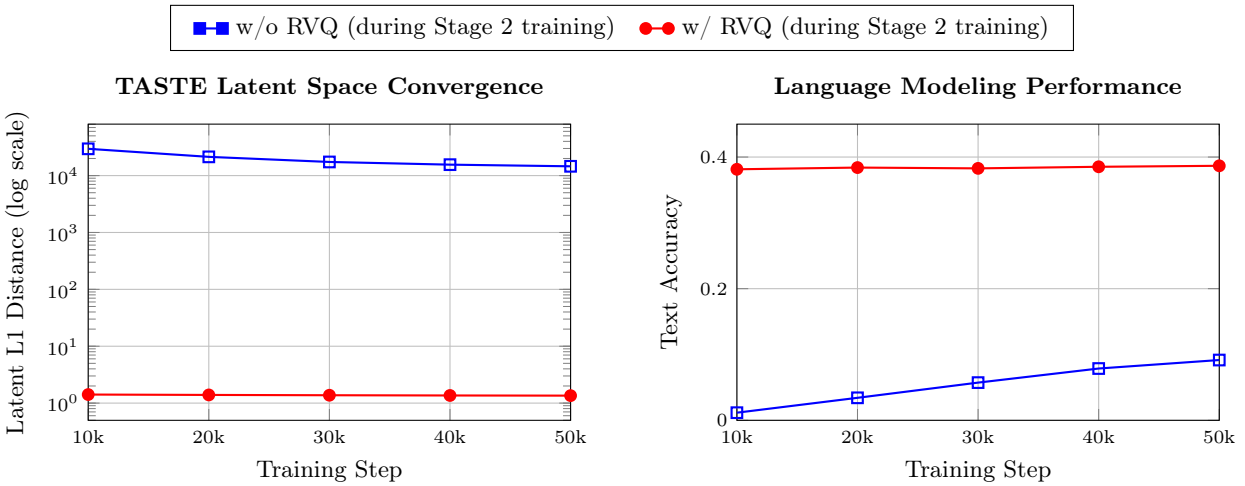


Figure 5 Effect of residual vector quantization (RVQ) during two-stage training. Removing RVQ yields higher Stage 1 S3 reconstruction accuracy after four epochs (46.83% versus 23.56%) but substantially different Stage 2 behavior over 50k steps. **Left:** Latent L1 distance decreases to 14,573 without RVQ and to 1.35 with RVQ. **Right:** Text accuracy reaches 9.16% without RVQ and 38.67% with RVQ.

Results and Analysis. The two stages want opposite things from the representation. Removing RVQ

improves Stage 1 reconstruction because an unconstrained continuous space is free to encode whatever the detokenizer needs. That same freedom makes the space unlearnable in Stage 2: latent error remains four orders of magnitude higher, and text accuracy falls to roughly one quarter of the RVQ model's. What fails is not the acoustic branch alone. Because text tokens and audio latents are predicted in the same forward pass, an unreachable regression target drags the text objective down with it.

RVQ therefore does not make the representation better; it makes it predictable. This is a precondition for the entire TASTE2 design rather than an incidental hyperparameter: a language model can be asked to emit continuous acoustic latents only when quantization has already made the target space finite. The ablation does not isolate which property of RVQ—codebook size, residual depth, or the discretization itself—supplies that constraint, so it establishes the requirement without explaining its mechanism.

5.3 Full-Duplex Interaction Evaluation

Full-Duplex-Bench (Lin et al., 2025b) evaluates whether the TASTE2 Spoken LM exhibits four interaction patterns: pause handling, backchanneling, smooth turn transitions, and user-interruption handling. These model-level tests are distinct from the deployed TASTE2 VoiceBot behavior described in §4.

