F-Actor: Controllable Conversational Behavior in Full-Duplex Models

Anonymous ACL submission

Abstract

Spoken conversational systems require more than accurate speech generation to have human-like conversations: to feel natural and engaging, they must produce conversational behaviour that adapts dynamically to the context. Current spoken conversational systems, however, rarely allow such customization, limiting their naturalness and usability. In this work, we present the first open, instruction-following full-duplex conversational speech model that can be trained efficiently under typical academic resource constraints. By keeping the audio encoder frozen and finetuning only the language model, our model requires just 2,000 hours of data, without relying on large-scale pretraining or multi-stage optimization. The model can follow explicit instructions to control speaker voice, conversation topic, conversational behaviour (e.g., backchanneling and interruptions), and dialogue initiation. We propose a single-stage training protocol and systematically analyze design choices. Both the model and training code will be released to enable reproducible research on controllable full-duplex speech systems.

1 Introduction

Developing a machine that can interact with humans in a natural, conversational way has been a research goal since the Dartmouth proposal in 1955 (McCarthy et al., 2006). Although today's conversational systems in text-to-text settings are approaching human-like communication (Cheng et al., 2024), their speech counterparts continue to exhibit significant limitations.

A key property of natural human conversation is its full-duplex nature (Stivers et al., 2009): humans can listen and speak at the same time. This facilitates natural turn-taking (Raux and Eskenazi, 2012) by enabling interruptions and backchannels, such as brief acknowledgments or sounds of agreement produced while the other speaker continues talking. For conversational speech systems to behave in a human-like manner, they should therefore be able to handle overlapping speech (Schegloff, 1982; Cho et al., 2022) and be able to dynamically adapt their behaviour in real time.

Recent work has begun to address full-duplex modelling using a variety of architectural choices: training special predictors for overlapping speech (Ruede et al., 2017; Chen et al., 2025b), modelling user and system speech as separate streams (Défossez et al., 2024; Hu et al., 2025) or by interleaving user and system chunks in a single sequence (Veluri et al., 2024; Lee et al., 2025). While the latter approaches enable models to *handle* interruptions and backchannels, they typically do not *model them explicitly* on the system side. That is, models are trained to robustly handle overlapping speech produced by the user, but are rarely studied in terms of whether the system itself interrupts or backchannels. As a result, existing evaluations focus on reactive rather than proactive conversational behaviour (Peng et al., 2025; Lin et al., 2025).

In addition, most current spoken conversational models offer limited customization. Properties such as the system's voice, conversational persona, topic framing, and interaction style, including tendencies toward backchanneling and interruption, strongly influence how appropriate and human-like a system feels in a given situation (Kühne et al., 2021; Cho et al., 2022). Commercial systems have demonstrated the importance of such design choices, e.g. by deliberately incorporating backchannels to increase perceived naturalness (Leviathan and Matias, 2018; Lin et al., 2022; Cho et al., 2022). Crucially, however, the optimal conversational behaviour can vary widely across users and use cases, motivating the need for controllable, instruction-following full-duplex speech systems.

A small number of recent works have explored instruction-following or persona-conditioned full-duplex models, for instance by specifying a target

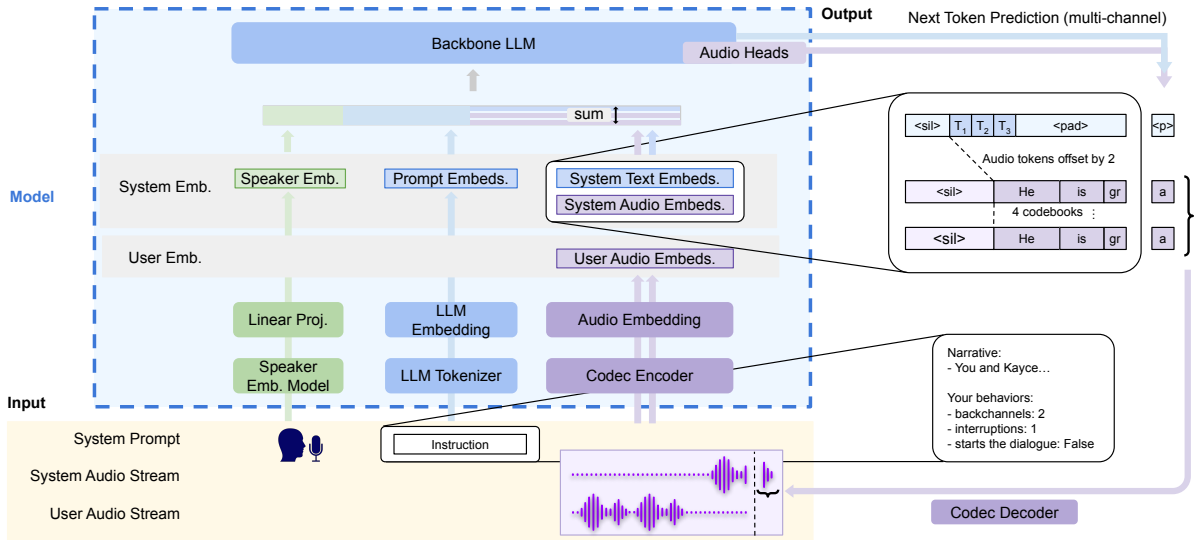

Figure 1: Overview of our controllable full-duplex model, which can be prompted to control (i) speaker voice, (ii) conversation topic, (iii) conversational behaviour (e.g., backchanneling and interruptions), and (iv) dialogue initiation. Only the LLM and audio heads are trained in a single-stage training, other components remain frozen.

voice (Chen et al., 2025b; Shi et al., 2025) or assigning the system a persona via prompt (Shi et al., 2025). To date, these models and their code have not been publicly released, they are computationally expensive to train, and the relevant behavioural characteristics are not evaluated systematically.

In this work, we present *F-Actor*, a **f**ull-duplex model that behaves like an **actor** following conversational instructions. We describe a practical approach for training it under typical academic resource constraints. By keeping the audio encoder frozen and finetuning only the LLM, our approach requires just 2,000 hours of training data and two days on four A100-40GB GPUs, without relying on large-scale pretraining or multi-stage optimization.

We demonstrate that it is possible to train a model that can follow explicit instructions regarding (i) speaker voice, (ii) conversation topic, (iii) conversational behaviour, such as backchanneling and interruptions, and (iv) dialogue initiation (user- or system-driven). Fig. 1 shows an overview of our model. Our contributions are three-fold:

1. We introduce the first open, controllable full-duplex model.
2. We systematically experiment with and analyze different design and architectural choices.
3. We develop a single-stage training protocol yielding good models on an academic budget.

We release both the model and the training code publicly, enabling reproducibility and further research on controllable full-duplex speech systems.

2 Background

This section reviews the key components of our full-duplex speech language model; alternative approaches are discussed in Section 3.

Full-Duplex Modelling. Full-duplex interaction, simultaneous speaking and listening, is often modelled using parallel user and system speech streams (Hu et al., 2025; Défossez et al., 2024). To enable the backbone LLMs to process speech streams, raw audio is typically mapped to discrete acoustic units (DAUs), which can be treated as tokens in the model vocabulary. Each speech stream then emits DAUs at every timestep, including silence DAUs, and embeddings for each stream are combined by summation (Défossez et al., 2024; Hu et al., 2025), or by token-wise fusion (Wang et al., 2025) to allow concurrent processing. To produce speech, the generated DAUs are mapped back into speech waveforms with pre-trained vocoders.

An alternative is to interleave user and system tokens within a single stream using speaker indicators (Veluri et al., 2024; Lee et al., 2025). However, this approach is limited by the chunk-size for interleaving and the longer sequence length, limiting its suitability for true full-duplex scenarios such as interruptions and backchannels.

