CHORUS: EXTENDING SINGLE-SPEAKER TTS TO MULTI-SPEAKER CONVERSATION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-speaker conversational speech generation synthesizes a conversation among several speakers from its transcript. Most existing systems generate the conversation as a single-channel mixture, which entangles the speakers. This makes the generated speech difficult to use for downstream tasks that require a separate channel for each speaker. It also complicates per-speaker evaluation, which requires diarization to attribute each speech segment to the corresponding speaker. Systems that generate one channel per speaker address these issues but are limited to two speakers and one form of timing control. We introduce *Chorus*, which assigns a weight-shared single-speaker text-to-speech (TTS) model to each speaker and connects the models through a communication block whose computation does not depend on the number of speakers. Chorus therefore supports an arbitrary number of speakers, chosen at inference time. Chorus retains the infilling objective of its base model, F5-TTS. Each model receives the audio of its channel, a mask over the frames to generate, and a text input in which special tokens mark the turns of that speaker. Within this formulation, four tasks are obtained by choosing which frames are masked and how the transcript is structured: generation of all channels from a transcript, generation of one channel conditioned on the recorded audio of the others, chunk generation of long conversations conditioned on the previous chunk, and generation with turn timing specified either by explicit start and end times or by the turn order alone. On the two-speaker ZipVoice-Dialog test set, Chorus is competitive with the baselines on intelligibility and speaker similarity, although it is fine-tuned on less conversational speech. From the turn order alone, it produces backchannels and interruptions whose timing a turn-taking judge scores above the ground truth. The same model generates two- to four-speaker conversations on AMI with a rise of one point in per-channel word error rate, making it, to our knowledge, the first system that generates one channel per speaker for more than two speakers. Demo page: `https://anonymous.4open.science/w/ICLR2027-Chorus-Demo-F324/`.

1 INTRODUCTION

Zero-shot text-to-speech (TTS) models synthesize natural speech in the voice of an unseen speaker, given only a short audio prompt (Chen et al., 2025a;b). Multi-speaker conversational speech generation extends this capability to generate an entire conversation among multiple speakers, with applications in content creation such as podcasts and audio dramas. Generated conversations can also serve as training data for systems that must handle multi-party speech. Speaker diarization models are often trained on simulated conversations (Fujita et al., 2019a;b; Landini et al., 2022), and synthetic conversations have been used to train conversational speech recognition (Cornell et al., 2024). Full-duplex spoken dialogue models are fine-tuned on conversational data with one channel per speaker, which is scarce in practice (Défossez et al., 2024; Nguyen et al., 2023). These models must learn when to take the floor, when to backchannel, and when to yield to an interruption. Such dynamics are only observable in data that preserves overlap and keeps each speaker on a separate channel. A generator that produces such data from text alone would relieve this bottleneck. This has led to a growing interest in multi-speaker conversational speech generation (Nguyen et al., 2023; Zhang et al., 2024; 2025; Ju et al., 2025; Zhu et al., 2026; Peng et al., 2025; Xie et al., 2025; Zhang et al., 2026).

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

Despite this progress, existing systems fall short of this promise in one or more respects. Most generate a single-channel mixture in which the speakers' voices are entangled (Zhang et al., 2024; Ju et al., 2025; Xie et al., 2025; Zhang et al., 2026). The systems that do generate one channel per speaker fix the number of speakers at two (Nguyen et al., 2023; Zhu et al., 2026). Each system also commits to one form of timing control, either a time-aligned input that fixes the timing before generation (Zhang et al., 2025) or the turn order alone, which gives the user no control over timing.

We introduce *Chorus*, which extends a pretrained single-speaker TTS model to multi-speaker conversational speech generation. Chorus assigns a weight-shared TTS model to each speaker, with each model generating one audio channel. The models communicate through a communication block that is invariant to the number of speakers, instantiated with Transform-Average-Concatenate (TAC) blocks (Luo et al., 2020). Because the same communication function can be applied to any number of channels, the number of speakers can be chosen at inference time. We instantiate the single-speaker TTS model with F5-TTS (Chen et al., 2025b), which generates a masked span of a mel spectrogram from its transcript (Eskimez et al., 2024). Chorus keeps this infilling objective and applies it to all channels simultaneously. Each model receives the audio of its own channel, a mask indicating the frames to generate, and a text input whose special tokens mark whether each turn belongs to that speaker. The mask and text input determine the task, allowing the same model to support several generation settings without changing its architecture. When one channel is left unmasked, the other channels are generated conditioned on the given audio, which we call response generation. When the previously generated chunk is placed before the masked region, the next chunk continues the conversation, enabling generation beyond a single pass. The text input also determines how turn timing is controlled. In timestamp mode, it is aligned frame by frame with the audio, so the position of each turn specifies its start and end. In turn-order mode, it specifies only the order of turns, allowing Chorus to determine their timing, including backchannels and interruptions.

Our contributions are as follows:

- We propose Chorus, which assigns a weight-shared single-speaker TTS model to each speaker and connects the models through a communication block that is invariant to the number of speakers. To our knowledge, Chorus is the first to generate one channel per speaker for more than two speakers.
- We formulate conversation generation as a single infilling task, so that one model supports long-form generation and response generation, and offers two modes of timing control.
- On the two-speaker ZipVoice-Dialog test set, Chorus is competitive with baselines on intelligibility and speaker similarity, although it is fine-tuned on less conversational speech. The same model generates two-, three-, and four-speaker conversations on AMI with a rise of only one point in per-channel word error rate. In timestamp mode, the voice activity of the generated channels matches the given timestamps with an F1 of 0.93 at 10 ms resolution. In response generation, where one channel is a real recording, the overlap between the generated and the given channel approaches that of the real conversation.

2 Related work

Single-channel conversation generation. Most existing systems generate a conversation as a single-channel mixture in which the speakers' voices are entangled (Zhang et al., 2024; 2025; Ju et al., 2025; Peng et al., 2025; Xie et al., 2025; Zhang et al., 2026; Zhu et al., 2026). The mixture provides no per-speaker labels for downstream training and complicates per-speaker evaluation. Several of these systems also identify speakers with speaker-indexed tags such as `[S1]` and `[S2]` (Peng et al., 2025; Xie et al., 2025; Zhang et al., 2026), so supporting more speakers requires adding more speaker-indexed tags. In contrast, Chorus generates one channel per speaker, and its special tokens encode turns relative to each speaker, so adding a speaker requires no new token.

Two-speaker conversation generation. dGSLM (Nguyen et al., 2023) was the first to generate one channel per speaker, using two weight-shared transformers that communicate through pairwise cross-attention. It is textless and, like the full-duplex spoken dialogue models Moshi (Défossez et al., 2024) and PersonaPlex (Roy et al., 2026), which assign one fixed channel each to a user and an agent, it continues a conversation from audio rather than synthesizing a given transcript.

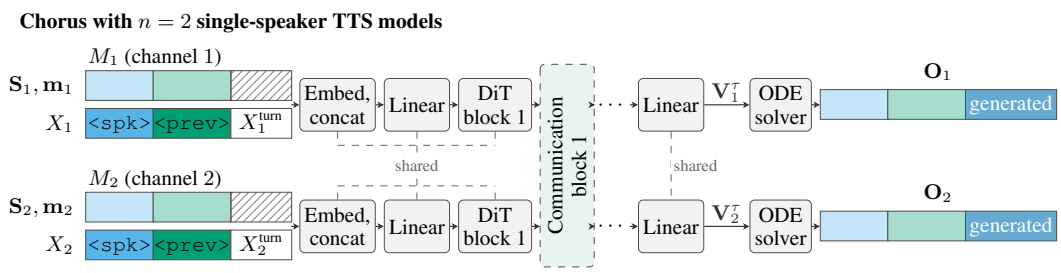

Figure 1: Chorus with two weight-shared single-speaker TTS models M_1 and M_2, instantiated with F5-TTS. Each receives the text input X_i, the acoustic input $\mathbf{S}_i$, and the mask $\mathbf{m}_i$. The communication block after each Diffusion Transformer (DiT) block is defined in Eq. (3).

