CONTINUOUS SPOKEN DIALOGUE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spoken dialogue language models (SDLMs) can support spoken interaction by modeling turn-taking behavior and semantic content. However, most existing SDLMs operate on semantic speech tokens, which may limit the modeling of fine-grained acoustic information. Furthermore, it is unclear whether their dialogue understanding benefits diverse speech generation tasks. In this work, we introduce Continuous Spoken Dialogue Language Model (CSDLM), a foundation model that **models two-channel spoken dialogue in a continuous latent space**. By modeling time-aligned channels through a unified **interleaved text-speech stream**, CSDLM supports diverse tasks, including both single-channel and two-channel speech generation, without task-specific fine-tuning. We also investigate effective voice-prompt conditioning designs with classifier-free guidance to improve speaker similarity in autoregressive flow-matching speech generation settings. We evaluate CSDLM on a suite of tasks including dialogue continuation, dialogue generation, contextual text-to-speech (TTS), and streaming TTS. Compared with other spoken dialogue models like dGSLM, CSDLM enables fine-grained control over speaking activity while improving the semantic coherence of the generated dialogue by almost 2x. Despite being a streaming model, it matches the lexical and timbre fidelity (in terms of word error rate and speaker similarity, respectively) of non-streaming baselines like VibeVoice and ZipVoice-Dialog, while generating channel-separated dialogue from a reference text. Audio samples are available on the project page.[1]

1 INTRODUCTION

Spoken conversation is a rich and dynamic form of interaction in which speakers coordinate not only what to say, but also when and how to say it. Spoken dialogue language models (SDLMs) aim to learn foundational conversational capabilities by directly modeling two-channel spoken dialogue (with one speaker per channel), including the semantic content and turn-taking behavior. Often, these models extend the next-token prediction paradigm of text LMs to spoken conversation, for example by modeling each channel separately in a two-tower architecture (Nguyen et al., 2023b).

SDLMs are often based on semantic speech tokens: dGSLM (Nguyen et al., 2023b) and SyncLLM (Veluri et al., 2024) use HuBERT-based discrete tokens (Hsu et al., 2021), while TurnGuide (Cui et al., 2026) uses the supervised semantic tokenizer of GLM-4-Voice (Zeng et al., 2024). While this allows semantic understanding, it may limit the backbone's ability to capture fine-grained acoustic information, which is important for modeling how utterances are conveyed in spoken conversation. In dGSLM, speaker identity is provided separately to the vocoder through a speaker embedding. TurnGuide uses the GLM-4-Voice speech decoder, which was fine-tuned on single-speaker speech. Additionally, most spoken dialogue language models are largely confined to dialogue continuation. Whether their learned conversational capabilities can be adapted to a broader range of speech generation tasks therefore remains under-explored.

Recent work on autoregressive speech generation has explored modeling continuous latents obtained from a variational autoencoder (VAE), allowing acoustic information to be modeled as part of the autoregressive sequence (Rouard et al., 2026; Peng et al., 2025). In particular, CALM (Rouard

[1] https://anonymous.4open.science/w/temp-3B2C/

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.

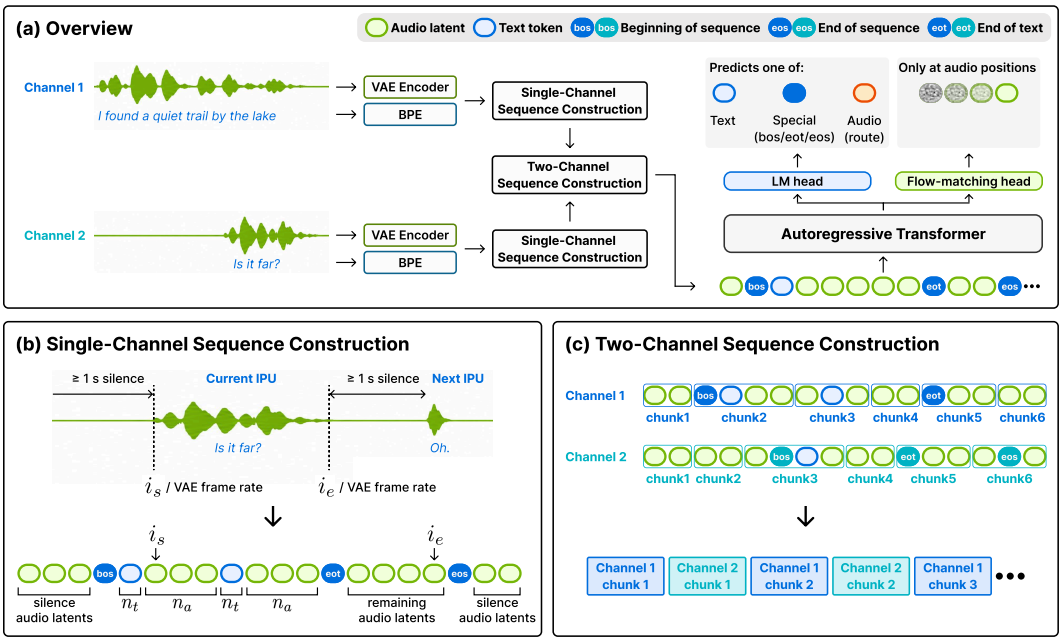


Figure 1: **Overview and sequence construction of CSDLM.** (a) An autoregressive Transformer with LM and flow-matching heads models the serialized two-channel text-audio sequence. (b) Within each inter-pausal-unit (IPU), text tokens and audio latents are interleaved at a fixed ratio, with the remaining audio latents appended after `eot`. (c) Time-aligned chunks containing a fixed number of audio latents alternate between channels. For illustration, panels (b) and (c) use $n_t = 1$ and $n_a = 3$, with an audio chunk size of 2 in panel (c).