Benchmark Specification. We use Full-Duplex-Bench v1.0, which assesses four interactive behaviors: (1) *Pause Handling*: whether the model waits during user pauses rather than falsely starting to speak; (2) *Backchannel*: whether the model provides brief listener feedback (e.g., "mm-hmm", "I see") during user speech; (3) *Smooth Turn Taking*: whether the model detects user turn-end and starts responding with low latency; and (4) *User Interruption*: whether the model stops and adapts when a user barges in mid-response. Pause handling is evaluated on both synthetic cases (Synth) and natural conversations drawn from the CANDOR corpus; smooth turn taking uses the CANDOR cases. The benchmark employs automatic metrics for reproducible evaluation.

Evaluation Metrics. We adopt the metrics defined by Full-Duplex-Bench (Lin et al., 2025b). **TOR (Turn-taking Overlap Rate)** measures turn-transition behavior. For Pause Handling and Backchannel, lower TOR is better: it captures the model incorrectly claiming the turn during a user pause or in a context that calls for only a brief backchannel. This differs from **Backchannel Frequency**, which measures how often the model produces backchannels; higher is better. For Smooth Turn Taking and User Interruption, higher TOR is better because it reflects successful detection of turn-end or interruption. **Latency** is measured in seconds; lower is better. **GPT-4o Score** evaluates the coherence and contextual appropriateness of responses after interruptions on a 1–5 scale.

Table 3 Full-Duplex-Bench v1.0 results on 727 samples across five tasks. TOR denotes turn-taking overlap rate: lower is better for pause handling and backchannel, whereas higher is better for smooth turn taking and user interruption. Lower latency and higher GPT-4o scores are better. Bold denotes the best value in each column across all systems, including ties.

Model	Pause Handling		Backchannel			Smooth Turn Taking		User Interruption		
	Synth TOR ↓	CANDOR TOR ↓	TOR ↓	Freq ↑	JSD ↓	CANDOR TOR ↑	Latency (s) ↓	TOR ↑	GPT-4o ↑	Latency (s) ↓
dGSLM	0.934	0.935	0.691	0.015	0.934	**0.975**	0.352	0.917	0.201	2.531
Moshi	0.985	0.980	1.000	0.001	0.957	0.941	**0.265**	**1.000**	0.765	0.257
Freeze-Omni	0.642	0.481	0.636	0.001	0.997	0.336	0.953	0.867	**3.615**	1.409
Gemini Live	0.255	**0.310**	**0.091**	0.012	0.896	0.655	1.301	0.891	3.376	1.183
TASTE2 (Merge)	**0.146**	0.389	0.491	**0.044**	**0.817**	0.908	1.207	**1.000**	3.275	**0.060**
TASTE2 (Direct)	0.161	0.319	0.164	0.018	0.942	0.840	1.239	0.971	2.856	0.134

Results and Analysis. Table 3 reveals a latency–coherence trade-off. dGSLM and Moshi react quickly but fail to produce useful post-interruption content, whereas Freeze-Omni and Gemini Live remain coherent but react slowly. TASTE2 (Merge) belongs to neither extreme: it stops on every interruption, reacts in 0.060 s, and still scores 3.275 afterwards. Text-aligned modeling makes this combination possible because the dialogue state that survives an interruption is a text-token sequence that can be truncated exactly at the delivered boundary (§4.4).

The backchannel columns must be read jointly: a low overlap rate is inexpensive for a system that seldom

speaks during user speech. TASTE2 (Merge) backchannels most often and matches human timing most closely, paying for that behavior with more overlap; Direct occupies the opposite corner. The two strategies therefore bracket a trade-off between speaking often enough to be a listener and staying out of the user's way, rather than differing in quality. Pause handling shows the same bracketing: Merge leads on synthetic pauses, while Direct is the stronger TASTE2 variant on natural CANDOR conversations and remains near the best displayed system.

One weakness is unambiguous: both TASTE2 variants take over one second to make a smooth turn transition. This is the direct cost of the design. The turn decision waits for VAD speech end and then queries the <|im_end|> distribution (§4.3.1), which buys accurate turn identification without a dedicated classifier and pays for it in latency. Across metrics, neither instruction-tuning strategy dominates the other, so the full-duplex conclusion does not rest on a single IF initialization; these comparisons remain descriptive, since deployment conditions differ across published systems and the commercial one is closed.