Discrete Acoustic Units. DAUs are typically produced by neural audio codecs such as Mimi (Défossez et al., 2024), EnCodec (Défossez et al., 2022), or SoundStream (Zeghidour et al., 2021). These codecs compress audio into discrete codes opti-

mized for reconstruction and are widely used as both inputs and outputs in recent speech language models (Défossez et al., 2024; Hu et al., 2025).

The codec models represent each audio frame using multiple discrete codes, drawn from multiple codebooks. The choice of quantization determines how these codes are modelled. Dependent codebooks, typically produced by residual vector quantization (Zeghidour et al., 2021, RVQ), require hierarchical prediction: the language model predicts the first code, while additional modules generate the remaining layers (Défossez et al., 2024; Wang et al., 2023). In this work, we choose independent codebooks from finite scalar quantization (Mentzer et al., 2024, FSQ), which simplify modelling by allowing all codes for a frame to be predicted jointly (Hu et al., 2025). Additional details on encoding and reconstruction are provided in App. A.

DAUs may be integrated into an LLM either by extending its vocabulary (Hu et al., 2025) or by using a dedicated embedding layer for audio tokens in parallel to the text token embedding layer. In this work, we choose the latter, which is often more efficient, as audio codebooks are substantially smaller than text vocabularies.

3 Related Work

Full-duplex models have gained significant attention, resulting in a diverse set of architectures.

Architectures. Most models rely on text-based LLM backbones, with the exception of Nguyen et al. (2023). Cascaded architectures (Chen et al., 2025b; Wang et al., 2024b; Chen et al., 2025a; Zhang et al., 2025a; Wang et al., 2024a) explicitly model turn-taking and overlapping speech. This is achieved through mechanisms such as chunk-wise state prediction (Wang et al., 2024b), additional prediction modules (Chen et al., 2025b), modified VAD (Chen et al., 2025a), or control tokens (Zhang et al., 2025a; Wang et al., 2024a; Yu et al., 2024).

End-to-end architectures typically support these behaviours natively and have therefore become increasingly popular. Among them, Moshi (Défossez et al., 2024) and its variants (Ohashi et al., 2025; Shi et al., 2025) represent a common design choice: employing separate system, user, and text streams, and codec-based models for speech input and output (Défossez et al., 2024; Défossez et al., 2022; Zeghidour et al., 2021). Other approaches use continuous features at the input (Hu et al., 2025; Fu et al., 2025), or at both input and output (Yu et al., 2024). Some architectures avoid multiple input–output streams by interleaving modalities (Veluri et al., 2024; Lee et al., 2025; Yu et al., 2024), or by embedding them jointly at the token level (Wang et al., 2025). For more details, we refer the reader to Arora et al. (2025a).

Instruction-Following. While recent models emphasize different aspects of full-duplex interaction, for example, incorporating chain-of-thought reasoning (Arora et al., 2025b), mechanisms for handling background noise (Liao et al., 2025), or improved turn-taking (Cui et al., 2025), only a few focus on instruction following.

One such model is BeDLM (Lee et al., 2025), which supports instructions with regards to the conversational narrative and speaking behaviour encoded via prompts and special tokens. However, BeDLM is primarily a dialogue generation model, as it generates both speakers rather than only the system side. MinMo (Chen et al., 2025b), a cascaded speech model with duplex capabilities, can be instructed using embeddings that control emotions, dialects, speaking rate, and voice imitation. These capabilities, however, are evaluated only in MinMo's TTS setting on in-house test sets, leaving it unclear whether they extend to the duplex model. Finally, Voila (Shi et al., 2025), an end-to-end full-duplex model, is trained for instruction following by conditioning the system on a persona via text prompts and on a specific voice via a learnable speaker embedding. However, these instruction-following abilities are not evaluated or reported.

Unfortunately, MinMo, Voila[1], and BeDLM did not release their code or models, making direct comparisons with them impossible.

4 Model

F-Actor, our full-duplex model, is based around an LLM backbone, augmented with a speaker embedding model and an audio encoder. Unlike previous work (Défossez et al., 2024; Hu et al., 2025), *F-Actor* is designed with instruction following in mind, that is, incorporating both a speaker embedding and an instruction-following prompt. In the following, we describe each component of the system in detail. An overview is provided in Fig. 1.

Architecture. We use `Llama3.2-1B-Instruct`[2] (Grattafiori et al., 2024) as our backbone LLM.

[1] Voila released the base model, but not the full-duplex.
[2] 🤗 meta-llama/Llama-3.2-1B-Instruct

For speech encoding and decoding, we employ `Nemo-nano-codec-22khz-0.6kbps-12.5fps`[3] (Casanova et al., 2025). The codec consists of four independent codebooks of size 4032, producing four DAU streams per input audio stream, operating at a frame rate of 12.5 fps. In contrast to Hu et al. (2025), we keep the NanoCodec encoder *frozen* and train only the LLM.

NanoCodec has two advantages: (i) it uses FSQ quantization (Mentzer et al., 2024) so the four codebooks are independent, enabling parallel prediction without requiring a depth-transformer architecture (Défossez et al., 2024), and (ii) it has shown strong performance on speech synthesis (Hu et al., 2025).

Instruction Following Component. We design our model to adhere to explicit instructions. An instruction specifies (a) the narrative of the conversation, (b) whether the system should initiate interaction, (c) the frequency of backchanneling and interruptions, and (d) the system's voice characteristics. We encode (a)–(c) as a textual prompt embedding and (d) as a speaker embedding projected into the LLM's token space. To obtain the speaker embeddings, we extract the first five seconds of speech from a speech sample and encode them as in Shi et al. (2025) using the ECAPA-TDNN architecture (Dawalatabad et al., 2021). We concatenate the speaker and textual embeddings and prepend the sequence to the audio stream during training.

During inference, the model begins by appending its first generated token to this instruction prefix, thereby initiating the conversation according to the specified behavioural parameters.

DAU Embeddings. Our model processes two speech streams simultaneously: a user stream *user* and a system stream *sys*, enabling full-duplex modeling of overlapping speech (see Section 2). For each stream $s \in \{\text{user}, \text{sys}\}$, NanoCodec produces four DAU sequences $\text{DAU}_1^s, ..., \text{DAU}_4^s$.

We embed each codebook stream DAU_i^s with a separate embedding layer specific to the stream (user or system) and codebook index i. Separate embedding matrices for the user and system streams allow the model to reliably distinguish speakers. The embedding dimension is chosen to match the token embedding dimension of the LLM backbone. Unlike Hu et al. (2025); Shi et al. (2025), who extend the LLM's vocabulary with DAU tokens, we use dedicated embedding layers to avoid computing a large joint softmax over text tokens and DAUs, thereby significantly improving efficiency. We then sum the codebook embeddings over the codebook indices and user and systems streams to produce an input embedding vector x:

$$x = \sum_{s \in \{\text{user, sys}\}} \sum_{i=1}^{4} \text{Embed}(\text{DAU}_i^s)$$

Finally, we concatenate the speaker embedding, the instruction prompt, and the DAU embedding to give the input for the LLM as shown in Fig. 1.

DAU Generation. On top of the LLM, we attach eight linear heads that project the last token final hidden state back into DAU logits. Let $H \in \mathbb{R}^{1 \times d}$ denote the last token's final hidden state (with hidden size d). Each linear head $W_k \in \mathbb{R}^{d \times |C|}$, $k = 1, \ldots, 8$, maps H to the logits for one DAU codebook: $\text{DAU}_k = HW_k$, where $|C| = 4032$ is the codebook size. The eight heads correspond to the four user and four system DAU streams. During inference, only the four system heads are used to sample DAU tokens, which are decoded into a waveform via the NanoCodec decoder.