ZipVoice-Dialog-Stereo (Zhu et al., 2026) does synthesize a given transcript with one channel per speaker, but like all of these systems it fixes the number of channels at two, so extending them to more speakers requires a new communication mechanism and retraining. In contrast, Chorus synthesizes a given transcript, and its communication block is invariant to the number of speakers.

Timing control. CoVoMix2 (Zhang et al., 2025) takes a time-aligned text input in which each turn is anchored to its start and end time. Without given times, it derives them from syllable counts and a fixed speaking rate, so the model never chooses the timing. The other systems (Zhu et al., 2026; Xie et al., 2025; Ju et al., 2025) take only the turn order and give the user no control over timing. Chorus is fine-tuned in both a turn-order mode and a timestamp mode, so one model can do both.

Number-invariant multi-channel processing. Blocks whose computation does not depend on the number of inputs are standard in multi-channel speech processing, where the number of microphones varies across recordings. Early neural mask-based beamforming applied one weight-shared network to each microphone and pooled the resulting masks by a median (Heymann et al., 2016) or maximum (Erdogan et al., 2016), including in systems jointly trained with end-to-end automatic speech recognition (ASR) (Ochiai et al., 2017a;b). Later work moved this pooling inside the network. The TAC block (Luo et al., 2020) averages transformed features across microphones between network layers, while other approaches use attention (Wang et al., 2020; Chang et al., 2021; Yu et al., 2023), graph convolution (Tzirakis et al., 2021), or co-attention followed by averaging (Horiguchi et al., 2022). Chorus applies this principle to speakers instead of microphones. Each channel is the output of one speaker, and the pooled information coordinates their turns rather than enhancing a common source.

3 Method

3.1 Preliminaries: F5-TTS

F5-TTS (Chen et al., 2025b) is a single-speaker TTS model based on speech infilling. Given an acoustic input, its transcript, and a mask indicating the frames to generate, F5-TTS generates the masked speech. Chorus retains this formulation and uses one F5-TTS model for each speaker. Figure 1 shows two F5-TTS models for two speakers, where the subscript i identifies the speaker and its model. Below, we omit this subscript and write $\mathbf{S}$, X, and $\mathbf{m}$ for the acoustic input, raw transcript, and mask of a single model, respectively.

Inputs. F5-TTS takes three inputs, shown as the bars on the left of each model in Figure 1. The acoustic input $\mathbf{S} \in \mathbb{R}^{B \times F}$ is a mel spectrogram with B frequency bins and F frames. The mask $\mathbf{m} \in \{0, 1\}^F$ indicates which frames are masked and should be generated, with $m_t = 1$ for a masked frame. The raw transcript $X \in \mathcal{V}^L$ is a sequence of L characters from the vocabulary $\mathcal{V}$. F5-TTS pads X with the filler token <fill> to F tokens, so that each acoustic frame has one corresponding text token. The characters of X are not explicitly aligned to the frames of $\mathbf{S}$, and F5-TTS learns this alignment during training.

Flow matching. To generate the masked frames of $\mathbf{S}$, F5-TTS uses conditional flow matching (Lipman et al., 2023), which transforms a Gaussian noise sample $\mathbf{Z}^0 \in \mathbb{R}^{B \times F}$ into the target mel spectrogram $\mathbf{S}$ along a learned vector field. At flow step $\tau \in [0, 1]$, the noisy input lies on the straight

path from $\mathbf{Z}^0$ to $\mathbf{S}$,

$$\mathbf{Z}^\tau = (1-\tau)\mathbf{Z}^0 + \tau\mathbf{S}, \tag{1}$$

whose vector field is $\mathbf{S}-\mathbf{Z}^0$. The network receives $\mathbf{Z}^\tau$, the acoustic input $\mathbf{S}$ with its masked frames set to zero, the raw transcript X, and τ. From these, it predicts the vector field $\mathbf{V}^\tau \in \mathbb{R}^{B\times F}$. Following the blocks of Figure 1, the padded X is embedded and concatenated frame by frame with $\mathbf{S}$ and $\mathbf{Z}^\tau$. The result is projected to the hidden dimension D and processed by a stack of Diffusion Transformer (DiT) blocks (Peebles & Xie, 2023). The output of the l-th DiT block is the hidden representation $\mathbf{H}^l \in \mathbb{R}^{D\times F}$. A final linear layer outputs $\mathbf{V}^\tau$. Training minimizes the squared error between $\mathbf{V}^\tau$ and the vector field of Eq. (1) on the masked frames,

$$\mathcal{L} = \mathbb{E}_{\tau,\mathbf{Z}^0}\left[\left\|\left(\mathbf{V}^\tau - \mathbf{S} + \mathbf{Z}^0\right)\operatorname{diag}(\mathbf{m})\right\|^2\right], \tag{2}$$

where $\operatorname{diag}(\mathbf{m})$ selects the masked frames and $\|\cdot\|$ denotes the Frobenius norm. The mask covers a random span of a real utterance, so that the unmasked frames act as an audio prompt.

Inference. The unmasked frames of $\mathbf{S}$ contain the audio prompt, X is the prompt transcript followed by the text to synthesize, and the number of masked frames is estimated from the number of characters to synthesize. Starting from $\mathbf{Z}^0$, F5-TTS solves the ordinary differential equation (ODE) defined by $\mathbf{V}^\tau$ from $\tau = 0$ to $\tau = 1$ in a fixed number of steps with classifier-free guidance (Ho & Salimans, 2021), and the masked frames of the result are the generated speech.

3.2 Communicating single-speaker TTS models

Chorus assigns a single-speaker TTS model to each speaker, so that the number of speakers is not fixed by the architecture. For a conversation with $n \in \mathbb{N}$ speakers, we instantiate n weight-shared models $M_1, \ldots, M_n$ of a single-speaker TTS model M, where M_i generates the speech of speaker i on channel i. As shown in Figure 1, each M_i receives the three inputs of Section 3.1 for its own channel, the text input $X_i \in \mathcal{V}'^L$, whose special tokens (Section 3.3) extend $\mathcal{V}$ to $\mathcal{V}'$, the acoustic input $\mathbf{S}_i \in \mathbb{R}^{B\times F}$, and the mask $\mathbf{m}_i \in \{0,1\}^F$. Each M_i fills in the masked frames of its own channel as in Section 3.1, which gives the output mel spectrogram $\mathbf{O}_i \in \mathbb{R}^{B\times F}$. However, each model generates its channel independently, so the speakers cannot coordinate their speech.

We therefore insert a communication block between the layers of M. To keep Chorus invariant to the number of speakers, the communication block itself must not depend on n. Such blocks are standard in multi-channel speech processing (Section 2), where a weight-shared network processes every microphone and its outputs are pooled over the microphones by a median (Heymann et al., 2016), a maximum (Erdogan et al., 2016), an average (Luo et al., 2020), or attention (Horiguchi et al., 2022). We use the TAC block as the communication block. Let $\mathbf{H}_i^l \in \mathbb{R}^{D\times F}$ denote the hidden representation $\mathbf{H}^l$ of Section 3.1 in the model M_i, after the l-th DiT block. The TAC block updates $\{\mathbf{H}_1^l, \ldots, \mathbf{H}_n^l\}$ in three steps:

$$\hat{\mathbf{H}}_i^l = f_{\mathrm{T}}(\mathbf{H}_i^l), \qquad \bar{\mathbf{H}}^l = f_{\mathrm{A}}\Big(\frac{1}{n}\sum_{j=1}^{n}\hat{\mathbf{H}}_j^l\Big), \qquad \mathbf{R}_i^l = f_{\mathrm{C}}\big([\hat{\mathbf{H}}_i^l; \bar{\mathbf{H}}^l]\big), \tag{3}$$

where $[\cdot;\cdot]$ denotes concatenation along the feature dimension. f_{T}, f_{A}, and f_{C} are fully connected layers with PReLU activation, shared across the n models. f_{C} maps the concatenated $2D$ features back to D, so that $\hat{\mathbf{H}}_i^l$, $\bar{\mathbf{H}}^l$, and $\mathbf{R}_i^l$ all lie in $\mathbb{R}^{D\times F}$. A gated residual connection then adds $\mathbf{R}_i^l$ back, $\tilde{\mathbf{H}}_i^l = \mathbf{H}_i^l + g^l\mathbf{R}_i^l$, where $g^l \in \mathbb{R}$ is a learnable scalar of the l-th TAC block, shared across models, and $\tilde{\mathbf{H}}_i^l$ replaces $\mathbf{H}_i^l$ as the input to the next layer. We insert one TAC block after every DiT block of F5-TTS and initialize every g^l to 0, so each M_i starts fine-tuning as the pretrained M.