5.4 System Latency Analysis

We measure TASTE2 VoiceBot latency using **time to first audio (TTFA)**, the interval from the end of user speech to the onset of system speech. We test one sample at each of five input lengths: 4, 11, 25, 48, and 60 seconds. For each interaction, we locate the silence-to-voice transition in the generated waveform and verify it manually. TASTE2 VoiceBot runs with batch size 1 on a single NVIDIA RTX A6000. ChatGPT Voice, Gemini Live, and Grok were measured through their official mobile applications on March 13, 2026; their hardware configurations are unknown.

Table 4 presents TTFA measurements across systems. Because these five single-sample measurements show no consistent trend with input length, we summarize each system using its average TTFA. TensorRT reduces TASTE2's average TTFA by 22%, but TASTE2 still does not match ChatGPT Voice on this measure. Profiling identifies the Agent Speech Generator as the primary end-to-end latency bottleneck, making a lighter synthesis architecture the most direct optimization target.

Table 4 Time to first audio (TTFA; ms) across input speech lengths; lower is better. ChatGPT Voice, Gemini Live, and Grok were measured using their official mobile apps on March 13, 2026. TensorRT (TRT) reduces TASTE2's average TTFA from 3,469 to 2,701 ms (22%).

System	**TTFA (ms)**					
Input Length	*4s*	*11s*	*25s*	*48s*	*60s*	**Avg.**
ChatGPT Voice	1056	1056	1120	1376	2240	1370
Gemini Live	4160	3968	3840	4800	3392	4032
Grok	2016	3424	1984	1952	2176	2310
TASTE2 w/o TRT	3392	3232	3008	4000	3712	3469
TASTE2	2464	2336	2912	3424	2368	2701 (-22%)

5.5 Paralinguistic Control

Setup. We use two human-rated diagnostics. The primary one is a 41-item subset of VStyle's `acoustic_attributes` (Song et al., 2025), which tests instruction-conditioned explicit control on both post-SFT models: **Speed** in full (10 `fast` and 10 `slow` items), one **Volume** item (`whisper`), and five **Emotion** classes with 4 items each (`angry`, `fear`, `happy`, `sad`, `surprise`). Because its coverage differs from the full VStyle protocol, these scores must not be compared with VStyle's published cross-system results. The supporting diagnostic asks whether the Spoken LM can retain an acoustic feature already present in a speech opening: using the Approach 1 checkpoint before model merging and dialogue SFT, we synthesize five unfinished English openings for each of the same eight features and let the model continue the aligned text–audio sequence. In both diagnostics, three human raters independently score each successful generation from 1 (completely inappropriate) to 5 (perfectly appropriate); means pool over raters and successful generations, and generations without valid audio are excluded.

Table 5 Human-rated diagnostics of paralinguistic feature continuation before dialogue SFT and explicit style control after dialogue SFT. **Stage 2 feature continuation** uses five synthesized speech openings per feature with the Approach 1 Spoken LM before model merging; raters assess whether each newly generated continuation retains the opening's feature. **VStyle explicit control** evaluates the 41-item subset described in §5.5 on both post-SFT models. Three raters score outputs from 1 to 5; Avg ± Std pools ratings over raters and successful generations. Invalid audio generations are counted as unsuccessful and excluded from the score statistics, and means below 3 are gray. Scores across the two diagnostics are not directly comparable because their tasks and sample sizes differ.