et al., 2026) demonstrated that an audio language model operating in a continuous latent space can achieve higher acoustic quality in speech continuation than a similar model using discrete tokens based on residual vector quantization (RVQ) (Zeghidour et al., 2021), while using a substantially smaller prediction head (and consequently, a lower computational cost).

In this work, we propose Continuous Spoken Dialogue Language Model (CSDLM), a foundation model for two-channel spoken dialogue based on autoregressive flow matching in a continuous latent space. CSDLM jointly models text tokens and continuous audio latents from both speakers in a unified interleaved sequence, as illustrated in Figure 1(a). Because of this architecture, it supports in-context learning for diverse speech generation tasks without task-specific fine-tuning. Our contributions are summarized below.

1. We introduce CSDLM, a **foundation model** for two-channel spoken dialogue that jointly models text tokens and continuous audio latents. CSDLM achieves competitive performance on dialogue continuation, dialogue generation, contextual text-to-speech, and streaming text-to-speech without task-specific fine-tuning.

2. By representing the speech and text from both channels in a single token sequence, CSDLM jointly models the semantic content, turn-taking behavior, and acoustic information of two-channel spoken conversation. This allows **inference-time control** over speaking activity while producing high-coherence dialogue compared with baselines like dGSLM.

3. To the best of our knowledge, CSDLM is the only model that supports **single- and two-channel speech generation from streaming text input**. Despite this, it achieves comparable output fidelity to non-streaming dialogue generation models (like VibeVoice and ZipVoice-Dialog): it obtains the best WERs and the second-best speaker similarity in dialogue generation.

4. To investigate effective **timbre conditioning** for autoregressive flow-matching speech generation, we explore several voice-prompt conditioning schemes. We demonstrate that injecting speaker embeddings as an LM prefix as well as in the flow-matching head provides balanced results for WERs and speaker similarity.

2 METHOD

2.1 PRELIMINARIES

Flow matching Flow matching models (Lipman et al., 2023) learn a time-dependent vector field that transforms a simple distribution into the target data distribution. Let $\mathbf{x}_1 \sim p_{\text{data}}$ denote a real data sample and $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denote a Gaussian noise sample. For $t \in [0, 1]$, a linear interpolation path is defined by

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \qquad \mathbf{v}_t = \frac{d\mathbf{x}_t}{dt} = \mathbf{x}_1 - \mathbf{x}_0. \tag{1}$$

The model is trained to minimize the loss formulated as

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, t}\left[\left\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{v}_t\right\|^2\right], \tag{2}$$

where θ denotes the model parameters. At inference time, a sample is computed by using an ODE solver to transform a Gaussian noise sample according to the learned vector field.

Autoregressive flow matching MAR (Li et al., 2024) models the distribution of a continuous latent sequence $(\mathbf{x}^1, \ldots, \mathbf{x}^n)$ in an autoregressive manner. Given the latents $(\mathbf{x}^1, \ldots, \mathbf{x}^{i-1})$, its Transformer produces a hidden state $\mathbf{h}^i$. A diffusion head then models $p(\mathbf{x}^i \mid \mathbf{h}^i)$ using a denoising objective, whose gradients are backpropagated through both the diffusion head and the Transformer. VibeVoice (Peng et al., 2025) adapts this autoregressive continuous-latent modeling framework to speech generation, while CALM (Rouard et al., 2026) extends it with a consistency-model head with flow matching for regularization. These models initialize their Transformer backbones with weights from pretrained large language models (LLMs) and model continuous audio latents.

2.2 INTERLEAVED SEQUENCE

SDLMs such as dGSLM, SyncLLM, and NTPP model spoken dialogue through speech tokens without any accompanying text (Nguyen et al., 2023b; Veluri et al., 2024; Wang et al., 2025). However, it was recently shown that predicting aligned text before the corresponding speech improves semantic modeling (Défossez et al., 2024). CALM (Rouard et al., 2026) adopts this approach using a multi-stream architecture that sums time-aligned text and audio embeddings. Such an architecture may restrict the model to a fixed number of channels; to avoid this, we propose an interleaved text-audio sequence which supports two-channel dialogue modeling, while still being flexible for training with single-channel or text-only sequences. In this section, we first describe how text and audio are interleaved within each single-speaker channel, and then explain how the two speaker channels are interleaved.