3.3 Text input construction with special tokens

The text input X_i of Section 3.2 tells the model M_i of speaker i which turns are its own and when to stay silent. We construct it in one of two modes, both shown in Figure 2(a). In *turn-order mode*, X_i is a character sequence that carries only the order of the turns, and Chorus decides all timing. In *timestamp mode*, X_i is aligned frame by frame to the acoustic input $\mathbf{S}_i$, and the position of each turn in X_i explicitly fixes its timing. Both modes use five special tokens, `<turn>`, `<other>`, `<turn_fill>`, `<spk>`, and `<prev>`, which extend the vocabulary $\mathcal{V}$ to $\mathcal{V}'$. In both modes, $\mathbf{S}_i$

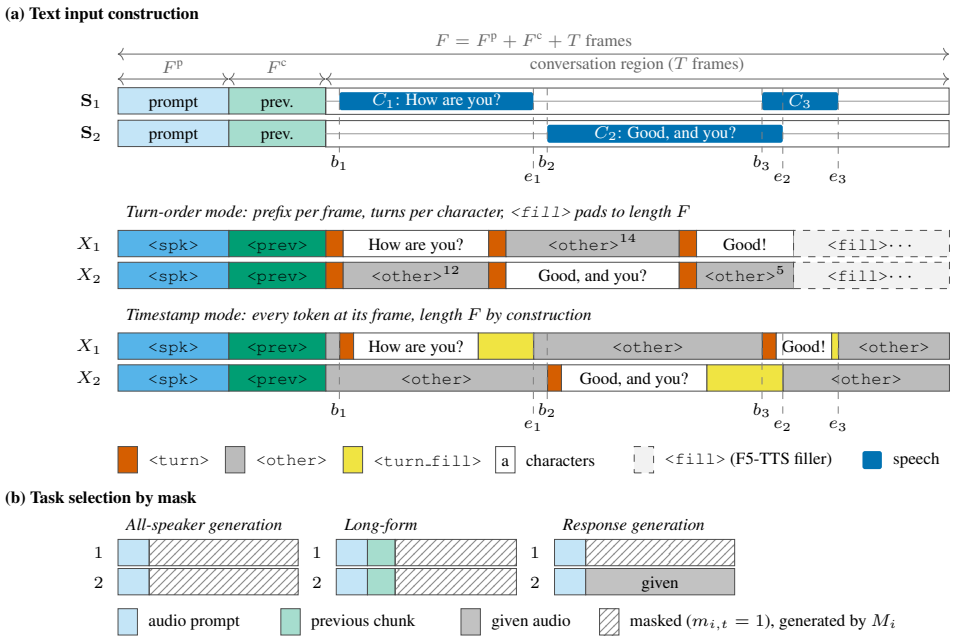

Figure 2: (a) Text input construction (Section 3.3) for a two-speaker conversation with three turns, where speaker 1 interrupts the end of C_2. (b) The mask of each task (Section 3.5).

consists of a prefix, which holds the audio prompt and, in long-form generation, the previous chunk, followed by the conversation region of $T \in \mathbb{N}$ frames.

Turn-order mode. Chorus orders the turns by their start times, and in turn-order mode the text input carries only this order. A backchannel or an interruption is therefore an ordinary turn, placed after the turn during which it starts, as C_3 in Figure 2(a). The <turn> token marks the start of each turn, and the <other> token fills the positions where another speaker is talking, signaling M_i to stay silent. Formally, for $K \in \mathbb{N}$ turns with raw transcripts $C_k \in \mathcal{V}^{|C_k|}$ and speakers $s_k \in \{1, \dots, n\}$, the turn sequence of M_i is

$$X_i^{\text{turn}} = \big(\texttt{<turn>}, \phi_i(C_1), \texttt{<turn>}, \dots, \phi_i(C_K)\big), \qquad \phi_i(C_k) = \begin{cases} C_k, & s_k = i, \\ \texttt{<other>}^{|C_k|}, & s_k \neq i, \end{cases} \tag{4}$$

where $\texttt{<other>}^{|C_k|}$ repeats <other> $|C_k|$ times, so that $\phi_i(C_k) \in \mathcal{V}'^{|C_k|}$ and X_i^{turn} has $K + \sum_{k=1}^{K} |C_k|$ tokens.

Timestamp mode. Timestamp mode lays the turn sequence of Eq. (4) out over frames, as in the last two rows of Figure 2(a). Let turn k span the frames from b_k to e_k, counted from the start of the conversation region, with $b_k, e_k \in \mathbb{N}$ and $e_k - b_k \geq |C_k|$. On the text input of speaker s_k, frame b_k carries <turn>, the next $|C_k|$ frames carry C_k, and the remaining frames of the turn, up to e_k, carry <turn_fill>. A turn therefore ends at its own e_k rather than at the start of the next turn, so overlapping turns have intersecting spans. Every other frame of the conversation region carries <other>, whether another speaker is talking or no one is, so M_i stays silent outside its own turns, which gives the turn sequence X_i^{turn} of timestamp mode. The interruption of C_3 in Figure 2(a) is therefore explicit: <turn> of X_1 appears at b_3 while <turn_fill> of X_2 continues until e_2.

Prompt and previous-chunk tokens. In both modes, the turn sequence is preceded by a prefix. As Figure 2(a) shows, the audio prompt occupies the first $F^{\mathrm{p}} \in \mathbb{N}$ frames of $\mathbf{S}_i$. In long-form generation (Section 3.5), the previous chunk occupies the next $F^{\mathrm{c}} \in \mathbb{N}$ frames, and $F^{\mathrm{c}} = 0$ otherwise. The conversation region holds the last T frames, so $F = F^{\mathrm{p}} + F^{\mathrm{c}} + T$. We mark the prefix with one token per frame and prepend the result to the turn sequence X_i^{turn} of the chosen mode: $X_i = \big(\texttt{<spk>}^{F^{\mathrm{p}}}, \texttt{<prev>}^{F^{\mathrm{c}}}, X_i^{\text{turn}}\big)$, so the prefix is identical on every channel. In turn-order mode, <fill> pads X_i to the F frames of $\mathbf{S}_i$, as in F5-TTS.

3.4 MULTI-TASK FINE-TUNING

We initialize Chorus from a pretrained F5-TTS checkpoint and fine-tune it on multi-speaker conversational data. The TAC blocks of Eq. (3) and the embeddings of the five special tokens of Section 3.3 are randomly initialized. For each fine-tuning sample, the text inputs are constructed in timestamp mode from the turn timestamps of the corpus with probability 0.5, and otherwise in turn-order mode. The previous chunk and the audio prompt are each dropped with probability 0.5 for classifier-free guidance. As in F5-TTS (Section 3.1), we mask a random span of the conversation region with the same mask on all n channels, so that the channels remain synchronized. With probability 0.3, one channel chosen uniformly at random is instead left fully unmasked. Chorus then reconstructs a speaker's speech given the complete audio of another, as in response generation at inference. The flow-matching loss of Eq. (2) is computed on the masked frames of each channel, and all parameters are fine-tuned jointly. Fine-tuning proceeds in two stages with the same objective and sampling. The first uses two-speaker corpora and single-speaker data, so $n \in \{1, 2\}$, and the second continues from it and adds conversations with more than two speakers.

