

Constructing Dynamic Conversations with Neural Back-channel Generation and Voice-line Scheduling

Yifang Pan
evan.pan@mail.utoronto.ca
University of Toronto
Toronto, Canada

Chris Landreth
c.landreth@rogers.com
JALI Research
Toronto, Canada

Karan Singh
karansher.singh@utoronto.ca
University of Toronto
Toronto, Canada

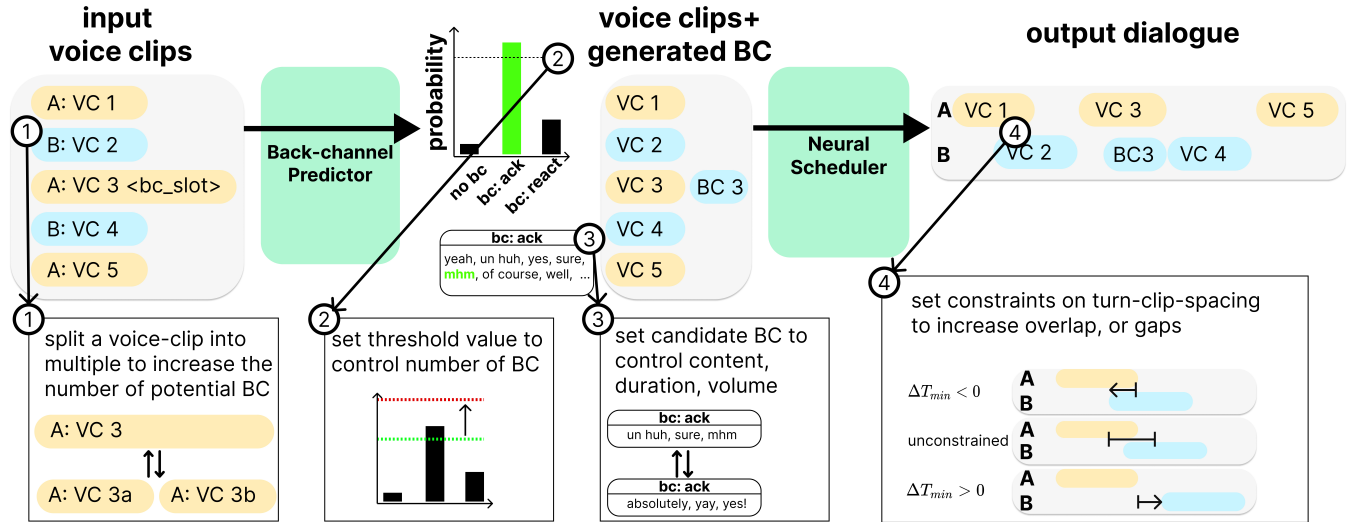


Figure 1: Input turn-clips are processed by the Back-channel Predictor, which determines and synthesizes a candidate verbal back-channel. The Neural Scheduler then sequences the turn-clips and generated back-channels with appropriate temporal gaps or overlap into a conversational dialogue. Circled numbers indicate animator-centric parameters: (1) turn-clip splitting, and (2) a back-channel probability threshold to control back-channel frequency; (3) a back-channel vocabulary with varying valence, duration, and intensity; and (4) timing constraints to control pause and overlap in conversational turn-taking.

Abstract

We present a novel approach to the problem of combining two voice-line channels of dyadic conversational audio into a unified conversation with controllable back-channeling and turn-taking dynamics. In voice performances for the game industry, the two tracks of voice-lines are typically recorded or synthesized in isolation. Our method takes two such unsynchronized tracks and uses a neural scheduler to sequence them into a single dynamic conversation, modifying both tracks by inserting or removing pauses and introducing voice overlap as appropriate. We further propose a back-channel predictor that is able to predict opportunity for back-channels for a given conversational turn’s audio and content, which the scheduler can then incorporate. We evaluate our scheduler and

predictor independently and in combination by: various quantitative comparison; a qualitative and perceptual comparison to prior art and naive baselines; a compelling set of results showing animator control over the diversity, extent and nature of back-channeling and turn dynamics.

CCS Concepts

• Computing methodologies → Animation; Neural networks.

Keywords

speech, audio-driven animation



This work is licensed under a Creative Commons Attribution 4.0 International License.
MIG '26, Charleston, SC, USA

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2824-2/2026/12
<https://doi.org/10.1145/3828647.3857292>

ACM Reference Format:

Yifang Pan, Chris Landreth, and Karan Singh. 2026. Constructing Dynamic Conversations with Neural Back-channel Generation and Voice-line Scheduling. In *The 19th ACM SIGGRAPH Conference on Motion, Interaction, and Games (MIG '26)*, December 11–13, 2026, Charleston, SC, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3828647.3857292>

1 Introduction

Voice recognition, synthesis, and *vocap* (voice-capture driven character animation) solutions are now commonly used for games, film and virtual avatars [Tan et al. 2021]. A fundamental advantage of voice is scalability, due to its ease and flexibility of recording. Characters in games like *Cyberpunk 2077* [Edwards et al. 2020] are driven by tens of thousands of lines of dialogue recorded in over ten languages. This dialogue however, is typically recorded by voice actors in isolation, read from a script with little preparation or context [Redding 2022]. Even for AAA games, “approximately 95% of the time, actors are recorded alone in the booth” (private communication [Pixel Audio Inc. 2026]). Consequently, despite expressive individual voice acting, simply splicing these recordings together produces conversational turns that sound uniformly robotic, lacking the natural overlaps and pauses that characterize interactions. Further, conspicuously missing in such conversations, are listener interjections or *back-channels*. Back-channels are short verbal utterances like “mmm-hmmm”, “cool” or “maybe”, that can indicate listener engagement, enthusiasm, or skepticism, but above all keep the conversation flowing [Clark and Brennan 1991]. While a vocabulary of back-channels can sometimes be recorded separately, and manually edited into conversations by an animator [Pixel Audio Inc. 2026], such an approach is tedious and does not scale to increasingly many hours of dialogue to be processed for games like *Cyberpunk 2077*.

We jointly address the scheduling and back-channel problem with a unified system, and present the first controllable neural solution to producing dynamic conversational turns from a script of independently recorded or TTS (text-to-speech) voice-lines. As seen in Fig. 1, our framework operates on a script-ordered set of voice-lines (recorded by voice actors or synthesized with TTS), and produces natural two-speaker dialogue audio via a two-component pipeline. The *Back-channel Predictor* first determines where listener back-channels should occur and of what type, operating at the voice-line level. The *Neural Scheduler* then determines the precise timing of back-channels, turn transitions and voice overlap for both channels jointly. Both components combine learned models with algorithmic inference, and expose hyperparameters that afford fine-grained control over the back-channel frequency, valence, and conversational dynamics of the output at inference time.

Outside of applications in game development, we further see our model as a stepping stone towards more natural and interactive voice LLMs. Successful text-LLMs like ChatGPT and Gemini increasingly provide voice-LLM support [Team 2025]. These models are trained on synthetic TTS data, and fine-tuned using small conversation datasets (e.g., Fisher or Switchboard [Cieri et al. 2004]), and thus produce voice assistants that lack back-channeling and natural conversation dynamics. Our work can transform large text datasets into dynamic conversations using synthesized voices for use in training future conversational voice assistants, and other full-duplex model applications [Roy et al. 2026].

Evaluation. We evaluate each component of our model in isolation and the full pipeline end-to-end. The Back-channel Predictor achieves state-of-the-art performance on sentence-level back-channel opportunity prediction (section 4), and the Neural Scheduler produces conversational timing statistics closest to ground

truth across nearly all turn-taking metrics (Table 3). Evaluated end-to-end, our system generates back-channels at the highest frequency while achieving the best temporal alignment with natural listener behavior on Full-Duplex Bench (Table 4), and is significantly preferred over PersonaPlex [Roy et al. 2026] in a perceptual user study (section 6).

Contribution. Our principal contribution is, to our knowledge, the first end-to-end pipeline that combines two independently recorded voice-line channels into a unified dyadic conversation with natural, controllable back-channeling and turn-taking dynamics. Its key components are:

- (1) A *Neural Scheduler*, based on our formulation of voice-line scheduling as next-token prediction, in which a full-duplex language model is given the content of every line and learns only its timing.
- (2) A *Back-channel Predictor*, built around a multimodal classifier that predicts from a clip’s text and audio whether it invites a back-channel, and of what type.
- (3) A search-based inference algorithm that uses the scheduler to score candidate timings, exposing animator-centric parameters for back-channel insertion and high-level control over conversational turn dynamics without retraining.

2 Related Work

We classify related work as pertaining to text-to-speech synthesis, conversational dynamics, and full-duplex language models.