Text Stream. Following Hu et al. (2025); Défossez et al. (2024), we not only generate audio, but also experiment with generating the corresponding text, using an additional system-side text stream. To do so, we need to align the text tokens and audio DAUs (alignment strategies are described in Section 6). After embedding the text stream, we add it to the system's speech stream. The LLM uses this summed embedding as input. We then use the LLM's original language-modeling head to predict the next text token.

5 Evaluation

We evaluate our full-duplex models along two criteria: general system capabilities and instruction-following performance. Implementation details of the metrics can be found in App. C.

5.1 Dialogue Generation for Evaluation.

To evaluate these capabilities, we generate dialogues between two instances of our model, allowing them to *talk to each other* (Veluri et al., 2024), assigning each model instance its respective personalized instructions.

5.2 General System Capabilities Eval.

Following prior full-duplex work (Arora et al., 2025b; Zhang et al., 2025b), we report perplexity

[3] 🤗 nvidia/nemo-nano-codec-22khz0.6kbps-12.5fps

on the speech and text streams. We assess speech quality with UTMOS (Saeki et al., 2022), a neural, reference-free metric that predicts mean opinion scores to estimate perceived speech naturalness and quality. To evaluate dialogue behaviour, we measure speaking-time balance between speakers and compute WER between transcribed generated audio and generated text to assess how understandable the audio is and how coherent the text and audio streams are. Note, that the model is conditioned only on the dialogue topic and is free to generate any plausible conversation for that topic, rather than reproducing the test set dialogues. Consequently, WER is computed only between the generated text and the transcription of the generated audio, and not against the test set transcripts.

5.3 Instruction-Following Capabilities Eval.

We evaluate instruction-following performance along four dimensions: (1) *Speaker Initiation*: Accuracy in initiating the conversation according to the prompt. (2) *Speaker Embedding Consistency*: Cosine similarity between the target speaker embedding and the generated speech, averaged across speakers and conversations. To detect potential drift, we also compute the distance (1–cosine similarity) between the first and last segments for each speaker. (3) *Narrative Adherence*: An LLM judge (`Llama-3.1-8B-Instruct`[4] (Grattafiori et al., 2024)) evaluates the alignment (of the transcript) with the narrative specified in the prompt (prompts in Fig. 4). (4) *Backchannels (BCs) and Interruptions*: We measure the correlation between prompt-specified and generated counts using Pearson's r, with two-sided p-values computed from the exact distribution.

For (4), we experiment with different algorithms for detecting BCs and interruptions. We initially adopted the FD-Bench (Peng et al., 2025) implementation, which uses Silero-VAD (Team, 2024), but default thresholds did not generalize well to the dataset used in this work (Lee et al., 2025, details in Section 6), often misclassifying interruptions or BCs. Detecting these events is challenging because short BCs and overlapping speech can be easily missed or misaligned with transcripts. We compare timestamps from Silero-VAD, Parakeet[5], and Vosk[6], performing a grid search over interruption and overlap threshold parameters. Parakeet achieves the most reliable performance and is used for all subsequent evaluations. Detailed results for all alignment methods are reported in Tab. 4, with grid-search details provided in App. C.2.

6 Experimental Setup

In this section, we describe the training data, setup and evaluation protocol of our full-duplex model.

6.1 Data

We use the Behavior-SD dataset (Lee et al., 2025), which contains 2,164 hours of English multi-turn, two-speaker conversations with annotations for narrative structure, backchannels (BCs), and interruptions. Behavior-SD is generated using CosyVoice TTS (Du et al., 2024) and, despite being synthetic, the authors claim that it exhibits strong emotion appropriateness and dialogue naturalness in human evaluations (Lee et al., 2025). While real conversational speech would be preferable, large-scale, wideband, full-duplex datasets do not exist (Chen and Yu, 2025), especially with BC and interruption annotations, making synthetic data a practical choice. We extend the data in the following ways to serve our needs for instruction-following models.

Text-Speech Alignment. As described in Section 4, we experiment with generating not only speech, but also the corresponding text. We explore both word- and utterance-level alignment of speech and text, as prior work has shown mixed results (Défossez et al., 2024; Hu et al., 2025). To this end, we align text and speech tokens using forced alignment with Kaldi (Povey et al., 2011) and a public model[7] trained on Librispeech (Panayotov et al., 2015). We exclude conversations with text–speech mismatches due to occasional TTS deviations from the transcript, resulting in 74.4% of the original training set.

Instruction Following Prefix. To train the model as an instruction-following full-duplex system, we construct a prompt, similar to Lee et al. (2025), specifying (i) the number of system interruptions and BCs in the conversation, provided by Behavior-SD; (ii) whether the system should initiate the dialogue or wait for the user; and (iii) the narrative of the conversation. For the latter, we use the annotated narrative in Behavior-SD, and transform it into a narrative from the system's perspective to eliminate data bias: In roughly 85% of the cases

[4] meta-llama/Llama-3.1-8B-Instruct
[5] nvidia/parakeet-tdt-0.6b-v2
[6] vosk-model-en-us-0.22
[7] kaldi-model-m13

Loss	Spk. Emb.	Text Stream	Text Align.	Audio Delay	PPL DSU ↓	PPL Text ↓	UTMOS ↑	WER % audio/text ↓	Avg. Speaking Diff (s) ↓
Behaviour-SD Dialogues Testset					-	-	3.78	4.5	15.44
s/u	✓	✗	n/a	n/a	25.23	n/a	3.20	n/a	**10.69**
s/u	✓	✓	utt.	0	22.85	1.48	3.33	23.19	11.50
s/u	✓	✓	utt.	1	22.85	1.51	3.39	17.35	12.25
s/u	✓	✓	utt.	2	22.43	1.51	3.41	15.18	11.30
s/u	✓	✓	word	0	25.47	**1.33**	3.17	33.40	12.21
s/u	✓	✓	word	1	24.71	1.47	3.31	19.53	14.00
s/u	✓	✓	word	2	23.22	1.57	3.40	8.35	13.93
s/u	✗	✓	word	2	24.14	1.57	**3.47**	**7.16**	13.77
s	✓	✓	word	2	**21.45**	1.55	3.41	7.59	13.42
s/u	✓	✓(BC/I tok.)	word	2	23.38	1.56	3.39	9.19	14.75

Table 1: General modeling results for full-duplex models trained to predict both system and user (s/u) or only the system role (s). Models are compared with and without a text stream, using word- or utterance-level alignment, and with different audio delays relative to the text. *BC/I tok* indicate special tokens for backchannels and interruptions.

the narrative begins with the speaker who starts the conversation, which biases the model to rely on the narrative rather than instruction (ii) to decide on conversation initiation. Rewriting this narrative was performed using `Gemma-1.1-7b-it`[8] (Team et al., 2024) for training, and GPT-5.1 for the test set. The latter was chosen to provide higher-quality data for evaluation. Prompts and example narratives are provided in Figs. 2 and 3.

Speaker Embedding. We condition the system on speaker embeddings extracted from the relatively small set of Behavior-SD speakers (52 speakers). By restricting training and evaluation to this closed speaker set, we prevent our model being used for voice cloning or impersonation. Details on the extraction and its integration are in Section 4.

6.2 Training Details

To maximize efficiency, our approach utilizes a single training stage that avoids additional speech or text pretraining as required by prior models such as Défossez et al. (2024). Only the LLM backbone and the linear DSU heads are trained, while the speech encoder remains frozen. Training is performed on four NVIDIA A100-SXM4-40GB GPUs with early stopping based on validation performance, resulting in around 48 hours of training per model. We use a maximum sequence length of 2,048 tokens, a per-GPU batch size of 1, and 8 gradient accumulation steps. Training and inference parameters are provided in Tab. 5 in App. D.