3.5 INFERENCE

Every task sets the acoustic input $\mathbf{S}_i$, the mask $\mathbf{m}_i$ (Figure 2(b)), and the text input X_i of Section 3.2 and then samples the masked frames. In *all-speaker generation*, every channel is masked over the whole conversation region, so the speech of all speakers is generated from the text input alone. The *prompt recording* of speaker i is a short recording of that speaker alone, with $d_i \in \mathbb{N}$ frames and transcript $P_i \in \mathcal{V}^{|P_i|}$, and its mel spectrogram occupies the first F^{p} frames of $\mathbf{S}_i$. Following F5-TTS, we set the length T of the conversation region before sampling, using the frame-per-character rate $d_i/|P_i|$ of each prompt recording, as $T = \sum_{i=1}^{n} \frac{d_i}{|P_i|} \sum_{k:s_k=i} |C_k|$, rounded to the nearest frame, where C_k and s_k are the raw transcript and speaker of turn k (Section 3.3).

Long-form generation. A conversation whose estimated duration exceeds 45 seconds is split greedily into chunks of at most 45 seconds and generated chunk by chunk. For every chunk after the first, the last 5 seconds of the previously generated chunk are given as the previous chunk (Figure 2(b)) without their text, which preserves the voices and the flow of the conversation across chunks.

Response generation. Given the complete audio of some speakers, Chorus generates the channels of the others. Each given channel is left unmasked and each target channel is masked over the whole conversation region (Figure 2(b)).

4 EXPERIMENTAL SETTING

4.1 DATASETS

Fine-tuning data. The first stage (Section 3.4) uses three two-speaker conversational corpora, Fisher (Cieri et al., 2004), SSSD (Sheikh et al., 2025), and CANDOR (Reece et al., 2023), and the train subsets of the single-speaker corpus LibriTTS (Zen et al., 2019), treated as conversations with $n = 1$. Appendix C checks that this single-speaker capability is retained. The four corpora total about 2.6k hours. The second stage adds NOTSOFAR (Vinnikov et al., 2024), 7 hours of meetings with 4 to 8 speakers each. The preprocessing details are described in Appendix A.

Evaluation data. We evaluate on four test sets, none of which is included in the fine-tuning data. (1) The English test set of ZipVoice-Dialog (Zhu et al., 2026) contains 280 two-speaker conversations with one prompt recording per speaker and serves the two-speaker comparison and response generation. (2) For more than two speakers, the AMI meeting corpus test set (Carletta et al., 2006) is cut at turn boundaries into segments of 15 to 45 seconds and grouped by the number of active speakers into two-, three-, and four-speaker subsets of about 3.5, 4.0, and 2.0 hours. Since the AMI headset channels contain cross-talk, each speaker of a segment is paired with a distinct random LibriTTS test-clean speaker, whose utterance serves as the prompt recording. (3) The CoVoMix2 test set (Zhang et al., 2025) pairs DailyDialog scripts (Li et al., 2017) with LibriSpeech voices (Panayotov et al., 2015), and its conversations longer than 150 words serve long-form generation. (4) The SSSD test set provides

spontaneous conversations with annotated turn timestamps, which serve timestamp mode. Its test transcripts are paired with LibriTTS utterances as prompt recordings so that the speakers are unseen.

4.2 Model and fine-tuning configuration

We instantiate Chorus with F5-TTS (Chen et al., 2025b) and initialize from its official pretrained checkpoint[1]. As described in Section 3.2, one TAC block is inserted after each of the 22 DiT blocks, adding 92.3 M parameters to the 336 M pretrained parameters. The fine-tuning and inference hyperparameters are given in Appendix B.

4.3 Metrics

Mixture metrics are computed on the system's single-channel output, while per-channel metrics are computed independently for each channel. Differences between Chorus and ZipVoice-Dialog-Stereo are tested with a paired t-test over test conversations. The real-time factor (RTF) is the synthesis time divided by the duration of the generated conversation, measured on one NVIDIA A100 GPU and, for Chorus, also with each model M_i on its own A100, where the average of Eq. (3) is exchanged by an all-reduce after each DiT block.

Speech quality. On the mixture, we follow the ZipVoice-Dialog protocol (Zhu et al., 2026). Intelligibility is the concatenated minimum-permutation WER (cpWER) (Watanabe et al., 2020), transcribed with WhisperD (Darefsky et al., 2024) on the under-30-second conversations, naturalness is UTMOS (Saeki et al., 2022) after removing silence, and speaker similarity is cpSIM, the cosine similarity of WavLM[2] embeddings (Chen et al., 2022) to the prompt recordings after diarizing the mixture with Pyannote[3] (Bredin et al., 2020; Bredin & Laurent, 2021). These metrics have known limitations: speaker-attributed transcription and diarization degrade on overlapping speech, and UTMOS is trained on English read speech and scores conversational speech lower (Zhu et al., 2026), so all three favor systems that produce little or no overlap, and even the ground-truth cpSIM in Table 1 falls below its per-channel score. For systems with one channel per speaker, we therefore score each channel on its own. WER-Ch transcribes the channel with Whisper large-v3 (Radford et al., 2023) and scores it against the speaker's turns $\{C_k \mid s_k = i\}$, and UTMOS-Ch and SIM-Ch are UTMOS and the same similarity computed on the channel with its silence removed. cpWER and WER-Ch use different recognizers and inputs and are not comparable, and the ground-truth WER-Ch is inflated by cross-talk in the real recordings, which the generated channels do not contain.

Speaker interaction. Following dGSLM (Nguyen et al., 2023), we count utterances, pauses, gaps, and overlaps, and report counts and total durations per minute. Since these statistics do not tell whether events land at natural moments, we also apply the Talking Turns turn-taking judge (Arora et al., 2025), trained on Switchboard (Godfrey et al., 1992). At each step it predicts the next event of the mixture as one of five classes: continuation (C), silence (NA), interruption (IN), backchannel (BC), and turn change (T). We report the per-class one-vs-rest area under the curve (AUC) against the realized events, computed with the voice activity detection (VAD) of pyannote (Bredin et al., 2020), and their mean over the five classes (Mean AUC). A higher AUC means that the realized events are more predictable and thus preferred by the judge.

5 Results

5.1 All-speaker generation

Speech quality. Chorus is fine-tuned from a pretrained single-speaker TTS model on less conversational speech than every system in Table 1 that reports it. On RTF, it is ahead of the three other single-channel systems, which are far larger, and with one GPU per speaker it is faster than the ZipVoice systems. Against ZipVoice-Dialog-Stereo, the only other system with one channel per

[1] https://huggingface.co/SWivid/F5-TTS/tree/main/F5TTS_Base
[2] https://huggingface.co/microsoft/wavlm-base-plus-sv
[3] https://huggingface.co/pyannote/speaker-diarization-3.1

Table 1: Comparison on the ZipVoice-Dialog English test set. Data (h): conversational training hours (n/r: not reported). Speakers: maximum number of speakers. Response: supports response generation. Timing: turn-order (order), timestamps, or both. RTF: real-time factor on one A100 / one A100 per speaker. Output is what the metrics are computed on, the mixture (cpWER, UTMOS, cpSIM) or each speaker's channel (WER-Ch, UTMOS-Ch, SIM-Ch), as defined in Section 4.3. The two panels are not comparable with each other. † marks a significant difference between Chorus and ZipVoice-Dialog-Stereo (paired t-test, $p < 0.05$). Bold marks the best system value.