2.1 Synthesizing Speech from Text

TTS has evolved from early sequence-to-sequence models [Shen et al. 2018; van den Oord et al. 2016; Wang et al. 2017] to efficient non-autoregressive architectures and GAN vocoders [Kim et al. 2021; Kong et al. 2020; Ren et al. 2021, 2019]. Modern TTS is typically a two-stage process using discrete audio codecs [Défossez et al. 2022; Zeghidour et al. 2022]: Large Language Models generate tokens from text [Borsos et al. 2023a; Wang et al. 2023], which are decoded via diffusion [Borsos et al. 2023b; Shen et al. 2024] or flow matching [Chen et al. 2024b; Du et al. 2024; Peng et al. 2024; Resemble AI 2025] into high-fidelity speech [Chen et al. 2024a; Li et al. 2023]. Text alone however, is insufficient for dyadic conversation, which requires modeling speaker overlap [Reece et al. 2023] and listener back-channels [Anderson et al. 2025] absent in text transcripts. While joint-synthesis systems like MoonCast [Ju et al. 2025] and CoVoMix [Zhang et al. 2025] simulate basic turn-taking, they lack precise control over turn boundaries and do not model back-channels. We address these by using a neural scheduling system to combine single-speaker lines into natural two-channel dialogue, with explicit control over both turn-taking and back-channel dynamics.

2.2 Controlling Back-channels and Turn Dynamics

Back-channels provide essential grounding [Clark and Brennan 1991; Clark and Schaefer 1989], where continuers (e.g., *mmmh-mmm*) signal turn maintenance [Schegloff 1982] and emotional responses (e.g., *oh!*) invite elaboration [Heritage 1984; Tolins and Fox Tree 2014]. Inter-turn timing is equally communicative; while

the modal gap for English is around 240 ms [Stivers et al. 2009], deviations can signal competition [Trimboli and Walker 1984], deception [Reynolds and Rendle-Short 2011], or deliberation [McConaghie 2021]. Existing computational models approach these dynamics through multi-class classification [Ekstedt and Skantze 2020; Lin et al. 2025a; Onishi et al. 2023; Wang et al. 2024] or continuous Voice Activity Projection (VAP) [Ekstedt and Skantze 2022; Inoue et al. 2024]. However, these statistical methods struggle with back-channel reliability and lack mechanisms for animator control. We address these limitations by combining statistical and procedural components, enabling explicit control over the timing and content of both turn transitions and back-channels.

2.3 Full-Duplex Language Models

The emergence of speech language models has made it possible to train systems that understand spoken input and respond through next-audio-token prediction [Xie and Wu 2024; Xu et al. 2025; Zeng et al. 2024]. However, these models process a single audio stream at a time, and are thus unable to generate conversational dynamics such as overlapping speech or back-channels. Full-duplex models address this by jointly modeling two token streams at all times. dGSLM [Nguyen et al. 2022] introduced a dual-tower Transformer trained on stereo telephone conversations, while LSLM [Ma et al. 2024] and NTPP [Wang et al. 2025] extended the paradigm with richer inter-stream interaction. Paired with streaming audio encoders and decoders, systems like Moshi [Défossez et al. 2024] can process live user speech on one channel while generating responses on the other. However, the scarcity of dual-channel training data results in many of these models producing believable conversational dynamics but incoherent speech. Recent work such as PersonaPlex [Roy et al. 2026] addresses this through large scale TTS-synthesized training data, but is sparse in back-channeling. We complement this line of research, as our system can generate back-channel-rich dialogue data that could serve as training material for the next generation of full-duplex conversational agents.

3 Method Overview

Our framework operates on voice clips, which can either be lines recorded by a voice actor or a synthetic voice generated from text. In conversation, a **turn** is a contiguous stretch of spoken content by a single speaker that ends when the floor passes to the other. We refer to the voice audio of a turn (or a portion of it) as a voice clip. Given an ordered set of said voice clips, our system synthesizes natural two-speaker audio via a two-component pipeline (Figure 1). The first component, the **Back-channel Predictor**, determines *where* in the conversation listener back-channels should occur: a coarse, voice-clip classification of whether a back-channel response for the clip is warranted. The second component, the **Neural Scheduler**, then determines *when* precisely each voice line is played, jointly scheduling both the original input lines and any synthesized back-channels to produce natural turn transitions with voice overlap and back-channels. The pipeline exposes hyperparameters that afford fine-grained control over the conversational dynamics of the output at inference time (Section 7.1).

4 Back-channel Predictor

The Back-channel Predictor supplements the input turn clips with synthesized back-channel clips. At its core is a trained classifier that predicts for each voice-clip whether the listener should produce a back-channel and, if so, whether it should be a low-salience acknowledgment or a high-salience reactive response. Predictions are then used to produce a corresponding back-channel clip, which is added to the set of input clips to be scheduled.

We predict back-channels per voice clip rather than continuously over time, as in Voice Activity Projection (VAP) [Ekstedt and Skantze 2022; Inoue et al. 2024]. Clips are the unit in which our input is authored, and we found in practice that VAP models predict whether a turn will continue well, but are poor at deciding when to insert a back-channel. The classifier therefore decides only *whether* a clip receives a back-channel, and the Neural Scheduler (section 5) decides its precise timing.

4.1 Back-channel Prediction Classifier

Given a voice-clip, the classifier measures the semantic and auditory opportunity for a back-channel response, and if so, what type. We organize labels by *emotional salience* rather than functional or linguistic taxonomy, as back-channel choices are highly personal, depending on speaker intent [Morikawa et al. 2022], and varying with speaker-listener relationship [Fukunaga et al. 2025]. This yields three classes:

- **No-back-channel** (class 0): the clip presents no opportunity for a back-channel response.
- **Acknowledgment** (class 1): the clip invites a neutral, low-salience response (e.g., *uh-huh*, *yeah*).
- **Reactive** (class 2): the clip warrants an emotional, high-salience response (e.g., *wow*, *really?*, *for sure*, *absolutely!*).

This salience-based organization directly governs the behavioral impact of each back-channel on the perceived dynamics of the conversation. Once classified, a back-channel can be produced by sampling a voice line from a user-defined pool of back-channel audio, or generated through TTS in the same voice from a user-defined back-channel vocabulary.

4.2 Input Representation

The classifier is conditioned on a windowed, linearized view of the conversation, centered on a candidate back-channel site. For a target voice clip at index t , the input is constructed as follows. A context window of $\pm k$ clips (we use $k = 5$) around t is extracted. Each clip within this window is prefixed with a speaker-relative tag: the speaker of the target clip is always labeled A, and the other speaker B, so the model learns speaker-role-invariant representations. Clips are delimited by `</s>` boundary tokens, and a special `<bc_slot>` token is appended to the target clip to mark the candidate insertion site. The classifier predicts the back-channel class at this marked position. Since prediction operates at the clip level, the density of candidate sites, and therefore the number of back-channels inserted into a conversation, can be controlled by varying the granularity at which longer turns are segmented into clips.

Earlier back-channel decisions are fed back into the context: a clip that received a back-channel is followed by its class as a

tag token (e.g. [BC:Acknowledgment]), which lets the classifier account for recent back-channel density. An input might look like:

```
A: I was at Walmart today [BC:Acknowledgment] </s>
A: Shopping for some clothes </s>
B: I was there too </s>
A: I didn't see you there. <bc_slot> </s>
B: We probably went at different times </s>
```

The target speaker's waveform for the clip (capped at 15 s) is paired with the text input, providing prosodic cues.

4.3 Training Data

We source training examples from the natural split of the Seamless Interaction dataset, which provides dual-channel audio with sentence-level transcripts [Agrawal et al. 2025]. Back-channel annotations are derived automatically from text using a two-stage pipeline. First, a rule-based pre-filter identifies candidate clips by requiring that each be short (under 1.0s or at most 3 words), temporally isolated from the same speaker's surrounding utterances (at least 0.3s silence on both sides), and preceded by the other speaker within 2 seconds. Candidates passing this filter are then submitted to a Qwen LLM, which classifies each into one of five back-channel types, Neutral, Positive, Emotional, Anticipation, and Confirmation, as defined in prior work [Morikawa et al. 2022], or rejects it as non-back-channel, using a ± 10 s conversational context window. These types are then collapsed into a 3-class scheme for training: *No-back-channel*, *Acknowledgment* (if Neutral or Positive), and *Reactive* (Emotional, Anticipation, or Confirmation). The pipeline yields 33,977 Reactive and 137,808 Acknowledgment back-channel clips. We additionally sample negative examples at a 5:1 ratio relative to positive instances, yielding 703,140 No-back-channel clips. The dataset is split 96/2/2 for train, validation, and test.