	Stage 2 Feature Continuation		**VStyle Explicit Control**			
			TASTE2 (Merge)		**TASTE2 (Direct)**	
Feature	Succ./Total	Avg ± Std	Succ./Total	Avg ± Std	Succ./Total	Avg ± Std
fast	5/5	4.87 ± 0.35	9/10	4.15 ± 0.95	10/10	3.53 ± 1.36
slow	5/5	4.27 ± 0.96	9/10	3.00 ± 1.21	10/10	3.77 ± 1.25
happy	5/5	3.80 ± 0.94	4/4	1.67 ± 0.98	3/4	1.67 ± 1.00
sad	3/5	3.56 ± 1.24	4/4	2.00 ± 1.35	3/4	1.89 ± 1.05
surprise	5/5	3.33 ± 1.11	4/4	2.42 ± 1.44	4/4	3.42 ± 1.56
whisper	4/5	3.27 ± 1.19	1/1	1.00 ± 0.00	1/1	1.00 ± 0.00
fear	5/5	2.40 ± 1.24	4/4	1.17 ± 0.58	4/4	1.33 ± 0.65
angry	2/5	2.00 ± 1.26	4/4	1.33 ± 0.65	4/4	3.33 ± 0.89

Results and Analysis. Each post-SFT TASTE2 model yields valid audio for 39 of the 41 VStyle items, so the main limitation is control rather than generation. `fast` is the only feature above 3 in both strategies and also clears that threshold in the Stage 2 continuation probe, providing a cross-strategy and cross-stage proof of concept. Direct alone clears 3 on `angry` and `surprise`, showing that emotion control is possible but strategy-dependent. The remaining attributes are weak or inconsistent, so explicit control does not generalize across attributes.

These patterns separate representational possibility from robust control. A text-aligned representation can carry a paralinguistic attribute through Stage 2 speech pretraining and still respond to a textual instruction after dialogue SFT, but that success does not yet transfer reliably across attributes or training strategies. The Stage 2 probe suggests that dialogue SFT is not the sole cause: `whisper` can be continued before SFT but not produced on request afterwards, while `fear` and `angry` are already weak in the earlier probe. With only five openings per feature on a single checkpoint, however, this is an observation rather than a diagnosis. Implicit paralinguistic adaptation remains outside the present scope.

5.6 Summary of Findings

Taken together, the four evaluations answer the question raised in §1—whether text-aligned speech tokenization is a viable modeling path toward full-duplex voice interaction—and they answer it with a claim narrower, and more useful, than "TASTE2 performs well."

What the representation costs. A language model that must also produce speech normally pays for it in sequence length. Acoustic tokens occupy positions of their own, and the regularities learned during text pretraining are diluted across a longer, mixed sequence; the systems in Table 2 give up between 16.9% and 70.3% of their text backbones' accuracy in proportion to how much room those acoustic positions take. TASTE2 does not make that trade. One continuous audio latent travels alongside each text token, so the model emits acoustic information at every step while its sequence stays exactly as long as the text sequence. The measured cost is 12.4%, and part of even that figure has nothing to do with speech: 3.3 of the 11.3 points the weaker strategy gives up follow from how instruction-following capability is introduced, and task-vector merging recovers them without touching the speech objectives.

Where it pays off. The benefit is not diffuse. It appears exactly where full-duplex behavior depends on the state the model carries across a turn boundary. Interruption is the clearest case: TASTE2 stops on every barge-in, reacts in 0.060 s, and still produces a coherent response afterwards (GPT-4o 3.275). Among the displayed systems this combination is unique—those that react as fast score below 0.8 on post-interruption coherence,

and those that answer as coherently need over a second to react. The mechanism is the representation itself. What survives an interruption is a sequence of text tokens, so the dialogue history can be truncated exactly at the delivered boundary and the model resumes from a state that matches what the user actually heard. Turn identification benefits for the same reason: a turn boundary is an ordinary token in that sequence, which is why TOR reaches 0.908 with no turn-taking classifier anywhere in the system.

What it does not yet deliver. The evaluations bound the claim as concretely as they support it. Routing turn decisions through the model's own distribution buys accuracy and costs time: smooth-turn latency is 1.207 s, against 0.265 s for the fastest displayed system. Deployed TTFA remains 2.701 s, with profiling placing the bottleneck in the Agent Speech Generator rather than in the Spoken LM. Explicit paralinguistic control is not reliable across attributes or training strategies: `fast` is the only feature above 3 in both post-SFT models, while Direct alone exceeds 3 on `angry` and `surprise`.