6.3 Experiments

We experiment with a range of architectural and training choices for our full-duplex model.

Loss on system (s) and user (u). We train models that either predict only the system role (s) or both user and system roles (s/u). Accordingly, the training loss is computed either only on the system tokens (s) or on both user and system tokens (s/u).

Text-Stream. We examine the effect of adding a text stream on the system side (see Section 4), enabling the model to predict not only audio but also the corresponding text (Défossez et al., 2024; Hu et al., 2025). We further experiment with introducing special tokens that precede BCs and interruptions (BC/I tok.) to help the model generate the numbers specified in the prompt.

Text Alignment. We experiment with different alignment strategies for the text stream, using either utterance-level (padding after each utterance) or word-level alignment (padding after each word).

Audio Delay. We experiment with delaying the audio stream by up to two tokens relative to the text stream, generating text tokens first while padding the audio stream before audio token generation begins (Défossez et al., 2024; Hu et al., 2025).

RVQ vs. FSQ. All primary experiments use an FSQ encoder to benefit from independent codebooks. To test the generality of our multi-head DSU prediction architecture, we also evaluate an RVQ-based setup using `Mimi`[9] (Défossez et al., 2024) with 8 codebooks. This allows us to verify whether predicting DSUs for each codebook in parallel remains effective with RVQ quantization.

6.4 Baselines

The only available instruction-following models, BeDLM, MinMo, and Voila (Section 3), do not

[8] 🤗 google/gemma-1.1-7b-it

[9] 🤗 kyutai/mimi

Loss	Spk. Emb.	Text Stream	Text Align.	Audio Delay	Correct Start (% ↑)	Spk. Sim. (cos ↑)	Spk. Drift (1-cos ↓)	Narrative (1–5 ↑)	BC Corr.* (per dial. ↑)	Inter. Corr.* (per dial.↑)
Behaviour-SD Dialogues Testset					100.00	0.62	0.62	4.04	0.92	0.74
Lower Baseline					50.0[a]	0.35[b]	0.75[c]	1.26[d]	-	-
s/u	✓	✗	n/a	n/a	54.00	0.52	0.66	1.94	0.56	0.21
s/u	✓	✓	utt.	0	88.76	0.52	0.67	2.12	0.31	0.13
s/u	✓	✓	utt.	1	99.00	**0.54**	0.63	2.20	0.40	0.22
s/u	✓	✓	utt.	2	99.60	0.53	**0.61**	2.23	0.36	0.16
s/u	✓	✓	word	0	89.80	0.52	0.66	2.00	0.53	0.23
s/u	✓	✓	word	1	99.60	**0.54**	0.63	2.15	0.54	**0.28**
s/u	✓	✓	word	2	99.80	**0.54**	**0.61**	**2.39**	0.54	0.25
s/u	✗	✓	word	2	99.40	0.36	0.64	2.20	0.50	0.19
s	✓	✓	word	2	99.60	**0.54**	0.64	2.27	**0.57**	0.17
s/u	✓	✓(BC/I tok.)	word	2	**100.00**	**0.54**	0.62	2.36	0.48	**0.28**

[a] random spk. starts; [b] swapped spk. embeddings; [c] first embedding = spk. 1, last = spk. 2; [d] random narrative.
∗ All correlations are statistically significant with $p < 0.01$

Table 2: Instruction-following results for full-duplex models trained to predict both system and user (s/u) or only the system role (s). Models are compared with and without a text stream, using word- or utterance-level alignment, and with different audio delays relative to the text. *BC/I tok* indicate special tokens for backchannels and interruptions.

release code or model checkpoints, making direct comparison infeasible. For instruction-following, we therefore use the Behavior-SD test dialogues as a *topline* across all metrics and report metric-specific *bottomlines*.

For turn-taking behaviour, we compare our model against Moshi (Défossez et al., 2022) and dGLSM (Nguyen et al., 2023), reporting Inter-Pausal Units (IPUs), intra-turn pauses, between-turn gaps, and overlaps per minute following their reports and definitions and VAD implementation provided by Nguyen et al. (2023), using Pyannote[10] (Plaquet and Bredin, 2023; Bredin, 2023).

7 Results

This section discusses the results of our full-duplex models in terms of general modeling abilities, instruction following, comparison with other SOTA models and the influence of sampling parameters.

7.1 General Modeling Abilities

All results for general modeling abilities are shown in Tab. 1. Introducing a text stream alongside audio generation substantially improves performance across all metrics, reducing perplexity from 25.23 to 21–23. Word-level alignment between audio and text further improves synchronization compared to utterance-level, reducing WER between generated text and transcribed audio from over 23% to approximately 7.2% when combined with an audio delay of 2 tokens. This delay consistently improves WER across settings without degrading speech naturalness, with UTMOS remaining stable around 3.4–3.5. Omitting the speaker embedding slightly improves speech quality (UTMOS = 3.47).

The best overall trade-off is achieved using word-level alignment and audio delay of 2 under the system-only loss, yielding the lowest perplexity (21.45) and near-minimal WER (7.59%) while maintaining competitive speech quality and balanced speaking time of both speakers. We observe that at least 1% of WER arises from words seen fewer than 100 times in the training data; although generated correctly in the text stream, they are mispronounced and therefore mistranscribed.

For reference, Behaviour-SD Dialogues represent ground-truth speech, achieving UTMOS of 3.78 and WER of 4.5%. While our full-duplex models do not match these oracle values, the results demonstrate the effectiveness of the proposed training strategy.

7.2 Instruction Following

Tab. 2 summarizes our models' performance across instruction-following metrics. We find that generally, the model trained on predicting both the system and user roles, with a text-stream, special interruption and backchannel tokens, and an audio delay of 2 performs the best.

Similarity to Prompted Speaker. Incorporating speaker embeddings produces speech that matches the target speaker with up to 54 points cosine similarity, independent of other configurations. Likewise, speaker drift is minimal, in all configurations equal to or lower (0.61) than in the Behavior-SD test set (0.62), indicating a stable speaker voice.

Correct Start. The ability to accurately predict

[10] pyannote-audio

whether the model should initiate the conversation strongly depends on whether the model also generates text. Such models, in combination with audio delay, achieve over 99% accuracy, whereas models generating only DAUs perform similar to random. Based on manual inspection, we find that the model follows the prompt: if both speakers are set to start, speech overlaps, whereas if no one is set to start, a 1-second pause precedes the first speaker.

Dialogue Narrative Adherence. In terms of following the narrative specified in the prompt, we observe a clear benefit from incorporating a text stream and audio delay. Both word- and utterance-level alignments perform comparably, with slightly improved adherence for longer audio delays. Restricting training to the system role alone results in a small degradation in narrative adherence, likely because the model does not explicitly represent the interlocutor's turns, making it more difficult to sustain a coherent, narrative-aligned dialogue.

Backchannels and Interruptions. Backchannels (BC) show a clear correlation between the frequency specified in the prompt and those generated by the model, particularly for word-level alignments. The overall best configuration with 2 token audio delay, yields a correlation of 0.54. Correlation for interruptions is lower (0.25). This is likely because interruptions are rare, averaging only 0.9 per conversation in the training set. Having special tokens preceding BCs and interruptions only yields improvements for interruptions.