4.4 Model Architecture and Training

The classifier uses a late-fusion architecture combining a text encoder with a frozen speech encoder. The text branch is a fine-tuned RoBERTa-base encoder [Liu et al. 2019] extended with back-channel-specific special tokens. The audio branch uses a frozen HuBERT-base encoder [Hsu et al. 2021] with a learnable weighted sum over all 13 hidden states, following the SUPERB protocol [Yang et al. 2021], downsampled via a stride-4 convolution to 12.5 Hz. A learned fusion CLS token is prepended to the concatenated text and audio representations, augmented with modality embeddings, and processed by a 2-layer Transformer encoder. The CLS output is passed through a linear head projecting to $C=3$ classes. Full architecture details are available in our code repository.¹

The model is trained with cross-entropy loss using inverse-square-root class weighting to mitigate the imbalance between No-back-channel and the two back-channel classes. HuBERT parameters remain frozen throughout; all other parameters are optimized jointly. Training proceeds for up to 10 epochs with early stopping on validation macro-F1 (patience of 3 epochs).

4.5 Classifier Evaluation

We compare our model against the state-of-the-art back-channel classifiers of Wang et al. and Lin et al.², along with ablated variants of our own model. Both Wang et al. and Lin et al. are originally formulated as tri-class classifiers over {continue, switch, back-channel}; for a fair comparison we retrained both for the binary task {Back-channel, No-back-channel}, and we ensure all models receive information regarding future context. All models are trained on our processed dataset (subsection 4.3) and evaluated on held-out data.

We report five metrics, all computed on the binary BC-vs-NoBC decision: **P** precision and **R** recall, **F1**, the harmonic mean of precision and recall at the model's argmax decision; **AUC**, the area under the ROC curve, which measures how well the model ranks examples by back-channel likelihood independently of any threshold; and **EER**, the equal-error-rate operating point on the ROC curve where false-positive and false-negative rates coincide ($EER \approx 1 - AUC$ for well-behaved curves). Higher F1 and AUC are better; lower EER is better. AUC and EER are derived from the calibrated score $1 - P(\text{No-back-channel})$, so they reflect the full probability output rather than the hardened prediction.

We additionally study four ablations, trained without corresponding input contexts: **no audio** silences the waveform; **no future text** drops context turns following the target; **no past text** drops context turns preceding the target; **no current text** removes the words on the target turn itself, leaving only its turn marker.

Table 1 reports results on the held-out test set. Our full model attains an AUC of 0.798, outperforming both prior baselines. Among the ablations, removing audio causes the largest recall drop ($0.49 \rightarrow 0.25$) with little effect on precision, indicating audio is primarily responsible for surfacing back-channel-eligible moments. Removing future or current text each costs 0.07–0.11 AUC, confirming that linguistic content in and immediately following the target turn carries substantial signal. By contrast, removing past context barely affects performance ($\Delta AUC \leq 0.03$), suggesting the model relies far more on the current turn and its immediate continuation than on broader conversational history.

The classifier's reliance on audio also suggests why the LLM annotator used to label our training data (subsection 4.3) is not a substitute for the classifier: it works from text alone, and it labels back-channels that a listener has already produced, whereas at inference time no listener response exists.

The tri-class breakdown (Table 2) shows the model handles No-back-channel well ($F1 = 0.888$) but struggles to separate the two BC subtypes. Acknowledgment retains an AUC (0.792) on par with the binary task, while Reactive is substantially harder ($F1 = 0.229$), reflecting both its lower frequency (400 examples) and greater linguistic and prosodic variability. This aligns with Onishi et al. [2023], who similarly find that fine-grained back-channel categories are inherently harder to predict. The inherent ambiguity of back-channel type motivates our decision to expose animator control as an explicit stage in the pipeline.

²We found Lin et al.'s original training data contains significant leakage of training labels in their input, resulting in inflated performance. Our reimplementation of their released architecture on our data yields performance more consistent with prior work.

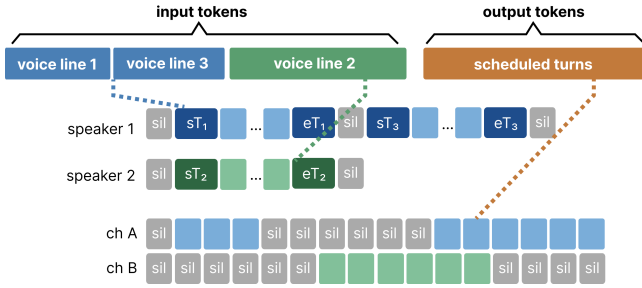
¹<https://github.com/EvanSamaa/NeuralSchedulerRelease>

Table 1: Binary back-channel classification results on held-out data. Higher F1, P, R, AUC are better; lower EER is better.

Model	F1 ↑	P ↑	R ↑	AUC ↑	EER ↓
<i>Baselines</i>					
Wang et al.	0.43	0.46	0.40	0.73	0.33
Lin et al.	0.42	0.34	0.55	0.69	0.36
<i>Ours</i>					
Full model	0.51	0.53	0.49	0.80	0.28
w/o audio	0.34	0.55	0.25	0.75	0.32
w/o future text	0.38	0.36	0.40	0.69	0.37
w/o current text	0.43	0.41	0.46	0.73	0.33
w/o past text	0.48	0.52	0.45	0.77	0.30

Table 2: Per-class performance of our full model on the three-way merged task (No-back-channel / Acknowledgment / Reactive).

Class	F1 ↑	P ↑	R ↑	AUC ↑	EER ↓	Support
No-back-channel	0.89	0.88	0.90	0.80	0.28	8,891
Acknowledgment	0.44	0.46	0.43	0.79	0.28	1,709
Reactive	0.23	0.24	0.22	0.76	0.31	400

**Figure 2: Illustration of how voice-line scheduling can be formulated as a next-token prediction problem.**

5 Neural Scheduler

While the Back-channel Predictor augments input voice-clips with back-channel clips, the Neural Scheduler orders the timing of all clips. The core of the Neural Scheduler is a full-duplex LLM fine-tuned on the task of scheduling an ordered set of voice lines. This section formally introduces the formulation of voice-clip scheduling, discusses how the model is trained, inference strategies, as well as evaluations of the scheduler.

5.1 Scheduling as Next Token Prediction

We formulate turn scheduling as a next token prediction problem by introducing two structurally distinct token regions: an *input* region that encodes the input turn-clips of each speaker, and an *output* region where the model generates interleaved output tokens, as shown in Figure 2.

Input. Each speaker’s input is an independent, non-interleaved sequence of audio tokens. For speaker $s \in \{A, B\}$, the input $s^{(s)}$

is a flat sequence of discrete audio tokens structured into N_s clips by per-turn start and end marker tokens $[sT_n]$ and $[eT_n]$, with a silence token *sil* separating consecutive clips:

$$s^{(s)} = (\text{sil}; [sT_n]; c_n^{(s)}; [eT_n])_{n=1}^{N_s} \quad (1)$$

where $c_n^{(s)}$ denotes the speech tokens for clip n of speaker s , as illustrated in Figure 2. Each clip’s speech tokens are drawn from an audio tokenizer; the single silence token *sil* between clips is a structural separator only, as it carries no temporal information and does not encode the actual duration of the pause, only that a clip boundary exists. As a result, the inputs are purely *content* representations: they specify what each speaker says and in what order, but are entirely stripped of timing. The two inputs generally have different lengths, reflecting the different number and duration of each speaker’s clips. The full input region is formed by concatenating both inputs sequentially, delimited by per-speaker boundary markers $[sA]$, $[eA]$, $[sB]$, $[eB]$:

$$x_{\text{in}} = ([sA]; s^{(A)}; [eA]; [sB]; s^{(B)}; [eB]) \quad (2)$$

Output. The output region x_{out} is where timing lives. It is a strictly interleaved, frame-aligned sequence of audio tokens:

$$x_{\text{out}} = (a_1^{(A)}, a_1^{(B)}, a_2^{(A)}, a_2^{(B)}, \dots, a_T^{(A)}, a_T^{(B)}) \quad (3)$$

of length $2T$, where each pair $(a_t^{(A)}, a_t^{(B)})$ represents one frame (at 12.5 Hz) of both channels simultaneously, similar to the formulation in [Wang et al. 2025]. Frames with no active speech are assigned the silence token *sil*, so the output region encodes the full temporal structure of the conversation including pauses, overlaps, turn transitions, through the pattern of speech and silence tokens across both channels.

The Neural Scheduler therefore learns the conditional distribution $p_\theta(x_{\text{out}} | x_{\text{in}})$, where the input region supplies the *what* and the output region captures the *when*. Crucially, x_{out} contains the same speech tokens as x_{in} , but interleaved across both channels with silence tokens encoding the pauses, overlaps, and turn transitions that constitute conversational dynamics.