Single-channel speech-text interleaving Fig. 1(b) shows an overview of the speech-text interleaving for a single channel. During training, we use the Montreal Forced Aligner (McAuliffe et al., 2026) to obtain word-level timestamps for the channel's transcript and segment the channel into inter-pausal units (IPUs) (Nguyen et al., 2023b). Each IPU is a contiguous span of speech bounded by pauses, with a silence threshold of one second used for segmentation in our setting. We tokenize the transcript within each IPU into BPE tokens and interleave them with the corresponding audio latents at a fixed ratio, following Du et al. (2024). IPU-level alignment of segments allows the model to capture pausing behavior and reduces reliance on individual word-level timestamps. Fixed interleaving within a segment allows predetermined insertion positions during text-conditioned generation without requiring the model to predict their timing.

Because continuous audio latents have a constant frame rate, the timestamp boundaries of each IPU determine its first and last audio latent indices. For a given IPU, let i_s and i_e denote the indices of its first and last audio latents in the audio-latent sequence, respectively. We insert a beginning-of-sequence (`bos`) token immediately before the latent at i_s. Within this range, we repeatedly interleave n_t text tokens followed by n_a audio latents. After all text tokens have been emitted, we insert an end-of-text (`eot`) token and append all remaining audio latents without enforcing the fixed $n_t : n_a$ ratio. An end-of-sequence (`eos`) token is then inserted immediately after the latent at i_e. Audio latents between the `eos` of one IPU and the `bos` of the next represent silence.

Time-aligned two-channel interleaving Each single-speaker channel follows the same sequence structure, with channel-specific IDs for `bos`, `eot`, and `eos`. Interleaving the two single-channel sequences while preserving their temporal alignment is nontrivial because equal-length sequences can contain different numbers of audio latents and therefore span different audio durations. The number of audio latents determines the timing of the sequence, whereas the number of text tokens is not tied to time. As illustrated in Fig. 1(c), we partition each single-channel sequence into contiguous chunks with a fixed audio chunk size, defined as the number of audio latents per chunk, regardless of the number of text tokens. Each chunk contains all tokens following the preceding chunk up to its final audio latent, including text and special tokens.

Since the two audio streams in a spoken conversation are time-aligned and use the same audio-latent representation, chunks with the same index correspond to the same temporal interval. We construct the two-channel sequence by serializing the time-aligned chunks in a fixed alternating order. This deterministic pattern allows the model to identify the active channel from the audio-latent count and chunk position, without requiring additional per-chunk channel markers or channel embeddings. Consequently, CSDLM can model dialogue semantics and turn-taking behavior, while supporting tasks such as dialogue generation and contextual text-to-speech. We set the text-audio interleaving ratio to $n_t : n_a = 5 : 6$ and the audio chunk size to one audio latent.

2.3 Model Architecture

Architecture CSDLM consists of an autoregressive Transformer backbone, a language modeling (LM) head, and a flow-matching head, as shown in Fig. 1 (a). We initialize the backbone and the LM head with the pretrained weights from Qwen3.5-2B-Base (Qwen Team, 2026). We use the VAE from VibeVoice (Peng et al., 2025), which encodes 24 kHz audio into continuous audio latents at a frame rate of 7.5 Hz. To match the dimensionality of continuous audio latents to Qwen's hidden size, we apply a linear projection to each audio latent. Given an interleaved sequence, the backbone produces a hidden state at every position. The LM head predicts BPE text tokens and special tokens at their corresponding positions. At each audio-latent position, it instead predicts a channel-specific audio route token (`route`). Conditioned on the same hidden state, the flow-matching head models the corresponding continuous audio latent and is trained only at audio-latent positions. The flow-matching head is a lightweight MLP (Li et al., 2024) with 50M parameters.

Training objective Let $\mathcal{L}_{\text{text}}$ denote the cross-entropy loss over all text tokens and special tokens except for audio route tokens. Let $\mathcal{L}_{\text{ART}}$ denote the cross-entropy loss for predicting audio route tokens at audio-latent positions, and let $\mathcal{L}_{\text{FM}}$ denote the flow-matching loss computed only at positions whose prediction targets are continuous audio latents. The overall training objective is

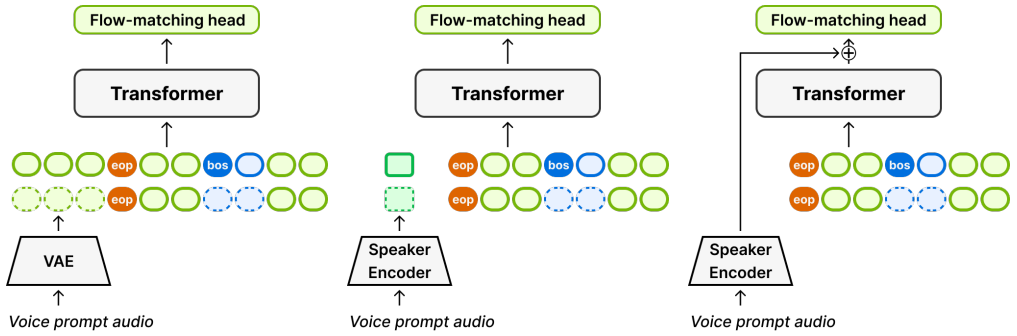
$$\mathcal{L} = \lambda_{\text{text}}\mathcal{L}_{\text{text}} + \lambda_{\text{ART}}\mathcal{L}_{\text{ART}} + \lambda_{\text{FM}}\mathcal{L}_{\text{FM}}. \tag{3}$$

Inference The same CSDLM checkpoint supports different generation tasks through different generation policies. For dialogue continuation, the LM head autoregressively predicts text and special tokens. When a `route` token is selected, the flow-matching head generates a continuous audio latent conditioned on the same hidden state, and this latent is used as the next input. Otherwise, the selected text or special token is used as the next input.