Output	System	Params	Data (h)	Speakers	Response	Timing	RTF↓	WER↓	UTMOS↑	SIM↑
mixture	Ground truth	–	–	–	–	–	–	2.85	2.57	0.856
	ZipVoice-Dialog	123M	7.6k	2	–	order	**0.17**	**3.81**	**3.53**	0.853
	MOSS-TTSD	8.3B	n/r	5	–	order	0.59	6.61	3.36	0.855
	FireRedTTS-2	2B	300k	4	–	order	1.30	11.92	3.25	**0.865**
	MoonCast	3.4B	515k	2	–	order	1.51	25.12	3.04	0.849
per speaker	Ground truth	–	–	–	–	–	–	13.48	2.75	0.884
	ZipVoice-Dialog-Stereo	123M	7.6k	2	–	order	**0.16**	**6.16**	3.37	0.856
	Chorus	428M	2.1k	any	✓	both	0.25/0.15	6.61	**3.50**†	**0.863**

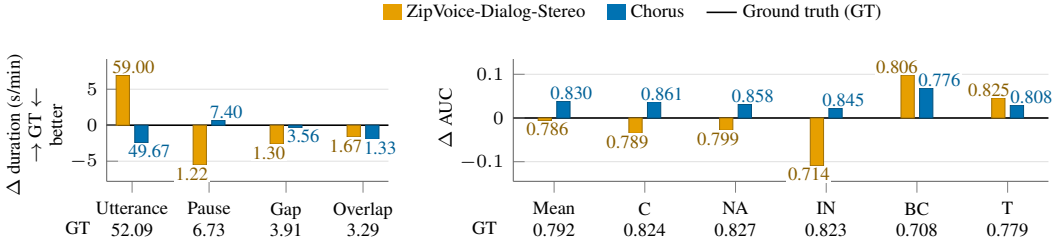

Figure 3: Speaker interaction on the ZipVoice-Dialog test set for the systems with one channel per speaker, as the difference from the ground truth (GT): the zero line is GT, its values are under the labels, and the system's value is on each bar. Left: total duration of utterances, pauses, gaps, and overlaps (s/min). Right: AUC of the turn-taking judge for continuation (C), silence (NA), interruption (IN), backchannel (BC), and turn change (T), and their mean (Mean AUC).

speaker, Chorus is significantly higher on UTMOS-Ch and higher on SIM-Ch ($p = 0.053$), while the difference on WER-Ch is not significant.

Speaker interaction. In the left panel of Figure 3, the utterance, pause, and gap durations of Chorus are close to the ground truth, whereas ZipVoice-Dialog-Stereo speaks more and pauses far less. Chorus deviates on overlap, of which it produces less than half that of the ground truth, while ZipVoice-Dialog-Stereo comes closer, despite being trained with a loss that penalizes overlapping speech (Zhu et al., 2026) (event counts in Appendix D). In the right panel, Chorus scores above the ground truth on Mean AUC and on every class of the turn-taking judge (Section 4.3), and above ZipVoice-Dialog-Stereo on Mean AUC. This holds for its backchannels and interruptions, which it produces from the turn order alone, since `<turn>` and `<other>` carry no timing (Section 3.3).

More than two speakers. No other public system generates one channel per speaker for more than two speakers (Table 1). The single-channel systems that support more speakers (Peng et al., 2025; Xie et al., 2025) cannot be evaluated with WER-Ch or SIM-Ch. Therefore, Table 2 reports Chorus on the two-, three-, and four-speaker subsets of AMI using the same weights and architecture. We report only per-channel metrics because cpWER with WhisperD is limited to two speakers (Section 4.3). From two to four speakers, WER-Ch rises only slightly, UTMOS-Ch declines only slightly, and SIM-Ch is nearly unchanged, while overlap grows with n.[4] With one GPU per speaker, the RTF stays flat from two to four speakers, since the models exchange only the average of Eq. (3). The samples are on the demo page.

Table 2: Chorus on AMI with n speakers (Section 4.1), same model for every n. Ovl.: overlap (s/min). RTF: on one A100 / one A100 per speaker.

n	WER-Ch↓	UTMOS-Ch↑	SIM-Ch↑	Ovl.	RTF↓
2	8.94	3.78	0.912	2.35	0.25 / 0.15
3	9.48	3.73	0.906	4.44	0.35 / 0.14
4	9.94	3.69	0.899	5.74	0.45 / 0.14

Long-form generation. The previous-chunk token (Section 3.5) lets the same model generate conversations well beyond the length it can generate in a single pass, where its WER-Ch is far above

[4]We do not apply the turn-taking judge here, since it is trained on two-speaker conversations (Section 4.3).

Table 3: Long-form generation on CoVoMix2 test conversations with longer than 150 words. Mean AUC is a mean per-class AUC of the turn-taking judge. Bold marks the best system value in each column.

System	WER-Ch↓	UTMOS-Ch↑	SIM-Ch↑	Mean AUC↑
ZipVoice-Dialog-Stereo	**3.52**	3.58	0.920	0.811
Chorus, single pass	19.01	3.90	0.927	**0.845**
Chorus, chunked	4.40	**3.94**	**0.933**	0.844

Table 4: Turn-order versus timestamp mode on scripted (CoVoMix2) and spontaneous (SSSD) conversations. F1 is a timing accuracy against the given timestamps at 10 ms resolution.

Test set	Mode	WER-Ch↓	UTMOS-Ch↑	SIM-Ch↑	F1↑
Scripted	Turn-order	3.98	**3.94**	0.928	0.623
	Timestamp	**3.54**	3.91	0.928	**0.934**
Spontaneous	Turn-order	**6.62**	**3.81**	**0.926**	0.669
	Timestamp	8.53	3.75	0.924	**0.947**

that of ZipVoice-Dialog-Stereo (Table 3). Chunked generation closes most of this gap and raises UTMOS-Ch, while keeping the speaker similarity and the Mean AUC of single-pass generation, both above ZipVoice-Dialog-Stereo.

5.2 Timestamp mode

Table 4 compares timestamp mode (Section 3.3) with turn-order mode on the same scripted (CoVoMix2) and spontaneous (SSSD) conversations. The CoVoMix2 scripts do not carry timestamps, so the turns are laid out consecutively with 0.4 s gaps. The timing accuracy is the F1 score (F1) over 10 ms windows between the VAD activity of each generated channel and the activity implied by the given timestamps. Timestamp mode raises F1 from 0.62 to 0.93 on scripted and from 0.67 to 0.95 on spontaneous conversations. On scripted conversations, WER-Ch improves slightly, and UTMOS-Ch and SIM-Ch are nearly unchanged, so the timing constraint costs no quality. On spontaneous conversations, WER-Ch rises and UTMOS-Ch falls, likely because these conversations contain more overlapping speech, and the text must be aligned to speech within the given spans.

5.3 Response generation

Given the ground-truth channel of one speaker from the ZipVoice-Dialog test set, Chorus generates the channel of the other (Section 3.5). Table 5 shows that the overlap of the resulting conversation more than doubles relative to all-speaker generation and approaches the ground truth, with the full event statistics in Appendix D. The overlap that all-speaker generation under-produces (Section 5.1) thus appears when one channel is a real recording, so the generated channel takes its timing from the given one. ZipVoice-Dialog-Stereo is not trained with a given channel and does not generate intelligible speech in this setting.

Table 5: Response generation on the ZipVoice-Dialog test set: one channel is given as its ground-truth recording and the other generated, averaged over both directions. Ovl.: overlap of the mixture (s/min).

System	WER-Ch↓	UTMOS-Ch↑	SIM-Ch↑	Ovl.
Ground truth	13.48	2.75	0.884	3.29
ZipVoice-Dialog-Stereo	100.23	1.31	0.537	6.34
Chorus	**9.31**	**3.18**	**0.829**	2.80

To verify the importance of the communication block, we replace the average step in Eq. (3) with each channel's own representation. Without this step, which allows the models to communicate, we observe excessive overlap, suggesting that the models cannot coordinate their speech timing with the given recording. Full results are shown in Appendix E.

6 Conclusion

We presented Chorus, which extends a pretrained single-speaker TTS model to multi-speaker conversational speech generation by assigning a weight-shared model to each speaker and connecting them through a communication block that is invariant to the number of speakers. Chorus generates one channel per speaker, supports long-form and response generation, and controls timing through either turn order or timestamps. It matches ZipVoice-Dialog-Stereo on ZipVoice-Dialog with less conversational training data, generates backchannels and interruptions without timing annotations, and generates two- to four-speaker conversations on AMI with only a small increase in per-channel word error rate. Limitations and future work are discussed in Appendix F.