5.2 Training Objective

Standard next-token prediction (NTP) loss alone is insufficient for training the Neural Scheduler, which we demonstrate in subsection 5.5.1. The fundamental difficulty is a severe class imbalance inherent to the output token sequence x_{out} . Within a speech turn, the task of predicting the next token can be trivially solved by simply tracking the token indices in x_{in} . Structural transition events, turn changes, mutual pauses, and simultaneous speech are collectively rare relative to within-turn continuation steps. As a result, a model that consistently predicts “continue the current speaker” achieves deceptively low cross-entropy while failing to capture the very events that define conversational timing dynamics. We call this *inertia bias*: since same-token continuations dominate the training signal, the model gravitates toward repeating the current channel’s token and never acquires a reliable internal representation of when a transition is imminent.

To counteract inertia bias, the Neural Scheduler is trained with a composite objective. The primary loss is a standard NTP cross-entropy over the $2T$ tokens of the output region:

$$\mathcal{L}_{\text{NTP}} = -\frac{1}{2T} \sum_{t=1}^T \sum_{s \in \{A,B\}} \log p_{\theta}(a_t^{(s)} \mid \mathbf{x}_{\text{in}}, a_{<t}^{(A)}, a_{<t}^{(B)}) \quad (4)$$

The auxiliary duration head is a two-layer MLP from the transformer’s final hidden state to 32 classes (subsection 5.3), trained to predict the *remaining silence duration* at each position: how many more consecutive frames the current token will persist on that channel before a transition. Concretely, the ground-truth label at position (t, s) is

$$d_t^{(s)} = \begin{cases} 0 & \text{if } a_t^{(s)} \text{ is a speech token,} \\ \min(r_t^{(s)}, 31) & \text{if } a_t^{(s)} = \text{sil,} \end{cases} \quad (5)$$

where $r_t^{(s)}$ is the countdown of consecutive silence frames remaining at frame t on channel s , including the current frame. Class 0 indicates that the channel is producing speech (no silence stretch to measure); classes 1–31 encode the remaining silence duration in 80 ms increments, with class 31 acting as a catch-all for silences of 31 or more frames (≥ 2.48 s). This labeling injects gradient signal at every frame *inside* a silence stretch, compelling the model to build an explicit, future-facing estimate of when the current segment will end rather than reasoning myopically about the next single frame. The duration cross-entropy is

$$\mathcal{L}_{\text{dur}} = \text{CE}(h_{\theta}(\mathbf{x}_{\text{out}}), \mathbf{d}) \quad (6)$$

The composite training objective is then a weighted sum of the two, in all of our experiments we use $\lambda_{\text{dur}} = 0.5$

$$\mathcal{L} = \mathcal{L}_{\text{NTP}} + \lambda_{\text{dur}} \mathcal{L}_{\text{dur}} \quad (7)$$

5.3 Architecture

The Neural Scheduler must process both single-channel script tokens in $\mathbf{x}_{\text{in}}$ and dual-channel interleaved tokens in $\mathbf{x}_{\text{out}}$. Rather than employing separate models per channel as in Moshi [Défossez et al. 2024], we use a single decoder-only LLM that handles both regions, following the interleaved-output paradigm of NTPP [Wang et al. 2025] and SyncLLM [Veluri et al. 2024]. The backbone is GLM-4-Voice [Zeng et al. 2024], which shares its architecture with the Qwen family [Xu et al. 2025]. We make three modifications to adapt the base model to the scheduling task.

Channel embeddings. To help the model disambiguate between token regions and between speakers, we introduce a learned channel embedding $\mathbf{e}_c \in \mathbb{R}^d$ for each of three channel identifiers $c \in \{0, 1, 2\}$. Tokens in the input region $\mathbf{x}_{\text{in}}$ receive the embedding $\mathbf{e}_0$; tokens on channel A in the output region $\mathbf{x}_{\text{out}}$ receive $\mathbf{e}_1$; tokens on channel B receive $\mathbf{e}_2$ (as shown in Figure 3). These embeddings are added to the query and key representations in each self-attention layer, enabling the model to distinguish between the single-channel input format and the two interleaved output channels throughout the network.

Positional embedding. We modify the rotary position encoding to reflect the temporal structure of each region. In the input region $\mathbf{x}_{\text{in}}$, positions increment sequentially by one per token. In the output region $\mathbf{x}_{\text{out}}$, the two tokens at each frame $(a_t^{(A)}, a_t^{(B)})$ share the

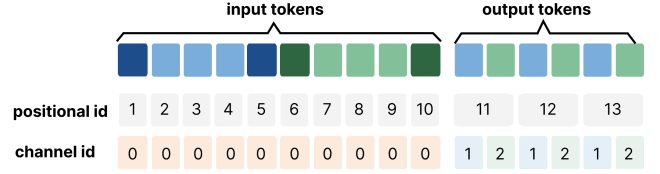


Figure 3: Illustration of positional id and channel id assignment for the corresponding embeddings.

same position index, so that the model’s notion of temporal order is consistent with the frame-aligned structure: position indices increment by one per frame rather than per token. Concretely, for a token at absolute index i in the output region with dialogue offset $\delta = i - |\mathbf{x}_{\text{in}}|$, the position index is $|\mathbf{x}_{\text{in}}| + \lfloor \delta/2 \rfloor$.

Duration prediction head and vocabulary. We extend the model with the duration prediction head described in subsection 5.2: a two-layer MLP projecting from the hidden dimension to 32 duration classes (corresponding to countdown values 0, 1, ..., 30 and a catch-all class 31 for durations ≥ 31 frames). Additionally, we expand the vocabulary with structural boundary tokens, per-speaker script delimiters ($[\text{sA}]$, $[\text{eA}]$, $[\text{sB}]$, $[\text{eB}]$), dialogue delimiters ($[\text{sD}]$, $[\text{eD}]$), and per-turn markers ($[\text{sT}_n]$, $[\text{eT}_n]$), to demarcate the regions and turns described in subsection 5.1.

5.4 Dataset Processing

We train the Neural Scheduler on the audio data from the Seamless Interaction dataset [Agrawal et al. 2025], a large-scale collection of two-channel conversations recorded on separate microphone channels. Processing each recording into a training example proceeds through four stages: audio tokenization, input and output region construction, turn ID assignment, and duration labeling. We deliberately exclude the Fisher corpus [Cieri et al. 2004] from training, reserving it exclusively for evaluation to avoid train-test contamination and to enable fair comparison against prior works.

Audio tokenization. Each channel is independently encoded into a sequence of discrete speech tokens using the WhisperVQ encoder from GLM-4-Voice [Zeng et al. 2024]. A voice activity detection (VAD) model [Team 2024] is then applied to each channel to obtain per-channel speech timestamps. Non-speech frames identified by VAD are replaced with the silence token *sil*, yielding two token streams that faithfully represent both the acoustic content and the activity pattern of each speaker.

Input and output region construction. The input region $\mathbf{x}_{\text{in}}$ is constructed by encoding each channel’s speech content into a compact, non-interleaved script. Consecutive runs of silence tokens are collapsed into a single *sil*, retaining only the turn boundaries rather than the raw temporal gaps. Each turn’s content is delimited by start and end markers $[\text{sT}_n]$ and $[\text{eT}_n]$, with a single silence token separating consecutive turns. For turns containing internal pauses (e.g., mid-utterance hesitations), a single *sil* is inserted between the sub-segments within the turn markers as well. The two per-channel scripts are concatenated sequentially and wrapped by per-speaker boundary markers as described in subsection 5.1. The output region $\mathbf{x}_{\text{out}}$ is formed by interleaving the two silenced token streams at

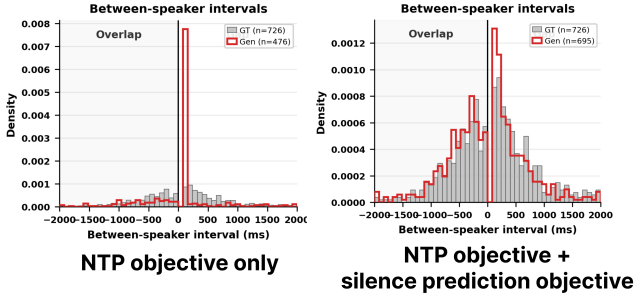


Figure 4: Between-speaker intervals (negative: overlap; positive: gap) in generated vs. ground-truth dialogue. Without duration prediction (left), generated intervals collapse into a spike near zero; with it (right), they track ground truth.

each frame, preserving the full temporal structure including pauses and overlaps. Long recordings are segmented into windows of at most 1024 tokens (approximately 40 s of conversation at 12.5 Hz per channel). During training, the cross-entropy loss is applied only over the output region; the input prefix is masked.

Turn ID assignment. Turn IDs annotate the input region to delineate conversational turns across both speakers. To assign them, VAD-derived speech segments from both channels are jointly sorted by onset time, and temporally adjacent segments from the same speaker are sometimes ($p = 50\%$) merged into contiguous turns. A 50 % overlap rule then handles back-channels: if at least half of a segment’s duration overlaps with the temporal span of an already-assigned turn, the segment is absorbed into that turn rather than initiating a new one, meaning that back-channels share the same turn ID as the active speech turn. Turn IDs are renumbered contiguously after all merging is complete, producing the indices n used in the turn markers $[sT_n]$ and $[eT_n]$.