6 Conclusion

TASTE2 establishes TASTE-based text-aligned speech modeling and deployment as a viable path toward full-duplex voice interaction. Reaching this point required an end-to-end adaptation of the original utterance-level method: a shared text-token vocabulary removes word-level averaging, one continuous audio latent rides alongside each text token so that the language model emits acoustic information at every step without a sequence longer than the text itself, interruption-aware dialogue training learns when to stop and when to take a turn, and the Speech Detokenizer produces S3 units incrementally. TASTE2 VoiceBot places this model in a working research deployment with VAD-triggered and model-gated turn decisions, streaming synthesis, barge-in handling, and playback-synchronized context feedback whose dialogue history excludes response content discarded before delivery.

The evaluations (§5.6) support a specific claim. Under the stronger instruction-tuning strategy the Spoken LM retains 87.6% of the text reference's benchmark accuracy without lengthening its sequence, and it delivers the full-duplex behavior that most directly depends on that representation: TASTE2 is the only displayed system that both stops immediately on a barge-in and remains coherent afterwards. Speaking rate further shows that explicit paralinguistic control is attainable in this architecture. The limits are equally specific. Turn taking is accurate but takes over a second, deployed mean TTFA remains 2.701 s with the bottleneck in synthesis, and explicit control remains inconsistent across attributes and training strategies: `fast` is the only feature above 3 in both post-SFT models, while Direct alone exceeds 3 on `angry` and `surprise`.

References

Anonymous. ProsodyLM: Uncovering the Emerging Prosody Processing Capabilities in Speech Language Models. *arXiv preprint arXiv:2507.20091*, 2025. URL https://arxiv.org/abs/2507.20091.

Siddhant Arora, Zhiyun Lu, Chung-Cheng Chiu, Ruoming Pang, and Shinji Watanabe. Talking Turns: Benchmarking Audio Foundation Models on Turn-Taking Dynamics. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2503.01174.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. AudioLM: A Language Modeling Approach to Audio Generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533, 2023. URL https://arxiv.org/abs/2209.03143.

Yuxuan Chen and Haoyuan Yu. From Turn-Taking to Synchronous Dialogue: A Survey of Full-Duplex Spoken Language Models. *arXiv preprint arXiv:2509.14515*, 2025. URL https://arxiv.org/abs/2509.14515.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-Audio: Advancing Universal Audio Understanding via Unified Large-Scale Audio-Language Models. *arXiv preprint arXiv:2311.07919*, 2023. URL https://arxiv.org/abs/2311.07919.

Wenqian Cui, Lei Zhu, Xiaohui Li, Zhihan Guo, Haoli Bai, Lu Hou, and Irwin King. Think Before You Talk: Enhancing Meaningful Dialogue Generation in Full-Duplex Speech Language Models with Planning-Inspired Text Guidance. *arXiv preprint arXiv:2508.07375*, 2025. URL https://arxiv.org/abs/2508.07375.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High Fidelity Neural Audio Compression. *Transactions on Machine Learning Research (TMLR)*, 2023. URL https://arxiv.org/abs/2210.13438.

Zhihao Du et al. CosyVoice: A Scalable Multilingual Zero-shot Text-to-speech Synthesizer based on Supervised Semantic Tokens. *arXiv preprint arXiv:2407.05407*, 2024a. URL https://arxiv.org/abs/2407.05407.

Zhihao Du et al. CosyVoice 2: Scalable Streaming Speech Synthesis with Large Language Models. *arXiv preprint arXiv:2412.10117*, 2024b. URL https://arxiv.org/abs/2412.10117.

Alexandre Défossez, Gabriel Synnaeve, Yossi Adi, Jade Copet, Antonia Creswell, Itai Gat, et al. Moshi: A Speech-Text Foundation Model for Real-Time Dialogue. *arXiv preprint arXiv:2410.00037*, 2024. URL https://arxiv.org/abs/2410.00037.