ETHICS STATEMENT

Chorus extends zero-shot voice cloning to entire conversations, so a fabricated conversation between cloned voices could support impersonation or disinformation. Our intended uses are content creation with consenting speakers and training-data generation for multi-party speech processing. All corpora were collected with participant consent under research licenses.

REPRODUCIBILITY STATEMENT

Chorus is initialized from the public F5-TTS checkpoint (Section 4.2), and the TAC block, the special tokens, and the text input construction are specified in Sections 3.2 and 3.3. The fine-tuning corpora and their preprocessing are given in Section 4.1 and Appendix A, the two-stage schedule in Section 3.4 and all fine-tuning and inference hyperparameters in Appendix B, and all evaluation sets are public (Section 4.1). The metrics and the tools that compute them are listed in Section 4.3. We plan to release the code upon publication.

AI USE STATEMENT

In this work, we have not used generative AI tools for any task that requires disclosure. They were not used to generate synthetic data sets, develop the conceptual framework or method, formulate or prove claims, propose or refine hypotheses, design the experiments, implement the method, clean or reformat the data, or interpret the results. In addition, we used AI-generative tools for language editing, manuscript polishing, and code writing assistance. We reviewed all AI-assisted work. Every edited text is revised. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Siddhant Arora, Zhiyun Lu, Chung-Cheng Chiu, Ruoming Pang, and Shinji Watanabe. Talking Turns: Benchmarking audio foundation models on turn-taking dynamics. In *International Conference on Learning Representations (ICLR)*, 2025.

Hervé Bredin and Antoine Laurent. End-to-end speaker segmentation for overlap-aware resegmentation. In *Proceedings of Interspeech*, pp. 3111–3115, 2021. doi: 10.21437/Interspeech.2021-560.

Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. pyannote.audio: Neural building blocks for speaker diarization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7124–7128, 2020. doi: 10.1109/ICASSP40776.2020.9052974.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The AMI meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction (MLMI)*, pp. 28–39. Springer, 2006. doi: 10.1007/11677482_3.

Feng-Ju Chang, Martin Radfar, Athanasios Mouchtaris, and Maurizio Omologo. Multi-channel transformer transducer for speech recognition. In *Proceedings of Interspeech*, pp. 296–300, 2021. doi: 10.21437/Interspeech.2021-655.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022. doi: 10.1109/JSTSP.2022.3188113.

Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models

are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, 33:705–718, 2025a. doi: 10.1109/TASLPRO.2025.3530270.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 6255–6271, 2025b. doi: 10.18653/v1/2025.acl-long.313. URL https://aclanthology.org/2025.acl-long.313/.

Christopher Cieri, David Miller, and Kevin Walker. 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)*, 2004. URL https://aclanthology.org/L04-1500/.

Samuele Cornell, Jordan Darefsky, Zhiyao Duan, and Shinji Watanabe. Generating data with text-to-speech and large-language models for conversational speech recognition. In *Proceedings of the ISCA Workshop on Synthetic Data's Transformative Role in Foundational Speech Models (SynData4GenAI)*, pp. 6–10, 2024. doi: 10.21437/SynData4GenAI.2024-2.

Jordan Darefsky, Ge Zhu, and Zhiyao Duan. Parakeet: A natural sounding, conversational text-to-speech model. https://jordandarefsky.com/blog/2024/parakeet/, 2024. Blog post.

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

Hakan Erdogan, John R. Hershey, Shinji Watanabe, Michael I. Mandel, and Jonathan Le Roux. Improved MVDR beamforming using single-channel mask prediction networks. In *Proceedings of Interspeech*, pp. 1981–1985, 2016. doi: 10.21437/Interspeech.2016-552.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, Yanqing Liu, Sheng Zhao, and Naoyuki Kanda. E2 TTS: Embarrassingly easy fully non-autoregressive zero-shot TTS. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pp. 682–689, 2024. doi: 10.1109/SLT61566.2024.10832320.

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with permutation-free objectives. In *Proceedings of Interspeech*, pp. 4300–4304, 2019a. doi: 10.21437/Interspeech.2019-2899.

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Yawen Xue, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with self-attention. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 296–303, 2019b. doi: 10.1109/ASRU46091.2019.9003959.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 517–520, 1992. doi: 10.1109/ICASSP.1992.225858.

Jahn Heymann, Lukas Drude, and Reinhold Haeb-Umbach. Neural network based spectral mask estimation for acoustic beamforming. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 196–200, 2016. doi: 10.1109/ICASSP.2016.7471664.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Shota Horiguchi, Yuki Takashima, Paola Garcia, Shinji Watanabe, and Yohei Kawaguchi. Multi-channel end-to-end neural diarization with distributed microphones. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7332–7336, 2022. doi: 10.1109/ICASSP43922.2022.9746749.

Zeqian Ju, Dongchao Yang, Kai Shen, Yichong Leng, Zhengtao Wang, Songxiang Liu, Xinyu Zhou, Tao Qin, Xiangyang Li, Jianwei Yu, and Xu Tan. MoonCast: High-quality zero-shot podcast generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=MVlKSYR7HX.

Federico Landini, Alicia Lozano-Diez, Mireia Diez, and Lukáš Burget. From simulated mixtures to simulated conversations as training data for end-to-end neural diarization. In *Proceedings of Interspeech*, pp. 5095–5099, 2022. doi: 10.21437/Interspeech.2022-10451.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 986–995, 2017. URL https://aclanthology.org/I17-1099/.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yi Luo, Zhuo Chen, Nima Mesgarani, and Takuya Yoshioka. End-to-end microphone permutation and number invariant multi-channel speech separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6394–6398, 2020. doi: 10.1109/ICASSP40776.2020.9054177.

Wataru Nakata, Yuki Saito, Yota Ueda, and Hiroshi Saruwatari. Sidon: Fast and robust open-source multilingual speech restoration for large-scale dataset cleansing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 17617–17621, 2026. doi: 10.1109/ICASSP55912.2026.11460553.

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. Generative spoken dialogue language modeling. *Transactions of the Association for Computational Linguistics*, 11:250–266, 2023. doi: 10.1162/tacl_a_00545. URL https://aclanthology.org/2023.tacl-1.15/.

Tsubasa Ochiai, Shinji Watanabe, Takaaki Hori, and John R. Hershey. Multichannel end-to-end speech recognition. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 2632–2641, 2017a.

Tsubasa Ochiai, Shinji Watanabe, Takaaki Hori, John R. Hershey, and Xiong Xiao. Unified architecture for multichannel end-to-end speech recognition with neural beamforming. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1274–1288, 2017b. doi: 10.1109/JSTSP.2017.2764276.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, Shaohan Huang, Yan Xia, and Furu Wei. VibeVoice technical report, 2025. URL https://arxiv.org/abs/2508.19205.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 28492–28518, 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Chandan K A Reddy, Vishak Gopal, and Ross Cutler. Dnsmos p.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 886–890, 2022. doi: 10.1109/ICASSP43922.2022.9746108.

Andrew Reece, Gus Cooney, Peter Bull, Christine Chung, Bryn Dawson, Casey Fitzpatrick, Tamara Glazer, Dean Knox, Alex Liebscher, and Sebastian Marin. The CANDOR corpus: Insights from a large multimodal dataset of naturalistic conversation. *Science Advances*, 9(13):eadf3197, 2023. doi: 10.1126/sciadv.adf3197.

Rajarshi Roy, Jonathan Raiman, Sang-Gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. PersonaPlex: Voice and role control for full duplex conversational speech models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16137–16141, 2026. doi: 10.1109/ICASSP55912.2026.11463413.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Proceedings of Interspeech*, pp. 4521–4525, 2022. doi: 10.21437/Interspeech.2022-439.

Monica Sekoyan, Nithin Rao Koluguri, Nune Tadevosyan, Piotr Zelasko, Travis Bartley, Nikolay Karpov, Jagadeesh Balam, and Boris Ginsburg. Canary-1b-v2 & parakeet-tdt-0.6 b-v3: Efficient and high-performance models for multilingual asr and ast. *arXiv preprint arXiv:2509.14128*, 2025.