Remaining duration labels. Each position in the output region is annotated with a remaining duration label as described in subsection 5.2. These labels provide per-frame supervision for the auxiliary duration prediction head.

5.5 Evaluation of Neural Scheduler

We evaluate the Neural Scheduler through an ablation of the duration prediction objective (subsection 5.5.1) and a quantitative comparison of conversational dynamics against baselines (subsection 5.5.2). Qualitative examples of its schedules (subsection 6.2) and the evaluation of the full system are presented in section 6.

5.5.1 Ablation: Duration Prediction Objective. Figure 4 ablates the auxiliary duration prediction objective described in subsection 5.2. Without it (left), inertia bias causes the generated between-speaker intervals to collapse into a narrow spike just above zero, as the model learns to perpetually continue the current token rather than anticipate transitions. Adding duration prediction (right) resolves this collapse, confirming that the auxiliary objective successfully counteracts inertia bias. Notably, the resulting distribution closely tracks ground truth across both overlaps and gaps, suggesting that the model has learned realistic turn timing dynamics.

Table 3: Turn-taking statistics (cumulated duration/min) for generated dialogues, compared to ground truth using mean absolute differences ($|\Delta|$) between predicted and ground truth values. Lower $|\Delta|$ values indicate better performance, with the best results highlighted in bold.

Model	Cumulated duration /min			
	$ \Delta\text{IPU} $	$ \Delta\text{Pause} $	$ \Delta\text{Gap} $	$ \Delta\text{Overlap} $
Naive Cascade	4.3	5.5	0.9	3.6
dGSLM	4.6	3.6	1.8	1.9
LSLM	4.1	3.4	1.5	2.3
NTPP	3.3	2.8	1.4	1.9
Moshi _{0.8}	16.0	6.8	8.3	2.1
Moshi _{0.9}	6.4	2.7	3.3	1.1
Moshi _{1.0}	<u>0.3</u>	<u>0.6</u>	0.3	<u>0.8</u>
Neural Scheduler	0.1	0.1	<u>0.6</u>	0.4

5.5.2 Quantitative Results. We quantitatively evaluate our Neural Scheduler’s understanding of conversational dynamics following the established protocol of NTPP [Wang et al. 2025] and Moshi [Défossez et al. 2024], where models are prompted with 10 seconds of audio and tasked to complete the conversation on unseen Fisher corpus [Cieri et al. 2004] samples. Performance is measured by comparing corpus statistics against ground truth across four VAD-derived turn-taking metrics: an **IPU** (Inter-Pausal Unit) is a contiguous speech segment on a single channel; a **Pause** is a within-speaker silence where the same speaker resumes; a **Gap** is a between-speaker silence followed by a turn change; and an **Overlap** is any interval of simultaneous activity on both channels. For each metric, we report the mean absolute difference $|\Delta|$ in cumulated duration (seconds per minute) against ground truth, where lower values indicate closer alignment with natural conversational dynamics.

We compare against Moshi [Défossez et al. 2024] at temperature (0.8, 0.9, 1.0), NTPP [Wang et al. 2025], dGSLM [Nguyen et al. 2022], LSLM [Ma et al. 2024], and the naive cascade baseline from [Wang et al. 2025], with baseline numbers taken directly from the respective publications under the same evaluation conditions. All models are evaluated on the same held-out test split of the Fisher corpus [Cieri et al. 2004].

Table 3 shows that our Neural Scheduler produces conversational dynamics closest to ground truth across nearly all metrics, outperforming both cascaded and full-duplex baselines. This is consistent with the nature of our approach: rather than generating content and timing jointly, the scheduler operates on pre-synthesized voice lines with known linguistic content, leaving timing as the sole degree of freedom to optimize.

Our formulation also sidesteps the intelligibility degradation observed in existing full-duplex models like Moshi [Défossez et al. 2024], which relies on softmax temperature to trade off turn-taking naturalness against speech quality during token inference, higher temperatures improve timing behavior at the cost of intelligibility. Because our model schedules fixed voice lines rather than generating tokens, it is immune to this tradeoff entirely.

5.6 Inference

At inference time, the Neural Scheduler operates as a *scoring function* rather than an autoregressive generator. Unlike autoregressive generation, where token identities are unknown, the tokens of $\mathbf{x}_{\text{out}}$ are fully determined by the onset positions of the voice-line clips: each frame either advances a clip or emits silence. This means that scheduling reduces to a search over onset assignments rather than a generation problem. As the Neural Scheduler is trained on natural conversation, its NLL reflects how consistent a given timing arrangement is with the conversational dynamics it has observed: arrangements with awkward gaps or poorly timed overlaps will receive high NLL, while natural-sounding ones will not. We therefore use NLL directly as a proxy for timing naturalness. Scheduling then proceeds by searching over candidate onset positions and committing the one that achieves the lowest NLL, effectively asking the model to judge arrangements rather than produce them. This formulation has three appealing properties: the speech content of every voice line is already fixed at inference time, so there is no risk of drift from the original voice lines; the search is fully deterministic, producing reproducible schedules; and the search space is explicitly parameterized, making the scheduler directly controllable without any modification to model weights, as we demonstrate in subsection 7.1.

Naively searching over all joint onset assignments is intractable, so we schedule greedily in three sequential passes, each conditioned on the placements committed by the previous one: (1) primary turns, (2) back-channels, and (3) a local re-timing correction. Algorithm 1 summarizes the procedure, Figure 5 illustrates it on a small example, and we detail each pass below. All timings are expressed in frames (token steps) at the model’s 12.5 Hz rate.

Scoring a placement. Given a set of committed onsets, we materialize the corresponding interleaved output region $\mathbf{x}_{\text{out}}$ by advancing the assigned clip on each channel at each frame and emitting sil elsewhere, then evaluate its NLL under the frozen model conditioned on $\mathbf{x}_{\text{in}}$. Because moving a single clip only perturbs timing locally, we score a window around the clip under consideration, its token span padded by a context margin of c frames on each side, rather than re-scoring the full dialogue; this keeps each candidate evaluation to one cheap forward pass. We write $\text{NLL}(\cdot)$ for this windowed score, where lower is more natural. We write $\mathcal{S}$ for the set of placements committed so far and $t@o$ for clip t placed with its onset at frame o , so $\text{NLL}(\mathcal{S} \cup \{t@o\})$ scores the current schedule with t added at o . For a placed clip, $\text{onset}(\cdot)$, $\text{end}(\cdot)$ and $\text{dur}(\cdot)$ give its first frame, end frame and length.

Pass 1: Primary turns. We first remove all back-channels and place the primary turns in conversational order. Each turn is anchored at the end of the most recently committed turn plus a nominal inter-turn gap g (the first turn is anchored at g), and we scan candidate onsets over a window of $\pm W$ frames around the anchor at stride σ , committing the onset of lowest windowed NLL. A hard floor prevents a turn from overlapping any previously committed turn on the *same* channel, while cross-channel overlap is permitted, which is what allows the model to schedule natural turn-transition overlap. This window is also the primary point of animator control (subsection 7.1): tightening or shifting the per-speaker bounds

forces simultaneous speech or widens inter-turn gaps into deliberate thinking pauses. Turns that contain internal pauses are optionally scheduled sub-segment by sub-segment, searching a short silence between consecutive sub-segments within a per-speaker range.

Pass 2: Back-channels. Back-channels are placed one at a time, each within the span of its host turn. For a back-channel of duration d hosted by a turn ending at frame h_{end} , we search onsets from the host’s start up to $h_{\text{end}} - \lceil rd \rceil$, so that at least a fraction r of the back-channel overlaps the host (multiple back-channels on one host are ordered after one another). Rather than an absolute NLL, each candidate is scored by its NLL *difference* $\Delta = \text{NLL}(\text{with back-channel}) - \text{NLL}(\text{baseline without it})$ against the current schedule, isolating the back-channel’s own contribution; we commit the minimizing onset. For the first back-channel on a given host we additionally search a host-accommodation strategy jointly with the onset: leave the host unchanged so the back-channel overlaps ongoing speech (*no-pause*), or split the host at the insertion point with a silence gap of d (*full*) or $\lfloor d/2 \rfloor$ (*half*). When a splitting strategy wins, the inserted silence is propagated as a rigid shift to every turn and back-channel scheduled at or after the split point, so downstream timing is preserved.