Erik Ekstedt and Gabriel Skantze. TurnGPT: A Transformer-based Language Model for Predicting Turn-taking in Spoken Dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2981–2990, 2020. URL https://aclanthology.org/2020.findings-emnlp.268/.

Erik Ekstedt and Gabriel Skantze. Voice Activity Projection: Self-supervised Learning of Turn-taking Events. In *Proceedings of Interspeech*, 2022. URL https://arxiv.org/abs/2205.09812.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. LLaMA-Omni: Seamless Speech Interaction with Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://arxiv.org/abs/2409.06666.

Qingkai Fang et al. LLaMA-Omni 2: LLM-based Real-time Spoken Chatbot with Autoregressive Streaming Speech Synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025b. URL https://aclanthology.org/2025.acl-long.912/.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. Emilia: An Extensive, Multilingual, and Diverse Speech Dataset for Large-Scale Speech Generation. In *IEEE Spoken Language Technology Workshop (SLT)*, 2024. URL https://arxiv.org/abs/2407.05361.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. URL https://arxiv.org/abs/2106.07447.

Zhaohan Hu et al. SALM-Duplex: Efficient and Direct Duplex Modeling for Speech-to-Speech Language Model. In *Proceedings of Interspeech*, 2025. URL https://arxiv.org/abs/2505.15670.

Shengpeng Ji, Yifu Chen, Minghui Fang, et al. WavChat: A Survey of Spoken Dialogue Models. *arXiv preprint arXiv:2411.13577*, 2024. URL https://arxiv.org/abs/2411.13577.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, et al. On Generative Spoken Language Modeling from Raw Audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354, 2021.

Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Yossi Adi, Juan Pino, Jiatao Gu, et al. Textless Speech-to-Speech Translation on Real Data. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 860–872, 2022.

Tianpeng Li et al. Baichuan-Audio: A Unified Framework for End-to-End Speech Interaction. *arXiv preprint arXiv:2502.17239*, 2025a. URL https://arxiv.org/abs/2502.17239.

Zehan Li, Hongjie Chen, Yuxin Zhang, Jing Zhou, Xuening Wang, Hang Lv, Mengjie Du, Yaodong Song, Jie Lian, Jian Kang, Jie Li, Yongxiang Li, Zhongjiang He, and Xuelong Li. TELEVAL: A Dynamic Benchmark Designed for Spoken Language Models in Chinese Interactive Scenarios. *arXiv preprint arXiv:2507.18061*, 2025b. URL https://arxiv.org/abs/2507.18061.

Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, et al. Full-Duplex-Bench v1.5: Evaluating Overlap Handling for Full-Duplex Speech Models. *arXiv preprint arXiv:2507.23159*, 2025a. URL https://arxiv.org/abs/2507.23159.

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. *arXiv preprint arXiv:2503.04721*, 2025b. URL https://arxiv.org/abs/2503.04721.

Jingwen Liu et al. EMO-Reasoning: Benchmarking Emotional Reasoning Capabilities in Spoken Dialogue Systems. *arXiv preprint arXiv:2508.17623*, 2025. URL https://arxiv.org/abs/2508.17623.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. Spoken Question Answering and Speech Continuation Using Spectrogram-Powered LLM. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2305.15255.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, et al. Spirit-lm: Interleaved Spoken and Written Language Model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025.

Yizhou Peng, Yi-Wen Chao, Dianwen Ng, Yukun Ma, Chongjia Ni, Bin Ma, and Eng Siong Chng. FD-Bench: A Full-Duplex Benchmarking Pipeline Designed for Full Duplex Spoken Dialogue Systems. In *Proceedings of Interspeech*, 2025. URL https://arxiv.org/abs/2507.19040.

Paul K. Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, et al. AudioPaLM: A Large Language Model That Can Speak and Listen. *arXiv preprint arXiv:2306.12925*, 2023. URL https://arxiv.org/abs/2306.12925.