7.3 Comparing Turn-Taking to SOTA Models

In Tab. 3, we compare turn-taking behaviour to Moshi (Défossez et al., 2024) and dGSLM (Nguyen et al., 2023), with all metrics reported as cumulative durations per category. For all four metrics, Inter-Pausal Units (IPUs), intra-turn Pause, between-turn Gap and Overlap, our model exhibits behavior similar to the training data and the baseline models. Like Moshi and non-cascaded dGLSM, our model's pauses are generally longer than the gaps, reflecting real-world conversational statistics (Heldner and Edlund, 2010).

7.3.1 The Role of Sampling Parameters

Similar to Défossez et al. (2024), we observe that sampling temperature has a significant impact on model behaviour. Results for different temperatures (0.6–1.0) using our best-performing model (word-level alignment; audio delay of 2) on the dev set are reported in Tab. 6 in App. E. As temperature increases, the dialogue becomes more dynamic, with more BCs and interruptions, more balanced speaking time between the speakers, and improved coherence and adherence to the narrative. However, higher temperatures also lead to slightly lower UTMOS scores and higher WER, reflecting a mild impact on speech quality. In our setup, a temperature of 0.9 provides the best trade-off, and all results reported in this paper use this setting.

Model		IPU	Pause	Gap	Overl.
Best non-casc.	Nguyen et al. (2023)	41.4s	13.8s	10.7s	6.1s
Best casc.		54.8s	0.0s	5.3s	0.0s
Ground Truth		53.5s	5.5s	4.4s	3.6s
Moshi	Défossez et al. (2024)	50.8s	7.0s	4.5s	4.1s
Ground Truth		51.1s	6.4s	4.2s	3.3s
Ours	(best)	59.3s	10.4s	3.0s	5.4s
Ground Truth	(Beh.-SD)	55.8s	10.8s	3.8s	3.0s

Table 3: Cumulated durations per minute across models for turn-taking events.

7.4 RVQ vs FSQ

Finally, we train our best-performing configuration (word-level alignment, audio delay of 2, s/u loss) using an RVQ encoder. Results are reported in Tab. 7 in App. E. Although we do not explicitly model codebook dependencies and instead predict all codebooks in parallel, performance is comparable to that of the FSQ-based models. However, as expected, speech quality degrades substantially, with UTMOS dropping to 2.5 and similarly low scores observed for speaker embedding metrics.

8 Conclusion

In this work, we introduce a framework for controllable full-duplex speech models that can be trained using less than 2,000 hours of speech. Despite the relatively limited training data and compute, our model *F-Actor* produces coherent conversations with speech characteristics similar to the training data. It can be prompted with regards to conversation topic, speaker voice, and conversation initiation. Furthermore, we take steps toward making our model controllable with respect to backchannels and interruptions, allowing a user to decide how proactively the system should communicate with users. Finally, we systematically evaluate key design choices and release our model and code publicly, laying the foundation for future work on instruction-following full-duplex speech models.

9 Limitations

We identify the following main limitations of our work:

1. While the model's predictions of backchannels and interruptions correlate with the numbers specified in the prompt, the model consistently produces fewer such events than requested. As discussed above, the prompt mainly serves as a directional signal indicating whether more or fewer backchannels or interruptions should occur, rather than enforcing an exact count. We believe this limitation is primarily due to the training data, in which backchannels and interruptions are relatively rare. Training on data with a higher density of these phenomena may allow the model to better learn their frequency and timing. Nonetheless, our results provide useful insights into how such behaviors can be modeled in full-duplex systems and point toward promising directions for future work.

2. The codec model we use, NanoCodec, does not currently support streaming encoder inference, which limits real-world deployment to chunk-wise processing rather than fully online generation.

3. We demonstrate the instruction-following capabilities of our full-duplex model in English conversations. While it would be very interesting to see whether our training recipe also works for other languages, this is currently infeasibly, as training data for other languages is not available (Chen and Yu, 2025).

10 Potential Risks

Spoken conversational models pose inherent risks, including misuse for deception, impersonation, or social engineering. In particular, systems capable of natural, interactive speech could be exploited for scams or other forms of fraudulent behaviour.

We take several steps to mitigate these risks. First, the model is restricted to a small fixed pool of speaker embeddings, which prevents imitation of arbitrary or identifiable real-world voices. Second, the model is trained exclusively on text-to-speech (TTS) data, resulting in speech outputs that retain a synthetic character rather than fully natural human speech. This reduces the likelihood that the system could be mistaken for a real individual in high-stakes settings.

Finally, the model is released for research purposes only, with the goal of advancing work on controllable full-duplex conversational speech. We encourage responsible use and stress that safeguards and deployment policies are necessary for any real-world applications.

References

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, and 1 others. 2024. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*.

Siddhant Arora, Kai-Wei Chang, Chung-Ming Chien, Yifan Peng, Haibin Wu, Yossi Adi, Emmanuel Dupoux, Hung-Yi Lee, Karen Livescu, and Shinji Watanabe. 2025a. On the landscape of spoken language models: A comprehensive survey. *Preprint*, arXiv:2504.08528.

Siddhant Arora, Jinchuan Tian, Hayato Futami, Jiatong Shi, Yosuke Kashiwagi, Emiru Tsunoo, and Shinji Watanabe. 2025b. Chain-of-thought reasoning in streaming full-duplex end-to-end spoken dialogue systems. *Preprint*, arXiv:2510.02066.

Hervé Bredin. 2023. pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In *Proc. INTERSPEECH 2023*.

Edresson Casanova, Paarth Neekhara, Ryan Langman, Shehzeen Hussain, Subhankar Ghosh, Xuesong Yang, Ante Jukic, Jason Li, and Boris Ginsburg. 2025. Nanocodec: Towards high-quality ultra fast speech LLM inference. In *26th Annual Conference of the International Speech Communication Association, Interspeech 2025, Rotterdam, The Netherlands, 17-21 August 2025*. ISCA.

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. 2025a. Fireredchat: A pluggable, full-duplex voice interaction system with cascaded and semi-cascaded implementations. *Preprint*, arXiv:2509.06502.

Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, Yabin Li, Xiang Lv, Jiaqing Liu, Haoneng Luo, Bin Ma, Chongjia Ni, Xian Shi, Jialong Tang, Hui Wang, and 17 others. 2025b. Minmo: A multimodal large language model for seamless voice interaction. *Preprint*, arXiv:2501.06282.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki

Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Yuxuan Chen and Haoyuan Yu. 2025. From turn-taking to synchronous dialogue: A survey of full-duplex spoken language models. *Preprint*, arXiv:2509.14515.

Myra Cheng, Kristina Gligoric, Tiziano Piccardi, and Dan Jurafsky. 2024. AnthroScore: A computational linguistic measure of anthropomorphism. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 807–825, St. Julian's, Malta. Association for Computational Linguistics.

Eugene Cho, Nasim Motalebi, S. Shyam Sundar, and Saeed Abdullah. 2022. Alexa as an active listener: How backchanneling can elicit self-disclosure and promote user experience. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW2).

Wenqian Cui, Lei Zhu, Xiaohui Li, Zhihan Guo, Haoli Bai, Lu Hou, and Irwin King. 2025. Think before you talk: Enhancing meaningful dialogue generation in full-duplex speech language models with planning-inspired text guidance. *Preprint*, arXiv:2508.07375.

Nauman Dawalatabad, Mirco Ravanelli, François Grondin, Jenthe Thienpondt, Brecht Desplanques, and Hwidong Na. 2021. Ecapa-tdnn embeddings for speaker diarization. In *Proc. Interspeech 2021*, pages 3560–3564.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *Transactions on Machine Learning Research*.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, Zhifu Gao, and Zhijie Yan. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *Preprint*, arXiv:2407.05407.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *Preprint*, arXiv:2410.00037.

Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, Long Ma, Xiawu Zheng, Rongrong Ji, Xing Sun, Caifeng Shan, and Ran He. 2025. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *Preprint*, arXiv:2501.01957.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Mattias Heldner and Jens Edlund. 2010. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *CoRR*, abs/2106.07447.

Ke Hu, Ehsan Hosseini-Asl, Chen Chen, Edresson Casanova, Subhankar Ghosh, Piotr Żelasko, Zhehuai Chen, Jason Li, Jagadeesh Balam, and Boris Ginsburg. 2025. Efficient and Direct Duplex Modeling for Speech-to-Speech Language Model. In *Interspeech 2025*, pages 2715–2719.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu-Anh Nguyen, Morgane Riviere, Abdelrahman Mohamed, Emmanuel Dupoux, and 1 others. 2022. Text-free prosody-aware generative spoken language modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8666–8681.

Katharina Kühne, Martin H. Fischer, and Yuefang Zhou. 2021. The human takes it all.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

Sehun Lee, Kang-wook Kim, and Gunhee Kim. 2025. Behavior-SD: Behaviorally aware spoken dialogue generation with 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 9574–9593, Albuquerque, New Mexico. Association for Computational Linguistics.

Yaniv Leviathan and Yossi Matias. 2018. Google duplex: An ai system for accomplishing real-world tasks over the phone.

Borui Liao, Yulong Xu, Jiao Ou, Kaiyuan Yang, Weihua Jian, Pengfei Wan, and Di Zhang. 2025. Flexduo: A pluggable system for enabling full-duplex capabilities in speech dialogue systems. *Preprint*, arXiv:2502.13472.

Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, Shinji Watanabe, and Hung yi Lee. 2025. Full-duplex-bench v1.5: Evaluating overlap handling for full-duplex speech models. *Preprint*, arXiv:2507.23159.

Ting-En Lin, Yuchuan Wu, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022. Duplex conversation: Towards human-like interaction in spoken dialogue systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3299–3308. ACM.

John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude E. Shannon. 2006. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Mag.*, 27(4):12–14.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2024. Finite scalar quantization: Vq-vae made simple. In *International Conference on Representation Learning*, volume 2024, pages 51772–51783.

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

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Mary Williamson, Gabriel Synnaeve, Juan Pino, Benoît Sagot, and Emmanuel Dupoux. 2025. SpiRit-LM: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52.

Atsumoto Ohashi, Shinya Iizuka, Jingjing Jiang, and Ryuichiro Higashinaka. 2025. Towards a Japanese Full-duplex Spoken Dialogue System. In *Interspeech 2025*, pages 1783–1787.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Yizhou Peng, Yi-Wen Chao, Dianwen Ng, Yukun Ma, Chongjia Ni, Bin Ma, and Eng Siong Chng. 2025. FD-Bench: A Full-Duplex Benchmarking Pipeline Designed for Full Duplex Spoken Dialogue Systems. In *Interspeech 2025*, pages 176–180.

Alexis Plaquet and Hervé Bredin. 2023. Powerset multi-class cross entropy loss for neural speaker diarization. In *Proc. INTERSPEECH 2023*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, and 1 others. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, volume 1, pages 5–1. Hawaii.

Antoine Raux and Maxine Eskenazi. 2012. Optimizing the turn-taking behavior of task-oriented spoken dialog systems. *ACM Trans. Speech Lang. Process.*, 9(1).

Robin Ruede, Markus Müller, Sebastian Stüker, and Alex Waibel. 2017. Yeah, right, uh-huh: A deep learning backchannel predictor. *Preprint*, arXiv:1706.01340.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022. In *Interspeech 2022*, pages 4521–4525.

Nicholas Sanders, Yuanchao Li, Korin Richmond, and Simon King. 2025. Segmentation-Variant Codebooks for Preservation of Paralinguistic and Prosodic Information. In *Interspeech 2025*, pages 5403–5407.

Emanuel Schegloff. 1982. *Discourse as an interactional achievement: Some uses of 'uh huh' and other things that come between sentences*, pages 71–93. Georgetown University Press.

Yemin Shi, Yu Shu, Siwei Dong, Guangyi Liu, Jaward Sesay, Jingwen Li, and Zhiting Hu. 2025. Voila: Voice-language foundation models for real-time autonomous interaction and voice role-play. *Preprint*, arXiv:2505.02707.

Tanya Stivers, Nick J. Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter De Ruiter, Kyung-Eun Yoon, Stephen C. Levinson, Paul Kay, and Krishna Y. 2009. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106:10587 – 10592.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Silero Team. 2024. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. `https://github.com/snakers4/silero-vad`.

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

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Wei Xia, and Yuanjun Xiong. 2024a. A full-duplex speech dialogue scheme based on large language model. In *Advances in Neural Information Processing Systems*, volume 37, pages 13372–13403. Curran Associates, Inc.

Qichao Wang, Ziqiao Meng, Wenqian Cui, Yifei Zhang, Pengcheng Wu, Bingzhe Wu, Irwin King, Liang Chen, and Peilin Zhao. 2025. Ntpp: Generative speech language modeling for dual-channel spoken dialogue via next-token-pair prediction. *Preprint*, arXiv:2506.00975.

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

Dan Wells, Hao Tang, and Korin Richmond. 2022. Phonetic analysis of self-supervised representations of english speech. In *Interspeech 2022*, pages 3583–3587.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. 2024. Salmonn-omni: A codec-free llm for full-duplex speech understanding and generation. *Preprint*, arXiv:2411.18138.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Hao Zhang, Weiwei Li, Rilin Chen, Vinay Kothapally, Meng Yu, and Dong Yu. 2025a. Llm-enhanced dialogue management for full-duplex spoken dialogue systems. *Preprint*, arXiv:2502.14145.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chao-Hong Tan, Zhihao Du, and ShiLiang Zhang. 2025b. OmniFlatten: An end-to-end GPT model for seamless voice conversation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14570–14580, Vienna, Austria. Association for Computational Linguistics.

A Background: Speech and LLMs

To enable Large Language Models (LLMs) to process and generate audio, speech waveforms must be mapped into a discrete sequence of units. This compression serves two primary functions. First, it downsamples the high sample rate acoustic signal along the temporal axis to manageable sequence lengths. Second, it creates a finite vocabulary of units, allowing the model to be trained using standard cross-entropy objectives analogous to text-based language modeling. However, the properties of these units depend fundamentally on the learning paradigm used to derive them, broadly categorized into reconstruction-based or discriminative-based approaches.

A.1 Discrete Representations of Speech

The first category of discrete representations fo speech consists of units often termed "acoustic tokens" or Discrete Acoustic Units (DAUs), derived from reconstruction-based Neural Audio Codecs (NACs) such as Mimi, EnCodec, or SoundStream (Défossez et al., 2024; Défossez et al., 2022; Zeghidour et al., 2021). These units are optimized to compress and perfectly reconstruct the input signal. Consequently, they encode all acoustic information, including speaker identity, prosody, and recording conditions, into a single stream that can be decoded directly back into a waveform. While this enables end-to-end modeling such as in Moshi and VALL-E (Défossez et al., 2024; Wang et al., 2023), the high information density of DAUs can make modeling stability challenging.