Pass 3: Local re-timing. A host split in Pass 2 can leave the remainder of the host turn timed awkwardly relative to the interjection. For each host turn that received a back-channel, we identify *break-points*, the sub-segment (or, if the back-channel lands in the host’s final segment, the following speaker’s turn) immediately after each back-channel. We freeze everything before a breakpoint q , search a new onset for it that minimizes NLL, and rigidly shift all subsequent material by the resulting delta; the accumulated shift then propagates to all downstream turns. The search runs from the end of the preceding clip on q ’s channel up to $\text{onset}(q) + D_q + E$, where D_q is the total duration of the back-channels before q on its host and E bounds the extra silence that may be added, so the resumption can move earlier or later. This mirrors how a speaker naturally adjusts the resumption of their turn after a listener interjects. The scan bounds, gap g , overlap ratio r , extra gap E , and context c are the only free parameters; the exact values used in our experiments, together with the full implementation, are available in our code repository.

6 Pipeline Evaluation

To our knowledge, ours is the first computational work to jointly address conversational synthesis through back-channel insertion and turn-clip scheduling. For this reason, we evaluate each component of our framework individually against relevant baselines in section 4 and section 5. We assess the full pipeline in three ways: (1) evaluating back-channel generation and placement using the full-duplex benchmark [Lin et al. 2025b]; (2) presenting illustrative conversational examples synthesized via text-to-speech; and (3) conducting a perceptual study to assess the overall system quality.

6.1 Full-Duplex Bench

We evaluate the joint performance of the back-channel generation and scheduling pipeline against existing full-duplex dialogue

Algorithm 1 Three-pass greedy scheduling. $\mathcal{S}$ is the schedule built so far and $t@o$ denotes clip t placed at onset frame o . Exact search ranges are given in the text, and the full procedure in our code release (<https://github.com/EvanSamaa/NeuralSchedulerRelease>).

Require: primary turns $\mathcal{P}$ in script order; back-channels $\mathcal{B}$, each with a host turn; frozen scheduler p_θ

- 1: $\mathcal{S} \leftarrow \emptyset$
- Pass 1: primary turns*
- 2: **for** each turn $t \in \mathcal{P}$ **do**
- 3: $\mathcal{W} \leftarrow$ onsets near the end of the previous turn
- 4: add $t@o$ to $\mathcal{S}$, with $o = \arg \min_{o' \in \mathcal{W}} \text{NLL}(\mathcal{S} \cup \{t@o'\})$
- 5: **end for**
- Pass 2: back-channels*
- 6: **for** each $b \in \mathcal{B}$ with host turn h **do**
- 7: $\mathcal{W} \leftarrow$ onsets that overlap h
- 8: pick $o \in \mathcal{W}$ and a pause for h with lowest NLL
- 9: add $b@o$ to $\mathcal{S}$; pause h at o and shift later clips
- 10: **end for**
- Pass 3: local re-timing*
- 11: **for** each clip q that resumes speech after a back-channel **do**
- 12: $\mathcal{W} \leftarrow$ onsets shortly after the previous clip
- 13: move q to its lowest-NLL onset in $\mathcal{W}$; shift later clips
- 14: **end for**
- 15: **return** $\mathbf{x}_{\text{out}}$ rendered from $\mathcal{S}$

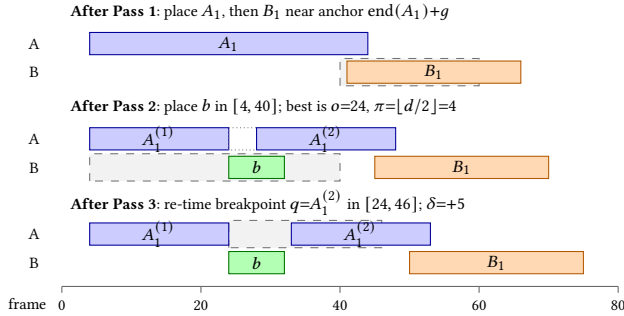


Figure 5: An illustrative example of the three passes. Pass 1 places the turns, Pass 2 inserts the back-channel b with a brief pause in A's turn (dotted), and Pass 3 re-times when A resumes. Dashed boxes mark each clip's search range.

systems using the Full-Duplex Bench evaluation framework [Lin et al. 2025b]. The benchmark assesses back-channel behavior along three axes: the Turn-Override Rate (TOR), which measures the proportion of back-channels that exceed a duration threshold and thus intrude on the main speaker's turn; Frequency (Freq), which measures the average rate of back-channel production per second; and the Jensen-Shannon Divergence (JSD) between the predicted back-channel temporal distribution and ground truth, where lower values indicate better alignment with natural back-channel timing.

As shown in Table 4, our proposed system achieves the best performance across all three back-channel metrics. A TOR of 0.000 indicates that none of our generated back-channels override the

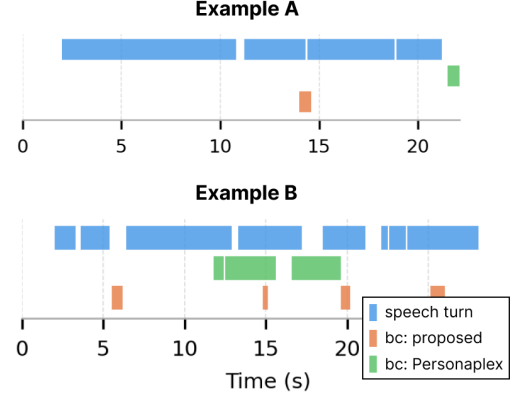


Figure 6: Voice activity timelines for two dialogue examples comparing back-channels generated by PersonaPlex and our proposed model.

main speaker; this is trivial, as our architecture only inserts back-channels. Our system also produces the highest back-channel frequency at 0.062 per second while achieving the lowest JSD of 0.644, suggesting that our model not only is capable of generating more back-channels but also schedules them in a manner most consistent with natural conversational behavior.

Table 4: Full-Duplex Bench back-channel evaluation results.

Model	Back-channel		
	TOR ($\downarrow$)	Freq ($\uparrow$)	JSD ($\downarrow$)
Freeze-Omni	0.636	0.001	0.997
Gemini	0.091	0.012	0.896
Moshi	1.000	0.001	0.957
dGSLM	0.691	0.015	0.934
PersonaPlex	0.327	0.025	0.649
Proposed (Ours)	0.000	0.062	0.644

6.2 Qualitative Examples

We qualitatively compare our model against baselines on both the back-channel generation and turn scheduling tasks. For back-channel generation, we compare against PersonaPlex [Roy et al. 2026], the state-of-the-art full-duplex audio language model, using examples randomly sampled from the full-duplex benchmark. Figure 6 visualizes the voice activity of back-channels inserted by each system. PersonaPlex exhibits two recurring failure modes: either deferring the back-channel until after the speaker has finished (Example A), or taking over the floor during the speaker's turn (Example B). In contrast, our model consistently inserts back-channels either during the speaker's turn or within natural pauses, better reflecting the timing of human listener responses.

Turn Scheduling. As no dedicated turn-scheduling baseline exists, we compare against a pause-based heuristic that concatenates voice lines with a fixed inter-turn pause of 200 ms, an approach used in

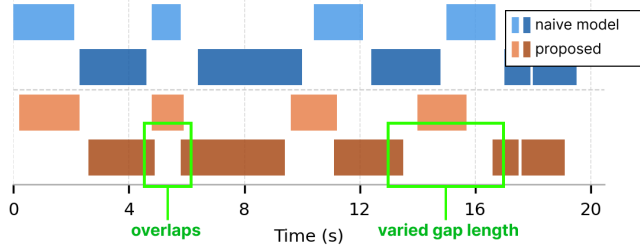


Figure 7: Voice activity timelines for one dialogue, comparing the turn scheduling of the naive pause-based baseline (top) with that of our proposed model (bottom).

prior work both for synthesizing two-channel speech datasets [Roy et al. 2026] and as an evaluation baseline [Nguyen et al. 2022; Wang et al. 2025]. We present examples across two settings: casual conversations synthesized with ChatterBox TTS [Resemble AI 2025], and more dramatic exchanges sourced from in-the-wild performances from movies and TV shows. As Figure 7 shows, the naive heuristic produces uniform inter-turn gaps throughout, whereas our scheduler generates varied timing with natural overlaps and pauses, yielding turn-taking dynamics that are markedly less mechanical.

6.3 User Study

While our qualitative examples allow readers to assess output quality, we further conduct a forced-choice preference study ($n = 20$ participants) to evaluate perceptual naturalness more rigorously. We showed 10 back-channeling examples, 5 turn-taking examples for conversational speech, and 5 for dramatic speech. In both back-channel and turn-taking conditions, participants listened to pairs of stimuli (ours vs. baseline) presented in random order and selected which sounded more natural, with instructions to attend to conversational timing rather than voice quality.