Zaid Sheikh, Shuichiro Shimizu, Siddhant Arora, Jiatong Shi, Samuele Cornell, Xinjian Li, and Shinji Watanabe. Scalable spontaneous speech dataset (SSSD): Crowdsourcing data collection to promote dialogue research. In *Proceedings of Interspeech*, pp. 3963–3967, 2025. doi: 10.21437/Interspeech.2025-2362.

Hubert Siuzdak. Vocos: Closing the gap between time-domain and Fourier-based neural vocoders for high-quality audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=vY9nzQmQBw.

Panagiotis Tzirakis, Anurag Kumar, and Jacob Donley. Multi-channel speech enhancement using graph neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3415–3419, 2021. doi: 10.1109/ICASSP39728.2021.9413955.

Alon Vinnikov, Amir Ivry, Aviv Hurvitz, Igor Abramovski, Sharon Koubi, Ilya Gurvich, Shai Pe'er, Xiong Xiao, Benjamin Martinez Elizalde, Naoyuki Kanda, Xiaofei Wang, Shalev Shaer, Stav Yagev, Yossi Asher, Sunit Sivasankaran, Yifan Gong, Min Tang, Huaming Wang, and Eyal Krupka. NOTSOFAR-1 challenge: New datasets, baseline, and tasks for distant meeting transcription. In *Proceedings of Interspeech*, pp. 5003–5007, 2024. doi: 10.21437/Interspeech.2024-1788.

Dongmei Wang, Zhuo Chen, and Takuya Yoshioka. Neural speech separation using spatially distributed microphones. In *Proceedings of Interspeech*, pp. 339–343, 2020. doi: 10.21437/Interspeech.2020-1089.

Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, David Snyder, Aswin Shanmugam Subramanian, Jan Trmal, Bar Ben Yair, Christoph Boeddeker, Zhaoheng Ni, Yusuke Fujita, Shota Horiguchi, Naoyuki Kanda, Takuya Yoshioka, and Neville Ryant. CHiME-6 Challenge: Tackling Multispeaker Speech Recognition for Unsegmented Recordings. In *6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, pp. 1–7, 2020. doi: 10.21437/CHiME.2020-1.

Kun Xie, Feiyu Shen, Junjie Li, Fenglong Xie, Xu Tang, and Yao Hu. FireRedTTS-2: Towards long conversational speech generation for podcast and chatbot, 2025. URL https://arxiv.org/abs/2509.02020.

Fan Yu, Shiliang Zhang, Pengcheng Guo, Yuhao Liang, Zhihao Du, Yuxiao Lin, and Lei Xie. MFCCA: Multi-frame cross-channel attention for multi-speaker ASR in multi-party meeting scenario. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pp. 144–151, 2023. doi: 10.1109/SLT54892.2023.10022715.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A corpus derived from LibriSpeech for text-to-speech. In *Proceedings of Interspeech*, pp. 1526–1530, 2019. doi: 10.21437/Interspeech.2019-2441.

Leying Zhang, Yao Qian, Long Zhou, Shujie Liu, Dongmei Wang, Xiaofei Wang, Midia Yousefi, Yanmin Qian, Jinyu Li, Lei He, Sheng Zhao, and Michael Zeng. CoVoMix: Advancing zero-shot speech generation for human-like multi-talker conversations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://openreview.net/forum?id=VNbQbv658b.

Leying Zhang, Yao Qian, Xiaofei Wang, Manthan Thakker, Dongmei Wang, Jianwei Yu, Haibin Wu, Yuxuan Hu, Jinyu Li, Yanmin Qian, and Sheng Zhao. CoVoMix2: Advancing zero-shot dialogue generation with fully non-autoregressive flow matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=0fq8vYOnxi.

Yuqian Zhang, Donghua Yu, Zhengyuan Lin, Botian Jiang, Mingshu Chen, Yaozhou Jiang, Yiwei Zhao, Yiyang Zhang, Yucheng Yuan, Hanfu Chen, Kexin Huang, Jun Zhan, Cheng Chang, Zhaoye Fei, Shimin Li, Xiaogui Yang, Qinyuan Cheng, and Xipeng Qiu. MOSS-TTSD: Text to spoken dialogue generation, 2026. URL https://arxiv.org/abs/2603.19739.

Han Zhu, Wei Kang, Liyong Guo, Zengwei Yao, Fangjun Kuang, Weiji Zhuang, Zhaoqing Li, Zhifeng Han, Dong Zhang, Xin Zhang, Xingchen Song, Lingxuan Ye, Long Lin, and Daniel Povey. ZipVoice-Dialog: Non-autoregressive spoken dialogue generation with flow matching. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38717–38729, 2026. doi: 10.18653/v1/2026.findings-acl.1928. URL https://aclanthology.org/2026.findings-acl.1928/.

APPENDIX

A CORPORA AND PREPROCESSING

Table 6 lists the corpora of both Chorus fine-tuning stages (Section 4.1). Each conversational corpus provides one recording per speaker, which we use as the per-speaker channels of Section 3.2. Chorus operates at 24 kHz, so every corpus recorded at a lower sampling rate, namely Fisher at 8 kHz telephone quality and NOTSOFAR at 16 kHz, is upsampled and restored to 24 kHz with Sidon (Nakata et al., 2026), and the remaining corpora are resampled to 24 kHz. The transcripts and turn timestamps of Fisher, CANDOR, NOTSOFAR, and LibriTTS come from the corpus annotations. SSSD is distributed without transcripts, so we transcribe each of its speaker recordings separately with Parakeet-TDT v3 (Sekoyan et al., 2025), which also returns word-level timestamps; consecutive words separated by less than 0.25 s are merged into one turn, whose span defines the speech region of that turn, so that the turns of SSSD are pseudo-labeled. All transcripts, whether from the annotations or from Parakeet, are normalized with the Whisper English text normalizer (Radford et al., 2023), which lowercases them and removes punctuation, both for fine-tuning and for scoring. Because each speaker recording of Fisher, SSSD, CANDOR, and NOTSOFAR can also capture the other participants as cross-talk (e.g., in SSSD and CANDOR through imperfect echo cancellation), we set every frame of a recording that lies outside the speech regions of its speaker, as given by the corpus annotations or, for SSSD, by the merged Parakeet timestamps, to zero, so that each channel contains only its own speaker and Chorus never learns to produce speech of another speaker on its channel. To remove low-quality recordings, we compute the OVRL score of DNSMOS P.835 (Reddy et al., 2022) on each turn of each speaker recording and average it over all turns of a session; sessions whose average falls below 3.0 are discarded, and this filter is applied to every corpus in Table 6, whose hours are reported after filtering. Finally, following CoVoMix (Zhang et al., 2024), the retained sessions are cut at turn boundaries into segments of 10 to 80 seconds, which preserves overlap within a segment.

B FINE-TUNING AND INFERENCE CONFIGURATION

We fine-tune all parameters on two NVIDIA H100 GPUs with a total batch size of 1.1 hours of audio. The learning rate is 10^{-4} for the TAC blocks and 10^{-5} for the pretrained F5-TTS weights. The first stage (Section 3.4) runs for 100K iterations and the second stage for a further 50K iterations. At inference, we use 64 sampling steps with a classifier-free guidance (Ho & Salimans, 2021) strength of

Table 6: Fine-tuning corpora after the DNSMOS filter. Stage 1 uses two-speaker conversations and single-speaker speech, and stage 2 adds meetings with more than two speakers (Section 3.4).

Corpus	Hours	Speakers per conversation	Stage
Fisher (Cieri et al., 2004)	1,323	2	1
SSSD (Sheikh et al., 2025)	676	2	1
CANDOR (Reece et al., 2023)	60	2	1
LibriTTS (Zen et al., 2019)	585	1	1
NOTSOFAR (Vinnikov et al., 2024)	7	4 to 8	2

2.0, and the Vocos vocoder (Siuzdak, 2024) converts the generated mel spectrograms to waveforms. Prompt recordings are zero-padded to a common length and cut to 4 seconds when longer.