For text-conditioned generation, we force the given text tokens and generate audio latents at positions determined by the predefined interleaving structure, without requiring audio route token predictions. Once all text tokens for the current utterance have been consumed, we force `eot`. We then check the LM head for `eos` at each subsequent audio-latent generation step. For contextual and streaming text-to-speech, `eos` ends generation. For dialogue generation, it ends the current utterance, and generation continues until all turns in the script have been generated.

On each channel, the `bos` token marks the beginning of an IPU. By adjusting its emission, we can control how often the model starts speaking, including while the other speaker is speaking. We therefore investigate BOS thresholding as a way to control speaking activity during dialogue continuation. When a `route` token is sampled for a channel outside an active `bos`–`eos` span, we compute a softmax over the temperature-scaled logits of that channel's `bos` and `route`. If the resulting `bos` probability is at least τ, we replace the sampled route token with `bos`.

Implementation details and hyper-parameters are provided in Appendix A.



(a) VAE prompt (b) Speaker-embedding prompt (c) Head-level prompt

Figure 2: Three voice-prompt conditioning designs. The `eop` token denotes the end of the task prompt, and dashed outlines indicate null conditioning.

2.4 Voice Prompt Conditioning

By prepending a voice prompt to the sequence, CSDLM supports in-context learning for zero-shot timbre conditioning. However, a voice-prompt prefix alone does not provide sufficient speaker similarity. We use classifier-free guidance (CFG) (Ho & Salimans, 2021) with respect to both text and the voice prompt. Although conditional and unconditional branches can be constructed in different ways for CFG, effective voice-prompt conditioning in autoregressive flow-matching speech generation remains underexplored. For example, CALM (Rouard et al., 2026) reports a trade-off between speaker similarity and audio quality when applying CFG to an audio prefix, and uses text-only CFG in its main TTS evaluation. We therefore investigate four voice-prompt conditioning designs that differ in the representation and injection point of the voice prompt. Across all designs, text CFG is implemented by replacing the embeddings of text tokens with zero vectors in the unconditional branch. During training, text and voice-prompt conditioning are independently dropped with a probability of 0.1 each.

VAE prompt conditioning As shown in Fig. 2(a), we encode the voice prompt using the same VAE used to obtain continuous audio latents for CSDLM. We prepend a task prompt to the interleaved sequence: `Voice prompt:  Speaker 0:` $\mathbf{A}^0$ `Speaker 1:` $\mathbf{A}^1$ `eop`, where $\mathbf{A}^k$ denotes the voice-prompt audio latent sequence of speaker k and `eop` denotes the end-of-task-prompt token. For the unconditional branch, we replace the voice-prompt audio latents with a learnable null embedding repeated to the same length.

Speaker-embedding prompt conditioning As shown in Fig. 2(b), we replace the VAE latent sequence in the task prompt with a single speaker embedding extracted by the pretrained speaker encoder of Qwen3-TTS (Hu et al., 2026). The task prompt is `Voice prompt:  Speaker 0:` $\mathbf{s}^0$ `Speaker 1:` $\mathbf{s}^1$ `eop`, where $\mathbf{s}^k$ denotes the speaker embedding of speaker k. For the unconditional branch, the speaker embedding is replaced with a learnable null embedding.

Head-level prompt conditioning In head-level prompt conditioning, we provide the speaker embedding directly to the flow-matching head rather than including it in the task prompt. We apply separate linear projections to the LLM hidden state and the speaker embedding and add the projected representations to condition the flow-matching head. For the unconditional branch, only the projected LLM hidden state is used.

Dual-path speaker-embedding conditioning It combines speaker-embedding and head-level prompt conditioning. For the unconditional branch, we apply the corresponding null conditions to both paths.

Table 1: **Voice-prompt conditioning results after 50K training steps.** Each entry reports WER (%) ↓ / SIM ↑. Bold and underline indicate the best and second-best results, respectively, for each metric within each CFG setting.

Method	w/o CFG	CFG 1.5	CFG 2.5	CFG 3.5
VAE (Target Reconstruction)		4.18 / 0.63		
VAE prompt	21.56 / 0.29	<u>6.73</u> / 0.34	6.02 / 0.42	<u>6.42</u> / 0.40
Speaker-embedding prompt	25.75 / <u>0.35</u>	6.97 / <u>0.51</u>	6.55 / <u>0.56</u>	6.56 / <u>0.56</u>
Head-level prompt	**16.70** / 0.30	**5.86** / 0.43	**5.15** / 0.47	**5.24** / 0.48
Dual-path speaker-embedding	<u>17.06</u> / **0.41**	7.20 / **0.54**	<u>5.74</u> / **0.58**	7.21 / **0.57**

2.5 Training Stages

We train CSDLM in two stages, first focusing on short single-speaker utterance generation and then extending the model to two-channel dialogue generation of up to 320 seconds.