For the back-channel generation examples, our model was significantly preferred over PersonaPlex (67.5%, 95% CI [60.7%, 73.6%], $p < 0.001$). Preference for turn scheduling is more nuanced. For casual conversational examples, preference for our model over the naive baseline was at chance (53.0%, 95% CI [43.3%, 62.5%], $p = 0.617$), suggesting that timing differences are difficult to perceive in dialogue with relatively flat turn-taking dynamics. We suspect this reflects the nature of the test stimuli: casual chat and customer service exchanges tend to feature simple, sequential turn structure where the distinction between uniform and natural gaps is subtle. For more dramatic examples featuring real voice-actor recordings, where timing expressivity is more perceptually salient, preference for our model was significant (64.3%, 95% CI [52.6%, 74.5%], $p = 0.022$), suggesting that the scheduler’s benefits become perceptually apparent when the source material has richer conversational dynamics. Additional dramatic examples are presented in the supplementary video.

7 Discussion

7.1 Animator Control

A key advantage of the search-based inference formulation is that the scheduler’s behavior is directly controllable through its search

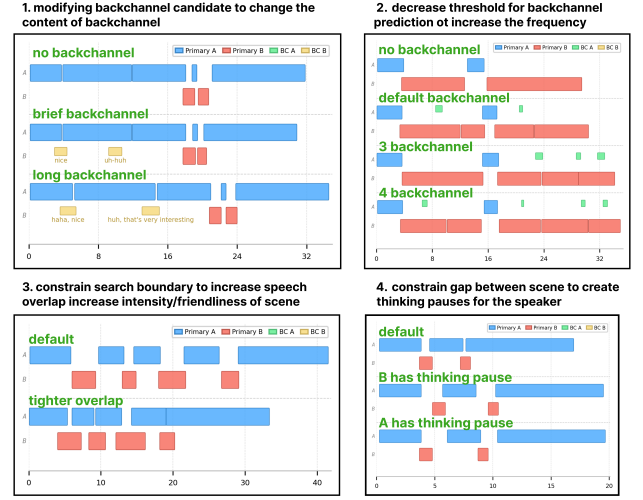


Figure 8: Four axes of animator control: (1) back-channel content, by swapping candidates; (2) back-channel frequency, by lowering the prediction threshold; (3) speech overlap, by tightening onset constraints; and (4) thinking pauses, by widening one speaker’s search window.

hyperparameters, without modifying model weights. We demonstrate four axes of control in Figure 8 and the supplementary video. The nature of back-channels can be varied by swapping the back-channel candidates, ranging from short unobtrusive acknowledgements (e.g., “nice”, “uh-huh”) to longer, semantically significant responses that cause the primary speaker’s turn to be retimed by Pass 3. The number of back-channels is controlled by adjusting the prediction threshold, with lower thresholds producing more frequent back-channels. Overlapping speech can be induced by tightening onset position constraints to force simultaneous activity on both channels. Finally, the search window can be widened for a speaker to introduce natural thinking pauses between turns.

7.2 Limitations

While the separation of timing and content is a core strength of our approach, it also limits how back-channels are chosen. The Back-channel Predictor decides whether a clip invites a back-channel and of which class, but not its wording: the back-channel itself is drawn from a vocabulary supplied by the animator. Our system therefore does not produce context-specific back-channels that are semantically grounded in what is being said, unlike voice-LLM-based methods such as PersonaPlex [Roy et al. 2026] which jointly model content and timing. This is a property of the Back-channel Predictor rather than of the Neural Scheduler, which simply schedules the clips it is given. The separation is deliberate: it keeps the content of every line, including back-channels, under the animator’s direct control, so that the nature and valence of back-channels can be specified by curating the candidate pool. The cost is that this requires manual intervention rather than automatic inference from context. A natural direction for future work is a hybrid approach that uses an LLM to select semantically appropriate back-channel candidates while delegating timing decisions to our scheduler.

A second limitation is that our model operates on fixed voice-line clips and cannot introduce speech disfluencies such as repetitions or stutters, which are also important components of natural conversation. Incorporating disfluency generation into the voice-line synthesis stage can be another promising direction for future work.

8 Conclusion

We presented the first controllable neural solution for combining independently recorded voice lines into dynamic conversations with natural turn-taking and back-channeling. By decoupling timing from content, our Neural Scheduler produces conversational dynamics that closely match ground truth while remaining directly controllable: overlap density, back-channel frequency, and character expressiveness can all be specified without any modification to model weights. We believe this separation of timing and content is an unexplored direction for conversational synthesis, with applications ranging from game dialogue production to training data generation for future voice assistants.

References

- Vasu Agrawal, Akinniyi Akinyemi, Kathryn Alvero, Morteza Behrooz, Julia Buffalini, Fabio Maria Carlucci, Joy Chen, Junming Chen, Zhang Chen, Shiyang Cheng, Praveen Chowdary, Joe Chuang, Antony D'Avirro, Jon Daly, Ning Dong, Mark Duppenhaler, Cynthia Gao, Jeff Girard, Martin Gleize, Sahir Gomez, Hongyu Gong, Srivathsan Govindarajan, Brandon Han, Sen He, Denise Hernandez, Yordan Hristov, Rongjie Huang, Hirofumi Inaguma, Somya Jain, Raj Janardhan, Qingyao Jia, Christopher Klaiber, Dejan Kovachev, Moneish Kumar, Hang Li, Yilei Li, Pavel Litvin, Wei Liu, Guangyao Ma, Jing Ma, Martin Ma, Xutai Ma, Lucas Mantovani, Sagar Miglani, Sreyas Mohan, Louis-Philippe Morency, Evonne Ng, Kam-Woh Ng, Tu Anh Nguyen, Amia Oberai, Benjamin Peloquin, Juan Pino, Jovan Popovic, Omid Poursaeed, Fabian Prada, Alice Rakotoarison, Rakesh Ranjan, Alexander Richard, Christophe Ropers, Safiyyah Saleem, Vasu Sharma, Alex Shcherbyna, Jia Shen, Jie Shen, Anastasis Stathopoulos, Anna Sun, Paden Tomasello, Tuan Tran, Arina Turkatenko, Bo Wan, Chao Wang, Jeff Wang, Mary Williamson, Carleigh Wood, Tao Xiang, Yilin Yang, Julien Yao, Chen Zhang, Jiemin Zhang, Xinyue Zhang, Jason Zheng, Pavlo Zhyzheria, Jan Zikes, and Michael Zollhoefer. 2025. Seamless Interaction: Dyadic Audiovisual Motion Modeling and Large-Scale Dataset. arXiv:2506.22554 [cs.CV] <https://arxiv.org/abs/2506.22554>
- Anderson et al. 2025. NaturalTurn: a method to segment speech into psychologically meaningful conversational turns. *Scientific Reports* (2025). <https://www.nature.com/articles/s41598-025-24381-1>
- Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2023a. AudioLM: A Language Modeling Approach to Audio Generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), 2523–2533. <https://arxiv.org/abs/2209.03143>
- Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. 2023b. SoundStorm: Efficient Parallel Audio Generation. arXiv preprint arXiv:2305.09636. <https://arxiv.org/abs/2305.09636>
- Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. 2024a. VALL-E 2: Neural Codec Language Models are Human Parity Zero-Shot Text to Speech Synthesizers. arXiv preprint arXiv:2406.05370. <https://arxiv.org/abs/2406.05370>
- Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2024b. F5-TTS: A Fairytale that Fakes Fluent and Faithful Speech with Flow Matching. arXiv preprint arXiv:2410.06885. <https://arxiv.org/abs/2410.06885>
- Christopher Cieri, David Miller, and Kevin Walker. 2004. The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva (Eds.). European Language Resources Association (ELRA), Lisbon, Portugal. <https://aclanthology.org/L04-1500/>
- Herbert H. Clark and Susan E. Brennan. 1991. Grounding in communication. In *Perspectives on Socially Shared Cognition*, Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley (Eds.). APA Books, 127–149.
- Herbert H. Clark and Edward F. Schaefer. 1989. Contributing to Discourse. *Cognitive Science* 13, 2 (1989), 259–294. doi:10.1207/s15516709cog1302_7
- Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High Fidelity Neural Audio Compression. arXiv preprint arXiv:2210.13438. <https://arxiv.org/abs/2210.13438>
- Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Mosh: A Speech-Text Foundation Model for Real-Time Dialogue. arXiv:2410.00037 [eess] doi:10.48550/arXiv.2410.00037
- 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. arXiv preprint arXiv:2407.05407. <https://arxiv.org/abs/2407.05407>
- Pif Edwards, Chris Landreth, Mateusz Poplawski, Robert Malinowski, Sarah Watling, Eugene Fiume, and Karan Singh. 2020. JALI-Driven Expressive Facial Animation and Multilingual Speech in Cyberpunk 2077. In *ACM SIGGRAPH 2020 Talks*. ACM, Virtual Event USA, 1–2. doi:10.1145/3388767.3407339
- Erik Ekstedt and Gabriel Skantze. 2020. TurnGPT: a Transformer-based Language Model for Predicting Turn-taking in Spoken Dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2981–2990. <https://arxiv.org/abs/2010.10874>
- Erik Ekstedt and Gabriel Skantze. 2022. Voice Activity Projection: Self-supervised Learning of Turn-taking Events. In *Proc. Interspeech 2022*. 5190–5194. <https://arxiv.org/abs/2205.09812>
- Yoshinori Fukunaga, Ryota Nishimura, Kengo Ohta, and Norihide Kitaoka. 2025. Backchannel Prediction for Natural Spoken Dialog Systems Using General Speaker and Listener Information. In *Interspeech 2025*. ISCA, 1078–1082. doi:10.21437/Interspeech.2025-1316
- John Heritage. 1984. A change-of-state token and aspects of its sequential placement. In *Structures of Social Action: Studies in Conversation Analysis*, J. Maxwell Atkinson and John Heritage (Eds.). Cambridge University Press, 299–345. doi:10.1017/CBO9780511665868.020
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhota, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. arXiv:2106.07447 [cs] doi:10.48550/arXiv.2106.07447
- Koji Inoue, Bing'er Jiang, Erik Ekstedt, Tatsuya Kawahara, and Gabriel Skantze. 2024. Real-time and Continuous Turn-taking Prediction Using Voice Activity Projection. arXiv preprint arXiv:2401.04868. <https://arxiv.org/abs/2401.04868>
- Zequan Ju, Dongchao Yang, Jianwei Yu, Kai Shen, Yichong Leng, Zhengtao Wang, Xu Tan, Xinyu Zhou, Tao Qin, and Xiangyang Li. 2025. MoonCast: High-Quality Zero-Shot Podcast Generation. arXiv preprint arXiv:2503.14345. <https://arxiv.org/abs/2503.14345>
- Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. In *International Conference on Machine Learning (ICML)*. <https://arxiv.org/abs/2106.06103>
- Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*. <https://arxiv.org/abs/2010.05646>
- Yinghao Aaron Li, Cong Han, Vinay S. Raghavan, Gavin Mischler, and Nima Mesgarani. 2023. StyleTTS 2: Towards Human-Level Text-to-Speech through Style Diffusion and Adversarial Training with Large Speech Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*. <https://arxiv.org/abs/2306.07691>
- Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H. Liu, and Hung-yi Lee. 2025b. Full-Duplex-Bench: A Benchmark to Evaluate Full-duplex Spoken Dialogue Models on Turn-taking Capabilities. arXiv:2503.04721 [cs] doi:10.48550/arXiv.2503.04721
- Yuxin Lin et al. 2025a. Predicting Turn-Taking and Backchannel in Human-Machine Conversations Using Linguistic, Acoustic, and Visual Signals. In *ACL*. <https://arxiv.org/abs/2505.12654>
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. (Sept. 2019).
- Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2024. Language Model Can Listen While Speaking. arXiv:2408.02622 [cs] doi:10.48550/arXiv.2408.02622
- Heidi McConaghie. 2021. Correlations Between Cognitive Pause Patterns and Listener Perceptions of Communicative Effectiveness and Likeability for People With Aphasia. *Theses and Dissertations* (June 2021).
- Akira Morikawa, Ryo Ishii, Hajime Noto, Atsushi Fukayama, and Takao Nakamura. 2022. Determining Most Suitable Listener Backchannel Type for Speaker's Utterance. In *Proceedings of the 22nd ACM International Conference on Intelligent Virtual Agents (IVA '22)*. Association for Computing Machinery, New York, NY, USA, 1–3. doi:10.1145/3514197.3549619
- Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoit Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux. 2022. Generative Spoken Dialogue Language Modeling. *Transactions of the Association for Computational Linguistics* (2022). <https://arxiv.org/abs/2203.16502>
- Toshiki Onishi, Naoki Azuma, Shunichi Kinoshita, Ryo Ishii, Atsushi Fukayama, Takao Nakamura, and Akihiro Miyata. 2023. Prediction of Various Backchannel Utterances