In contrast, Discrete Speech Units (DSUs) are derived from discriminative Self-Supervised Learning (SSL) models like HuBERT or WavLM (Hsu et al., 2021; Chen et al., 2022). Typically obtained via offline clustering of continuous hidden representations (e.g., via K-Means), DSUs correlate strongly with phonetic structure rather than low-level acoustics. While sometimes erroneously referred to as "semantic tokens," they function effectively as a learned, phone-like vocabulary (Wells et al., 2022). Unlike DAUs, DSUs tend to discard paralinguistic and prosodic information while preserving more phonetic content at lower bitrates. At higher bitrates, however, they allow for the restoration of paralinguistic and prosodic information at the cost of diminishing the phonetic interpretability of the discrete units (Sanders et al., 2025). Although some previous work has decoded DSUs directly back into a waveform (Lakhotia et al., 2021), more recent work chooses to model DSUs with additional information like F0 (Kharitonov et al., 2022; Nguyen et al., 2025) or to use a two-stage or coarse-to-fine generation process. In this latter approach, a separate model generates continuous representa-

tions from DSUs before generating the waveform from those continuous representations (Anastassiou et al., 2024).

A.2 Quantization Methods

The choice of quantization method fundamentally dictates the architecture of the Speech Language Model, specifically regarding how it models the multiple codes representing a single time step. The two primary paradigms involve dependent codebooks, typically resulting from Residual Vector Quantization (Zeghidour et al., 2021, RVQ), and independent codebooks, such as those derived from Finite Scalar Quantization (Mentzer et al., 2024, FSQ). In RVQ-based models like EnCodec or Mimi (Défossez et al., 2022; Défossez et al., 2024), the codebooks operate sequentially where each quantizes the residual error of the previous one. This creates a strong dependency among the codes within a single frame. Therefore, generation often requires a hierarchical approach. The primary language model autoregressively predicts the code for the first RVQ layer, while a secondary mechanism, such as the Depth Transformer in Moshi or the nonautoregressive model in VALL-E, predicts the codes for subsequent RVQ layers for that frame (Défossez et al., 2024; Wang et al., 2023). Conversely, methods like NanoCodec use FSQ to project the latent space into independent subspaces, resulting in codebooks that are statistically independent (Casanova et al., 2025). This independence simplifies the modeling task by allowing the Language Model to predict all codes for a frame simultaneously. Such models, as seen in Hu et al. (2025), typically utilize a multiple head output layer, termed a Hydra head, which predicts the indices for all independent codebooks in parallel without the need for an additional sequential depthwise modeling step.

B Data

B.1 Prompts

To train the model as an instruction-following full-duplex system, we construct a prompt, similar to Lee et al. (2025), that specifies (i) the number of interruptions and backchannels in the conversation, provided by Behavior-SD; (ii) whether the system should initiate the dialogue or wait for the user; and (iii) the topic of the conversation. For the latter, we use the annotated narrative in Behavior-SD, e.g., *Karina was playing her music loudly and Yoseph did not enjoy it. The bass was thumping and the lyrics were explicit. Yoseph felt annoyed and asked Karina to turn it down.*

We observed that in roughly 85% of the cases the narrative begins with the name of the speaker who also starts the conversation. In early experiments, this caused the model to ignore the explicit instruction about who should speak first, as it relied instead on the narrative. To prevent this, we rewrite the narrative to be system-centric. In doing so, we also replace the system's name with *you*, which encourages the model to interpret the narrative from the system's own perspective. For instance, if the system corresponds to *Karina*, the rewritten version becomes: *You were playing your music loudly and Yoseph did not enjoy it. The bass was thumping and the lyrics were explicit. Yoseph felt annoyed and asked you to turn it down.* We use `google/gemma-1.1-7b-it`[11] (Team et al., 2024) to perform this rewriting on the train set and GPT-5.1 on the test set.

As an alternative formulation, we also experiment with goal-oriented prompting, in which the system does not receive a narrative but instead a set of goals it should accomplish during the dialogue. These goal descriptions are likewise generated using the same LLM.

Example prompts can be found in Fig. 2.

System Prompt using Narrative:

```
Generate a dialogue between you (Karina)
 and another speaker (Yoseph) based on
the given narrative. Follow the specific
 behavior instructions for you.

Narrative:
- You were playing your music loudly and
 Yoseph did not enjoy it. The bass was
thumping and the lyrics were explicit.
Yoseph felt annoyed and asked you to
turn it down.

Your behaviors:
- backchannels: 9
- interruptions: 0
- starts the dialogue: False

Ensure that the dialogue reflects the
behaviours of you.
```

Figure 2: Example prompts from the train set.

B.2 Rewriting Narratives

Rewriting prompts can be found in Fig. 3.

[11] https://huggingface.co/google/gemma-1.1-7b-it

Prompt to rewrite narrative to system/user style.

```
Your task is to rewrite a written narrative from the perspective of a specified
speaker,
replacing only that speaker's name and pronouns with 'you' and 'your'.
Do NOT change other characters or their pronouns.
Do NOT change the narrative events or add explanations. Maintain the narrative flow
and adjust active/passive voice as needed.

Example:
Input: Replace Karina and corresponding pronouns with you and your.
Karina was playing her music loudly and Yoseph did not enjoy it.
The bass was thumping and the lyrics were explicit. Yoseph felt annoyed and asked
Karina to turn it down.
Output: You were playing your music loudly and Yoseph did not enjoy it.
The bass was thumping and the lyrics were explicit. Yoseph felt annoyed and asked
you to turn it down.

New Input Narrative:
Replace {speaker} and corresponding pronouns with you and your.
{narrative}
Output Narrative:
```

Figure 3: Prompt to rewrite the narrative from Behavior-SD.

C Evaluation

C.1 General System Capabilities Eval.

To assess general system capabilities, we report the perplexity on both the speech and text streams. Speech quality is evaluated using UTMOS (Saeki et al., 2022), a model that predicts mean opinion scores and has been applied previously to other full-duplex models (Arora et al., 2025b; Zhang et al., 2025b). We first use Parakeet[12] to obtain segment-level timestamps for each speaker, calculate UTMOS for each segment, and then average these scores across segments and speakers to obtain the final dialogue-level score. Additionally, we measure the speaking-time difference between the two speakers to ensure balanced participation and that no single speaker dominates the conversation. Finally, we transcribe the generated audio using Parakeet and calculate the word-error rate (WER) against the generated text stream to evaluate the alignment between the model's speech and text outputs. We remove special tokens, such as dedicated backchannel and interruption tokens before calculating WER.

Instruction-Following Capabilities Evaluation We evaluate the model's instruction-following along four dimensions:

1. **Speaker Initiation**: The ratio of the correct speaker starting the conversation according to the prompt. We determine the first speaker using Parakeet-generated segment [13].

2. **Speaker Embedding Consistency:** Cosine similarity between the target speaker embedding and the generated speech for each speaker, averaged across speakers. To compute this, we randomly sample snippets of 3–5 seconds from each dialogue, encode them with ECAPA-TDNN (Dawalatabad et al., 2021) , and compare to the target embeddings. To assess potential speaker drift over the dialogue, we also calculate the distance (1–cosine similarity) between the first and last segments for each speaker and average across speakers.

3. **Narrative Adherence:** An LLM judge (`Llama-3.1-8B-Instruct`[14] (Grattafiori et al., 2024)) evaluates the alignment (of the Parakeet transcript) with the narrative specified in the prompt. Prompts for the LLM judge are provided in Fig. 4.

4. **Backchannels and Interruptions:** We measure the number of backchannels and interruptions per speaker and report correlations with the prompt-specified counts. Pearson's correlation coefficient (r) quantifies the linear relationship, and two-sided p-values are computed using the exact distribution for the null hypothesis $r > 0$. More details on the detection of backchannels and interruptions can be found in App. C.2.

C.2 Counting Backchannels and Interruptions

For evaluating backchannels and interruptions, we experiment with several detection algorithms. We initially adopt the FD-Bench (Peng et al., 2025) implementation, which uses Silero-VAD (Team, 2024) to obtain speech timestamps. However, default thresholds for merging words into utterances and defining backchannels or interruptions did not generalize well to the Behavior-SD test set, often misclassifying interruptions or assigning incorrect timestamps to short backchannels. Detecting these events is challenging due to their short duration, so sometimes they are not transcribed or assigned a timestamp at all.