Stage 1: monologue We first train on ∼160K hours of monologue data from Emilia (He et al., 2024), HiFiTTS-2 (Langman et al., 2025), LibriTTS (Zen et al., 2019), GLOBE (Wang et al., 2024), and in-house data. We use `Generate a monologue by one speaker in one audio channel.` as the task prompt and train for 200K steps.

Stage 2: dialogue Starting from the Stage 1 checkpoint, we continue training on the same monologue corpus and approximately 22K hours of dialogue data from Fisher (Cieri et al., 2004) and in-house data. At each training step, monologue and dialogue batches are sampled at a ratio of $1:14$. We use the Stage 1 task prompt for monologue data and `Generate a dialogue between two speakers in two audio channels.` for dialogue data. We train for an additional 50K steps.

3 Experiments and Results

We first compare the voice-prompt conditioning schemes, and then evaluate CSDLM on four zero-shot speech generation tasks: dialogue continuation, dialogue generation, contextual text-to-speech (TTS), and streaming TTS. We compute word error rate (WER) by transcribing generated speech using Whisper-large-v3 (Radford et al., 2023) and comparing the transcription with the target text. For speaker similarity (SIM), we extract embeddings from the generated and reference speech using a WavLM-large-based speaker verification model (Chen et al., 2022) and compute their cosine similarity.

3.1 Voice Prompt Conditioning

We evaluate voice-prompt conditioning methods on zero-shot TTS using the LibriSpeech-PC test-clean set (Meister et al., 2023), following the evaluation-set construction of F5-TTS (Chen et al., 2025). We train each conditioning variant for 50K steps following the Stage 1 monologue training setup. We report WER and SIM, using the voice prompt as the reference for SIM. We evaluate each method without CFG and with CFG scales of 1.5, 2.5, and 3.5. The unconditional branch uses null conditioning for both text and the voice prompt. Table 1 presents the results.

Applying CFG substantially reduces WER and increases SIM across all four conditioning designs. A CFG scale of 2.5 provides the lowest WERs across all conditioning schemes. Increasing the CFG scale from 1.5 to 2.5 improves SIM across all four designs, whereas increasing it further to 3.5 generally does not result in further improvement.

VAE prompt yields the lowest SIM across all evaluated CFG settings. Unlike pretrained speaker embeddings, VAE latents are not designed to isolate speaker-related attributes, which may make it more challenging for the model to extract information relevant to speaker similarity. Head-level prompt achieves the lowest WER, possibly because conditioning only the flow-matching head allows the LLM backbone to focus more on linguistic content. By combining head-level prompting with

Table 2: **Dialogue continuation results.** τ denotes the BOS threshold. Turn-taking statistics are reported as count/min (cumulative duration as a percentage of the 90-second continuation). IPU statistics are averaged across the two channels.

	Semantics		Turn-taking statistics			
Model	GPT-Score ↑	PPL	IPU	Pause	Gap	Overlap
Ground Truth	3.82	8.34	9.3 (49.2%)	4.3 (5.9%)	5.6 (5.0%)	7.8 (9.6%)
dGSLM	1.40	24.18	14.6 (45.6%)	9.9 (10.0%)	8.5 (7.3%)	9.9 (8.8%)
CSDLM$_{\tau=0.1}$	**2.76**	8.58	7.9 (34.1%)	6.8 (15.5%)	4.6 (6.9%)	3.8 (3.6%)
CSDLM$_{\tau=0.03}$	2.74	10.59	10.0 (40.7%)	6.5 (12.8%)	5.7 (6.3%)	6.9 (6.7%)
CSDLM$_{\tau=0.01}$	2.67	8.13	12.1 (45.3%)	5.5 (9.6%)	6.6 (6.4%)	11.3 (11.4%)
CSDLM$_{\tau=0.003}$	2.56	5.18	14.2 (48.8%)	4.5 (7.5%)	7.0 (6.0%)	16.0 (17.7%)

speaker-embedding, the dual-path method achieved the highest SIM on all CFG settings, possibly because it allows both the LLM and the flow-matching head to model speaker characteristics. At a CFG scale of 2.5, it obtained the highest SIM of 0.58 and the second-lowest WER of 5.74%. Based on these results, we use dual-path speaker-embedding conditioning with a CFG scale of 2.5 for the following experiments, prioritizing speaker similarity while maintaining competitive WER.

3.2 Dialogue Continuation

We evaluate two-channel dialogue continuation on 1,000 samples from the Fisher test set defined by TurnGuide (Cui et al., 2026). Given the first 30 seconds of a dialogue as a prompt, each model generates a 90-second continuation. To the best of our knowledge, dGSLM (Nguyen et al., 2023b) is the only SDLM with publicly available inference code and pretrained weights supporting two-channel dialogue continuation. We therefore use it as our baseline.[2]

Semantic quality We evaluate semantic quality using GPT-Score and perplexity (PPL), with evaluation details provided in Appendix B.1. As shown in Table 2, across all the BOS threshold settings, CSDLM outperforms dGSLM on both metrics of semantic quality. GPT-Scores remain within a relatively narrow range of 2.56–2.76, consistently above dGSLM's score of 1.40, while the perplexity is close to that of the ground truth reference. This suggests that CSDLM can generate semantically meaningful dialogue across different levels of speaking activity. We also provide qualitative examples of dialogues generated from both models in Appendix C.