C SINGLE-SPEAKER SETTING

To check that fine-tuning preserves the single-speaker capability, we run Chorus with $n = 1$ on the LibriTTS test-clean subset and compare it with the pretrained F5-TTS under the same inference configuration (Appendix B). With a single model, the average in Eq. (3) is the channel itself and no `<other>` token appears in the text input. Each utterance is synthesized from a prompt recording drawn from a different utterance of the same speaker, and we report WER (Whisper large-v3), UTMOS, and SIM to the prompt recording, as in Section 4.3. Table 7 shows that Chorus stays close to the ground truth in UTMOS and SIM, while its WER rises from 2.99 to 3.21. The single-speaker capability is otherwise retained.

Table 7: Single-speaker results on the LibriTTS test-clean subset ($n = 1$). Bold marks the best system value in each column.

System	WER↓	UTMOS↑	SIM↑
Ground truth	2.15	4.08	0.954
F5-TTS	**2.99**	3.88	0.946
Chorus (ours)	3.21	**4.09**	**0.947**

D EVENT STATISTICS

Table 8 reports the turn-taking event counts and durations of all-speaker generation, of which Figure 3 shows only the durations, and of response generation (Section 5.3) for each direction separately. Overlap depends on which channel is given: with channel 2 given, the overlap of the resulting conversation matches the ground truth in duration, and with channel 1 given it reaches two thirds of it. Both directions produce far more overlap than all-speaker generation, so the overlap that all-speaker generation under-produces appears once one channel is a real speaker.

Table 8: Turn-taking event statistics on the ZipVoice-Dialog test set as events per minute (#/min) and total duration in seconds per minute (s/min), for all-speaker generation (Figure 3 shows the durations only) and for response generation (Section 5.3) in each direction separately and pooled.

	Utterance		Pause		Gap		Overlap	
System	#/min	s/min	#/min	s/min	#/min	s/min	#/min	s/min
Ground truth	23.13	52.09	10.37	6.73	4.69	3.91	5.50	3.29
ZipVoice-Dialog-Stereo, all-speaker generation	13.37	59.00	1.72	1.22	4.54	1.30	4.40	1.67
Chorus, all-speaker generation	21.68	49.67	10.70	7.40	5.73	3.56	3.04	1.33
Chorus, channel 1 given	22.48	51.62	9.41	6.09	6.00	3.99	4.47	2.31
Chorus, channel 2 given	22.51	53.26	8.57	5.63	5.64	3.49	5.67	3.30
Chorus, both directions pooled	22.50	52.44	8.99	5.86	5.82	3.74	5.07	2.80

E ABLATION OF THE COMMUNICATION BLOCK

The communication block is the only part of Chorus through which a model can observe the other channels. To isolate its contribution, we compare Chorus with a variant in which the average in Eq. (3) is removed: f_{A} receives the channel's own $\hat{\mathbf{H}}_i^l$ instead of the average $\frac{1}{n}\sum_{j=1}^{n}\hat{\mathbf{H}}_j^l$, so that $\bar{\mathbf{H}}^l$ becomes the per-channel $\bar{\mathbf{H}}_i^l = f_{\mathrm{A}}(\hat{\mathbf{H}}_i^l)$ and no information passes between the models. The variant keeps f_{T}, f_{A}, and f_{C}, so it has the same number of parameters and is fine-tuned with the same

Table 9: Ablation of the communication block in response generation on the ZipVoice-Dialog test set. *w/o average* replaces the average step of Eq. (3) by the channel itself. Both models are fine-tuned for the first stage only. Ovl. is the overlap of the mixture in seconds per minute. Mean AUC is a mean per-class AUC of the turn-taking judge.

System	WER-Ch↓	UTMOS-Ch↑	SIM-Ch↑	Ovl.	Mean AUC↑
Ground truth	13.48	2.75	0.884	3.29	0.792
w/o average	11.65	3.48	0.814	5.01	0.798
w/ average (Chorus)	13.44	3.40	0.821	2.54	0.802

Table 10: Ablation of the communication block in all-speaker generation on the ZipVoice-Dialog test set, with the models of Table 9. Ovl. is overlap in seconds per minute. Mean AUC is mean per-class AUC of the turn-taking judge.

System	WER-Ch↓	UTMOS-Ch↑	SIM-Ch↑	Ovl.	Mean AUC↑
Ground truth	13.48	2.75	0.884	3.29	0.792
w/o average	10.86	3.48	0.813	3.53	0.819
w/ average (Chorus)	11.44	3.45	0.813	1.18	0.824

objective and sampling as Chorus. Both models are fine-tuned for the first stage only (Section 3.4), so their numbers are not comparable with Tables 1 and 5.

Response generation. Table 9 evaluates both models in response generation on the ZipVoice-Dialog test set (Section 5.3). The variant is better on WER-Ch, slightly better on UTMOS-Ch, and close on SIM-Ch, and the turn-taking judge separates the two only slightly. The communication block therefore does not improve the quality of an individual channel. Its effect appears in the overlap with the given channel. Chorus stays just below the ground-truth overlap, whereas the variant exceeds it by about half. Without the average step, the generated channel receives no information about the given channel and cannot know when the given speaker is talking. Its turns are placed from the text input alone, so it talks over the given speaker. The average step is what lets Chorus take the timing of its turns from the audio of another speaker.

All-speaker generation. Table 10 repeats the ablation with no channel given. The variant is again better on WER-Ch, slightly better on UTMOS-Ch, and equal on SIM-Ch, and the judge prefers both models to the ground truth and separates them only slightly. The variant's overlap is slightly above the ground truth, whereas Chorus under-produces it, as in Figure 3. This does not contradict the previous paragraph. In all-speaker generation, every channel is generated from a text input that carries the same turn order (Section 3.3), so the channels of the variant are aligned through the text alone, and their overlap follows the turn order rather than the speech of the other channel. The communication block is therefore needed when the timing of a turn must be taken from audio, as in response generation, and not when the text input already fixes the order of every turn.

F Limitations and future work

We list the limitations of Chorus in the order in which the paper introduces the corresponding components, each with the direction we plan to pursue.

Partial timing control. Timestamp mode fixes the start and end of every turn, and turn-order mode fixes none. A user therefore cannot yet time only some turns and leave the rest to Chorus. Allowing both modes within one conversation is a natural extension. A possible direction is to mix both modes within one conversation during fine-tuning.

Context in long-form generation. In long-form generation, each chunk after the first is conditioned on the last 5 seconds of the previously generated chunk, given as audio without text (Section 3.5). Nothing earlier in the conversation reaches the current chunk. Chorus therefore keeps the voices and the local flow across the chunk boundary, but it has no view of the preceding conversation as a whole. Conditioning each chunk on a global representation of all preceding chunks, instead of on their last few seconds, may make long conversations more coherent.

Offline generation. Chorus generates each chunk offline with a non-autoregressive model, so it cannot yet serve as a streaming component of a real-time system. Extending Chorus to a streaming single-speaker TTS model is left for future work.

Single instantiation. Chorus fixes neither the single-speaker TTS model nor the communication block, but we evaluate a single instantiation, F5-TTS with TAC blocks. Section 2 surveys other blocks that are invariant to the number of inputs, such as attention across the inputs, and replacing the TAC block with one of them may improve the interaction between the speakers. Two properties of F5-TTS are, however, required by the current formulation. The communication block pools the channels frame by frame, so the single-speaker TTS model must keep its channels frame-aligned. The multi-task formulation of Sections 3.3 to 3.5 selects the task through the mask, so it assumes an infilling single-speaker TTS model. Instantiating Chorus with other single-speaker TTS models and communication blocks is left for future work.

Downstream use. Section 1 motivates Chorus as a source of training data for multi-party speech processing, such as speaker diarization and conversational speech recognition. We do not test this claim. All our experiments evaluate the generated speech itself, and none uses it to train a downstream system. A possible direction is to train a diarization or conversational speech recognition system on conversations generated by Chorus and to measure the gain on real test data.