- Based on Multimodal Information. In *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents (IVA '23)*. Association for Computing Machinery, New York, NY, USA, 1–4. doi:10.1145/3570945.3607298
- Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. 2024. VoiceCraft: Zero-Shot Speech Editing and Text-to-Speech in the Wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*. <https://arxiv.org/abs/2403.16973>
- Pixel Audio Inc. 2026. Pixel Audio: Full Spectrum Audio Design & Immersive Sound Design. <https://pixelaudio.ca/>. Accessed: May 9, 2026.
- Kevin Redding. 2022. How to Become a Video Game Voice Actor. <https://www.backstage.com/magazine/article/video-game-voice-actor-guide-75034/>.
- Andrew G. Reece et al. 2023. The CANDOR corpus: Insights from a large multimodal dataset of naturalistic conversation. *Science Advances* (2023). doi:10.1126/sciadv.adf3197
- Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. In *International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/2006.04558>
- Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. FastSpeech: Fast, Robust and Controllable Text to Speech. In *Advances in Neural Information Processing Systems (NeurIPS)*. <https://arxiv.org/abs/1905.09263>
- Resemble AI. 2025. Chatterbox-TTS. <https://github.com/resemble-ai/chatterbox>. GitHub repository.
- Edward Reynolds and Johanna Rendle-Short. 2011. Cues to Deception in Context: Response Latency/Gaps in Denials and Blame Shifting. *The British Journal of Social Psychology* 50, 3 (Sept. 2011), 431–449. doi:10.1348/014466610X520104
- Rajarshi Roy, Jonathan Raiman, Sang-gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. 2026. PersonaPlex: Voice and Role Control for Full Duplex Conversational Speech Models. arXiv:2602.06053 [cs] doi:10.48550/arXiv.2602.06053
- Emanuel A. Schegloff. 1982. Discourse as an interactional achievement: Some uses of 'uh huh' and other things that come between sentences. *Semiotica* 40 (1982), 71–93.
- Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. <https://arxiv.org/abs/1712.05884>
- Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. 2024. NaturalSpeech 2: Latent Diffusion Models are Natural and Zero-Shot Speech and Singing Synthesizers. In *International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/2304.09116>
- Tanya Stivers, N. J. Enfield, Penelope Brown, et al. 2009. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences* 106, 26 (2009), 10587–10592. doi:10.1073/pnas.0903616106
- Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. 2021. A Survey on Neural Speech Synthesis. arXiv preprint arXiv:2106.15561. <https://arxiv.org/abs/2106.15561>
- Gemini Team. 2025. Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805 [cs.CL] <https://arxiv.org/abs/2312.11805>
- Silero Team. 2024. Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier. <https://github.com/snakers4/silero-vad>.
- Jackson Tolins and Jean E. Fox Tree. 2014. Addressee backchannels steer narrative development. *Journal of Pragmatics* 70 (2014), 152–164. doi:10.1016/j.pragma.2014.06.006
- Carol Trimboli and Michael B. Walker. 1984. Switching pauses in cooperative and competitive conversations. *Journal of Experimental Social Psychology* (1984). doi:10.1016/0022-1031(84)90027-1
- Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. In *9th ISCA Workshop on Speech Synthesis (SSW9)*. <https://arxiv.org/abs/1609.03499>
- 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*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 21390–21402. doi:10.18653/v1/2024.emnlp-main.1192
- Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023). <https://arxiv.org/abs/2301.02111>
- Jinhan Wang, Long Chen, Aparna Khare, Anirudh Raju, Pranav Dheram, Di He, Minhua Wu, Andreas Stolcke, and Venkatesh Ravichandran. 2024. Turn-Taking and Backchannel Prediction with Acoustic and Large Language Model Fusion. arXiv:2401.14717 [cs] doi:10.48550/arXiv.2401.14717
- 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. arXiv:2506.00975 [cs] doi:10.48550/arXiv.2506.00975
- Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. 2017. Tacotron: Towards End-to-End Speech Synthesis. In *Proceedings of Interspeech*. <https://arxiv.org/abs/1703.10135>
- Zhifei Xie and Changqiao Wu. 2024. Mini-Omni: Language Models Can Hear, Talk While Thinking in Streaming. arXiv:2408.16725 [cs] doi:10.48550/arXiv.2408.16725
- Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. 2025. Qwen3-Omni Technical Report. arXiv:2509.17765 [cs] doi:10.48550/arXiv.2509.17765
- Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I. Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. 2021. SUPERB: Speech Processing Universal PERFORMANCE Benchmark. arXiv:2105.01051 [cs] doi:10.48550/arXiv.2105.01051
- Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2022. SoundStream: An End-to-End Neural Audio Codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2022), 495–507. <https://arxiv.org/abs/2107.03312>
- Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. GLM-4-Voice: Towards Intelligent and Human-Like End-to-End Spoken Chatbot. arXiv:2412.02612 [cs] doi:10.48550/arXiv.2412.02612
- Leying Zhang et al. 2025. CoVoMix2: Advancing Zero-Shot Dialogue Generation with Fully Non-Autoregressive Flow Matching. arXiv preprint arXiv:2506.00885. <https://arxiv.org/abs/2506.00885>