Turn-taking behavior We report the frequency and cumulative duration of IPUs, pauses, gaps, and overlaps, as detailed in Appendix B.1. From Table 2, we see that lowering the BOS threshold increases both the frequency and cumulative duration of IPUs. When τ decreases from 0.1 to 0.003, the mean IPU count per channel increases from 7.9 to 14.2 per minute, while the mean cumulative IPU duration per channel increases from 34.1% to 48.8% of the continuation duration. Pauses also become shorter and less frequent, while overlaps become more regular. Together, these changes indicate that lowering τ leads to more frequent speech and more time spent speaking overall. Effectively, BOS thresholding provides fine-grained control over speaking activity at inference time.

3.3 Dialogue Generation

We evaluate two-speaker spoken dialogue generation from text on the CoVoMix2 test set (Zhang et al., 2026), which contains 1,000 dialogue scripts. Each dialogue script is paired with two voice-prompt clips from LibriSpeech test-clean (Meister et al., 2023), one for each speaker. To match the turn-based structure of the test set, we constrain CSDLM to generate non-overlapping turns by restricting the emission of new bos tokens within an active IPU span, and force-decoding it if the model does not emit within 0.5 seconds after the previous turn's eos. We compare CSDLM with Dia (Nari Labs, 2025), MoonCast (Ju et al., 2026), VibeVoice (Peng et al., 2025), and ZipVoice-Dialog (Zhu et al., 2026). As indicated in the table, all of these are non-streaming models, and only

[2]We exclude TurnGuide (Cui et al., 2026) because its released inference code and pretrained weights only support generating one speaker's channel conditioned on the other speaker's audio throughout generation.

Table 3: **Dialogue generation results.** cpSIM uses diarization and permutation matching; SIM uses known channel-to-speaker correspondence (only available for 2ch models). Bold and underline indicate the best and second-best results, respectively.

Model	2ch	Streaming	WER (%) ↓	cpSIM ↑	SIM ↑
Dia	✗	✗	21.34	0.43	–
MoonCast	✗	✗	6.85	0.52	–
VibeVoice	✗	✗	<u>4.61</u>	**0.60**	–
ZipVoice-Dialog	✓	✗	**4.30**	0.46	<u>0.48</u>
CSDLM (Ours)	✓	✓	**4.30**	<u>0.54</u>	**0.55**

Table 4: **Contextual TTS results.** Bold and underline indicate the best and second-best results, respectively.

Model	2ch	Streaming	WER (%) ↓	SIM ↑	EmoSIM ↑
VibeVoice (w/o context)	✗	✗	<u>14.21</u>	0.27	0.67
VibeVoice	✗	✗	22.93	**0.53**	<u>0.70</u>
CSDLM (w/o context)	✓	✓	14.84	0.40	0.67
CSDLM	✓	✓	**9.09**	<u>0.47</u>	**0.72**

ZipVoice-Dialog can generate two-channel speech. We report WER and cpSIM for all models, and additionally report channel-wise SIM for two-channel outputs. Evaluation details are provided in Appendix B.2.

As shown in Table 3, CSDLM matches ZipVoice-Dialog with the lowest WER of 4.30% and achieves the second-highest cpSIM of 0.54. Among the two-channel models, CSDLM also achieves higher SIM than ZipVoice-Dialog. These results show that CSDLM can generate intelligible dialogue and follow the reference voice prompts effectively, while supporting both two-channel and streaming generation. It is worth noting that, unlike most existing dialogue generation models, which either do not support overlapping speech or support it only to a limited extent, CSDLM can generate fully overlapping speech, such as backchannels.

3.4 Contextual TTS

Contextual TTS requires generating a target speaker's utterance from text while conditioning on the preceding context of a two-speaker conversation. Since there are no existing benchmarks for contextual TTS, we constructed an evaluation set from the two-channel conversational subset of Expresso (Nguyen et al., 2023a), which contains diverse emotional expressions. We select 558 target sentences from 250 conversations. For each target, the previous two turns are provided as conversational context. We compare CSDLM with VibeVoice (Peng et al., 2025), evaluating both models with and without conversational context, keeping the target text and timbre conditioning unchanged.

We report the WER, SIM, and EmoSIM for both models. EmoSIM measures emotion similarity between the generated speech and the ground-truth target recording. We extract emotion embeddings using Empathic-Insight-Voice-Small (Schuhmann et al., 2025) and compute their cosine similarity. WER and EmoSIM are computed against the ground truth text and speech, respectively, while SIM uses the target speaker's speech in the context as the reference.