Jun Song et al. VStyle: A Benchmark for Voice Style Adaptation with Spoken Instructions. *arXiv preprint arXiv:2509.09716*, 2025. URL https://arxiv.org/abs/2509.09716.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. SALMONN: Towards Generic Hearing Abilities for Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2024a. URL https://arxiv.org/abs/2310.13289.

Changli Tang, Wenyi Yu, Guangzhi Sun, et al. SALMONN-omni: A Codec-free LLM for Full-duplex Speech Understanding and Generation. *arXiv preprint arXiv:2411.18138*, 2024b. URL https://arxiv.org/abs/2411.18138.

Moonshot AI Team. Kimi-Audio Technical Report. *arXiv preprint arXiv:2504.18425*, 2025. URL https://arxiv.org/abs/2504.18425.

Liang-Hsuan Tseng, Yi-Chang Chen, Kuan-Yi Lee, Da-Shan Shiu, and Hung-yi Lee. TASTE: Text-Aligned Speech Tokenization and Embedding for Spoken Language Modeling. *arXiv preprint arXiv:2504.07053*, 2025. URL https://arxiv.org/abs/2504.07053.

Bandhav Veluri, Benjamin N Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. Beyond Turn-Based Interfaces: Synchronous LLMs as Full-Duplex Dialogue Agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 21390–21402, 2024. URL https://aclanthology.org/2024.emnlp-main.1192/.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy F. Chen. AudioBench: A Universal Benchmark for Audio Large Language Models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4297–4316, 2025a. URL https://aclanthology.org/2025.naacl-long.218/.

Haoyang Wang et al. ParaS2S: Benchmarking and Aligning Spoken Language Models for Paralinguistic-aware Speech-to-Speech Interaction. *arXiv preprint arXiv:2511.08723*, 2025b. URL https://arxiv.org/abs/2511.08723.

Xiong Wang, Yangze Chen, Chaoyou Li, et al. Freeze-Omni: A Smart and Low Latency Speech-to-speech Dialogue Model with Frozen LLM. *arXiv preprint arXiv:2411.00774*, 2024. URL https://arxiv.org/abs/2411.00774.

Zhifei Xie and Changqiao Wu. Mini-Omni: Language Models Can Hear, Talk While Thinking in Streaming. *arXiv preprint arXiv:2408.16725*, 2024. URL https://arxiv.org/abs/2408.16725.

Jin Xu et al. Qwen2.5-Omni Technical Report. *arXiv preprint arXiv:2503.20215*, 2025. URL https://arxiv.org/abs/2503.20215.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving Interference When Merging Models. In *Advances in Neural Information Processing Systems*, 2023.

Matthew Yue et al. VoiceBench: Benchmarking LLM-Based Voice Assistants. *arXiv preprint arXiv:2410.17196*, 2024. URL https://arxiv.org/abs/2410.17196.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An End-to-End Neural Audio Codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022. URL https://arxiv.org/abs/2107.03312.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *Proceedings of Interspeech*, pages 1526–1530, 2019. URL https://arxiv.org/abs/1904.02882.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. GLM-4-Voice: Towards Intelligent and Human-Like End-to-End Spoken Chatbot. *arXiv preprint arXiv:2412.02612*, 2024. URL https://arxiv.org/abs/2412.02612.

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 *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. URL https://arxiv.org/abs/2305.11000.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, and Chaohong Tan. OmniFlatten: An End-to-end GPT Model for Seamless Voice Conversation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025a. URL https://arxiv.org/abs/2410.17799.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. SpeechTokenizer: Unified Speech Tokenizer for Speech Language Models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2308.16692.

Yihan Zhang et al. NTPP: Generative Speech Language Modeling for Dual-Channel Spoken Dialogue via Next-Token-Pair Prediction. In *International Conference on Machine Learning (ICML)*, 2025b. URL https://arxiv.org/abs/2506.00975.