[12] 🤗 nvidia/parakeet-tdt-0.6b-v2

[13] 🤗 nvidia/parakeet-tdt-0.6b-v2

[14] 🤗 meta-llama/Llama-3.1-8B-Instruct

To address this, we compare timestamps obtained from Silero-VAD (Team, 2024), Parakeet[15], and Vosk[16]. We perform a grid search over three parameters:

1. Split threshold: Controls when consecutive words are merged into a single utterance, varied from 0.20 to 0.90 in steps of 0.005.

2. Interruption threshold: Specifies how close to the end of a conversation a segment must occur to be considered an interruption, varied from 0.10 to 0.70 in steps of 0.005.

3. Overlap tolerance: Defines the maximum allowed temporal overlap between segments of the same speaker for them to be treated as overlapping speech rather than separate utterances, varied from 0.05 to 0.50 in steps of 0.005.

The best-performing configuration uses a split threshold of 0.565, an interruption threshold of 0.405, and an overlap tolerance of 0.435, which we adopt for all subsequent evaluations.

Model	Segm.	Missing ↓		Extra ↓	
		BC	Inter.	BC	Inter.
Parakeet	word	**1.0** $\pm$ **1.4**	**0.5** $\pm$ **0.7**	**0.2** $\pm$ 0.4	0.5 $\pm$ 0.7
	segm.	2.7 $\pm$ 2.9	1.0 $\pm$ 1.0	3.0 $\pm$ 2.9	1.0 $\pm$ 1.9
Kaldi	word	1.7 $\pm$ 1.9	0.8 $\pm$ 0.9	1.1 $\pm$ 1.6	**0.4** $\pm$ **0.6**
	segm.	1.3 $\pm$ 1.5	0.9 $\pm$ 0.9	0.7 $\pm$ 0.7	0.6 $\pm$ 0.7
Silero-VAD	segm.	1.6 $\pm$ 1.7	1.0 $\pm$ 1.0	0.3 $\pm$ 0.6	0.8 $\pm$ 1.0

Table 4: Results of interruption (*Inter.*) and backchannel (*BC*) detection algorithms on the Behavior-SD test set, averaged per dialogue across the test set. For each model, we report performance using its native segmentation (*segm.*) and using word-level timestamps merged with our grid-searched parameters (*word*). Silero-VAD uses the default FD-Bench settings (Peng et al., 2025).

Parakeet provides the most reliable performance on Behavior-SD (Lee et al., 2025), missing on average only 1.0 backchannels and 0.5 interruptions per conversation. This means, 79.4% of the backchannels are correctly classified, and 79.2% of the interruptions are correctly classified. Results for all alignment methods using these threshold are reported in Tab. 4. Based on these findings, we adopt Parakeet for all subsequent evaluations.

[15] nvidia/parakeet-tdt-0.6b-v2
[16] vosk-model-en-us-0.22

D Training Details

Training and inference parameters are shown in Tab. 5. We train our models on four NVIDIA A100-SXM4-40GB. Training takes around 46-48 hours, depending on the configuration.

For inference, we show that sampling parameters, in particular the temperature, have a great influence on model performance in Section 7.3.1. We find that for our setup, a temperature of 0.9 works best and hence report all results (except for those in Tab. 6) with this temperature.

Hyperparameter	Value
Train Batch Size (micro-batch)	1
Gradient Accumulation Steps	8
Learning Rate	5e-5
Max Steps (optimizer updates)	100,000
Examples per Step	32
Max Sequence Length	2048
Precision	bfloat16
Audio Vocab Size	4032
Early Stopping Patience	10
Gradient Clipping	1.0
Weight Decay	0.01
Max Sequence Length	1024
Temperature	0.9*
Top-k	40
Top-p	1

Table 5: Training and inference hyperparameters for training our full-duplex model using DeepSpeed. We use four NVIDIA A100-SXM4-40GB GPUs for training.
*Unless explicitely stated otherwise.

E Results

We report and discuss results for all of our models in Section 7, with two exceptions, that are reported in the appendix: Detailed results using different temperatures during inference are shown in Tab. 6. Results comparing models using NanoCodecs (FSQ) and Mimi (RVQ) audio encoding are shown in Tab. 7.

System Prompt for LLM Judge:

```
You are an expert evaluator of dialogues.
You only respond with a single digit from 1 to 5 based on the evaluation criteria.
```

User Prompt to Judge Dialogue Coherence:

```
Evaluate how well the following dialogue fits the given narrative.

Criteria:
1. Relevance: Does the dialogue clearly reflect the situation or topic described in
the narrative?
2. Consistency: Are the characters, events, and tone in the dialogue consistent with
 the narrative?
3. Faithfulness: Does the dialogue avoid introducing contradictions or unrelated
content?
- Do NOT judge fluency or engagement - only topical/narrative alignment.
- Score strictly between 1 and 5 (1 = Not related at all, 5 = Perfectly fits the
narrative).

Narrative:
{narrative}

Dialogue:
{transcription}

Score:
```

Figure 4: Prompt for the LLM Judge to judge the instruction following capabilities of our full-duplex model.

Temperature	UTMOS ↑	WER % audio/text ↓	Avg. Speaking Diff (s) ↓	Correct Start (% ↑)	Spk. Sim. (cos ↑)	Spk. Drift (1-cos ↓)	Narrative (1–5 ↑)	BC Corr.* (per dial. ↑)	Inter. Corr.* (per dial.↑)
Behaviour-SD	3.78	4.5	15.44	100	0.62	0.62	4.04	0.92	0.74
Lower Baseline	-	-	-	50[a]	0.35[b]	0.75[c]	1.26[d]	-	-
0.6	3.48	6.20	24.29	100.0	0.54	0.59	2.57	0.30	0.13
0.8	3.44	7.50	16.12	100.0	0.54	0.63	2.26	0.47	0.22
0.9	3.42	8.49	14.62	100.0	0.54	0.60	3.00	0.50	0.30
1.0	3.38	10.09	12.07	99.4	0.53	0.63	2.08	0.52	0.22

[a] random spk. starts; [b] swapped spk. embeddings; [c] first embedding = spk. 1, last = spk. 2; [d] random narrative.
$*$ All correlations are statistically significant with $p < 0.01$

Table 6: Instruction-following results for our best model (text-stream, audio delay of 2, word-alignment, trained on system + user) for different sampling parameters.

FSQ/RVQ	UTMOS ↑	WER % audio/text ↓	Avg. Speaking Diff (s) ↓	Correct Start (% ↑)	Spk. Sim. (cos ↑)	Spk. Drift (1-cos ↓)	Narrative (1–5 ↑)	BC Corr.* (per dial. ↑)	Inter. Corr.* (per dial.↑)
Behaviour-SD	3.78	4.5	15.44	100	0.62	0.62	4.04	0.92	0.74
Lower Baseline	-	-	-	50[a]	0.35[b]	0.75[c]	1.26[d]	-	-
FSQ (NanoC.)	**3.4**	8.35	13.93	99.8	**0.54**	0.61	2.39	**0.54**	**0.25**
RVQ (MimiC.)	2.5	**7.18**	**11.12**	**100.0**	0.40	**0.59**	**2.58**	0.40	0.25

[a] random spk. starts; [b] swapped spk. embeddings; [c] first embedding = spk. 1, last = spk. 2; [d] random narrative.
$*$ All correlations are statistically significant with $p < 0.01$

Table 7: Instruction-following results for our best model (text-stream, audio delay of 2, word-alignment, trained on system + user) for different sampling parameters.