As shown in Table 4, without conversational context, CSDLM achieves WER and EmoSIM comparable to VibeVoice while yielding higher SIM. The relatively high WER and low SIM in this setting may reflect the challenges posed by Expresso's highly expressive reference speech. For VibeVoice, adding conversational context improves SIM and EmoSIM but degrades WER from 14.21% to 22.93%. This increase may reflect a difficulty in preserving text fidelity while following highly expressive conversational context. In contrast, context improves all three metrics for CSDLM, reducing WER from 14.84% to 9.09% and increasing EmoSIM from 0.67 to 0.72. Although VibeVoice achieves higher SIM with context, CSDLM shows a marginally higher EmoSIM, sug-

Table 5: **Streaming TTS results.** All TTS models are evaluated in a streaming generation setting. TTFA reports median latency. Bold and underline indicate the best and second-best results, respectively.

Model	WER (%) ↓	SIM ↑	TTFA (ms) ↓
Target	2.47	0.69	–
VAE (Target Recon.)	4.18	0.63	–
CosyVoice2	5.66	<u>0.66</u>	1600
CosyVoice3	5.03	**0.70**	1423
VoXtream2	**3.35**	<u>0.66</u>	**113**
CSDLM (Stage 1)	<u>4.41</u>	0.59	<u>145</u>
CSDLM (Stage 2)	4.52	0.57	

gesting that it can leverage conversational context to generate speech with appropriate emotional expression without sacrificing text fidelity.

3.5 Streaming TTS

Finally, we evaluate streaming TTS in the same zero-shot setting and with the same metrics as in Section 3.1. In this setting, all models receive text incrementally and begin generating speech before the complete input text is available. We compare CSDLM with CosyVoice2 (Du et al., 2024), CosyVoice3 (Du et al., 2025), and VoXtream2 (Torgashov et al., 2026). We evaluate the Stage 1 and Stage 2 checkpoints of CSDLM, using 8 flow-matching steps per audio latent. We additionally measure latency on a single NVIDIA A100 80GB GPU with batch size 1, using the same 30 examples for all models with three repetitions each. TTFA measures the time from the start of request processing until the first waveform chunk is available, and we report its median.

As shown in Table 5, both CSDLM checkpoints achieve lower WER than CosyVoice2 and CosyVoice3. From Stage 1 to Stage 2, WER increases from 4.41% to 4.52%, while SIM decreases from 0.59 to 0.57. These small changes suggest that CSDLM largely retains its streaming TTS performance while supporting both single-channel and two-channel speech generation. CSDLM achieves the second-lowest TTFA among the evaluated models at 145 ms.

It is worth investigating why CSDLM obtains a lower SIM than the baseline models. First, CosyVoice2's (Du et al., 2024) resynthesis of ground-truth speech tokens achieves a SIM of 0.68, while ours is capped at 0.63, indicating a lower ceiling. Second, CosyVoice3 (Du et al., 2025) and VoXtream2 were trained on $\sim$1M hours of audio, which is 5x our training data[3]. We believe that CSDLM can close the gap if we use better VAE latents and scale to larger training data.

4 Conclusion

We introduced Continuous Spoken Dialogue Language Model (CSDLM), an autoregressive flow-matching framework for two-channel spoken dialogue that models continuous VAE latents. CSDLM combines a deterministic interleaving of two time-aligned text-audio sequences with an autoregressive Transformer, a language modeling head, and a flow-matching head to jointly model semantic content, turn-taking behavior, and acoustic attributes of spoken conversation. We further explored voice-prompt conditioning for autoregressive flow matching and found that dual-path speaker-embedding conditioning provides effective voice conditioning. Experiments show that CSDLM supports zero-shot dialogue continuation, dialogue generation, contextual text-to-speech, and streaming text-to-speech without task-specific fine-tuning, achieving competitive results across these tasks. These results suggest that CSDLM's ability to model two-channel spoken conversation can serve as a foundation for speech generation.

[3]VoXtream2 uses a frozen Depth Transformer initialized with pretrained weights from CSM(Iribe et al., 2025), which was trained on $\sim$1M hours.

AI USE STATEMENT

We used AI tools to assist with code implementation and manuscript editing. The research ideas and experimental design were developed by the authors. The authors take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We describe the model architecture, training procedure, and experimental setups in the main text. Additional implementation and evaluation details are provided in Appendices A and B. We plan to release our code.

ETHICS STATEMENT

The in-house training data were used with permission. Given CSDLM's voice-conditioning capabilities, we encourage responsible use with appropriate speaker consent and clear disclosure of synthetic speech to reduce potential misuse.

REFERENCES

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

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, 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 (Volume 1: Long Papers)*, pp. 6255–6271, 2025.

Zhengyang Chen, Sanyuan Chen, Yu Wu, Yao Qian, Chengyi Wang, Shujie Liu, Yanmin Qian, and Michael Zeng. Large-scale self-supervised speech representation learning for automatic speaker verification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6147–6151. IEEE, 2022.

Christopher Cieri, David Miller, and Kevin Walker. The fisher corpus: a resource for the next generations of speech-to-text. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva (eds.), *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL https://aclanthology.org/L04-1500/.

Wenqian Cui, Lei Zhu, Xiao-Hui Li, Zhihan Guo, Haoli Bai, Lu Hou, and Irwin King. Turnguide: Enhancing meaningful full duplex spoken interactions via dynamic turn-level text-speech interleaving. *arXiv preprint arXiv:2508.07375*, 2026.

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

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*, 2024.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, et al. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*, 2025.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *Proc. of SLT*, 2024.

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

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*, 2026.

Brendan Iribe, Ankit Kumar, and the Sesame team. Crossing the uncanny valley of conversational voice, 2025. URL https://www.sesame.com/blog/crossing-the-uncanny-valley-of-voice.

Zeqian Ju, Dongchao Yang, Shen Kai, Yichong Leng, Zhengtao Wang, Songxiang Liu, Xinyu Zhou, Tao Qin, Xiangyang Li, Jianwei Yu, et al. Mooncast: High-quality zero-shot podcast generation. *Advances in Neural Information Processing Systems*, 38:6283–6317, 2026.

Ryan Langman, Xuesong Yang, Paarth Neekhara, Shehzeen Hussain, Edresson Casanova, Evelina Bakhturina, and Jason Li. Hifitts-2: A large-scale high bandwidth speech dataset. In *Interspeech*, 2025.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 56424–56445. Curran Associates, Inc., 2024. doi: 10.52202/079017-1797. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/66e226469f20625aaebddbe47f0ca997-Paper-Conference.pdf.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Michael McAuliffe, Kaylynn Gunter, Michael Wagner, and Morgan Sonderegger. Montreal Forced Aligner and the state of speech-to-text alignment in 2026. In *Proc. Interspeech 2026*, 2026.

Aleksandr Meister, Matvei Novikov, Nikolay Karpov, Evelina Bakhturina, Vitaly Lavrukhin, and Boris Ginsburg. Librispeech-pc: Benchmark for evaluation of punctuation and capitalization capabilities of end-to-end asr models. In *2023 IEEE automatic speech recognition and understanding workshop (ASRU)*, pp. 1–7. IEEE, 2023.

Mistral AI. Mistral-small-24b-base-2501. https://huggingface.co/mistralai/Mistral-Small-24B-Base-2501, 2025.

Nari Labs. Dia-tts, 2025. URL https://github.com/nari-labs/dia. GitHub repository.

Tu Anh Nguyen, Wei-Ning Hsu, Antony D'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, Felix Kreuk, Yossi Adi, and Emmanuel Dupoux. Expresso: A Benchmark and Analysis of Discrete Expressive Speech Resynthesis. In *Interspeech 2023*, pp. 4823–4827, 2023a. doi: 10.21437/Interspeech.2023-1905.

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

OpenAI. Gpt-5.5 system card. https://deploymentsafety.openai.com/gpt-5-5/references, 2026.

Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, et al. Vibevoice technical report. *arXiv preprint arXiv:2508.19205*, 2025.

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

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

Simon Rouard, Manu Orsini, Axel Roebel, Neil Zeghidour, and Alexandre Défossez. Continuous audio language models. In *ICLR*, 2026. URL https://openreview.net/forum?id=MFrJ3NzA5H.

Christoph Schuhmann, Robert Kaczmarczyk, Gollam Rabby, Felix Friedrich, Maurice Kraus, Kourosh Nadi, Huu Nguyen, Kristian Kersting, and Sören Auer. Emonet-voice: A fine-grained, expert-verified benchmark for speech emotion detection. *arXiv preprint arXiv:2506.09827*, 2025.

Nikita Torgashov, Gustav Eje Henter, and Gabriel Skantze. Voxtream2: Full-stream tts with dynamic speaking rate control. *arXiv preprint arXiv:2603.13518*, 2026.

Bandhav Veluri, Benjamin N Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. Beyond turn-based interfaces: Synchronous llms as full-duplex dialogue agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21390–21402, 2024.

Qichao Wang, Ziqiao Meng, Wenqian Cui, Yifei Zhang, Pengcheng Wu, Bingzhe Wu, Irwin King, Liang Chen, and Peilin Zhao. NTPP: Generative speech language modeling for dual-channel spoken dialogue via next-token-pair prediction. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5Gke1dfRVA.

Wenbin Wang, Yang Song, and Sanjay Jha. Globe: A high-quality english corpus with global accents for zero-shot speaker adaptive text-to-speech, 2024.

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

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 *Interspeech 2019*, pp. 1526–1530, 2019. doi: 10.21437/Interspeech.2019-2441.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*, 2024.

Leying Zhang, Yao Qian, Xiaofei Wang, Manthan Thakker, Dongmei Wang, Jianwei Yu, Haibin Wu, Yuxuan Hu, Jinyu Li, Yanmin Qian, et al. Covomix2: Advancing zero-shot dialogue generation with fully non-autoregressive flow matching. *Advances in Neural Information Processing Systems*, 38:128915–128938, 2026.

Han Zhu, Wei Kang, Liyong Guo, Zengwei Yao, Fangjun Kuang, Weiji Zhuang, Zhaoqing Li, Zhifeng Han, Dong Zhang, Xin Zhang, et al. Zipvoice-dialog: Non-autoregressive spoken dialogue generation with flow matching. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38717–38729, 2026.

A Implementation Details

A.1 Training

For single-speaker monologue sequences, we set $(\lambda_{\text{text}}, \lambda_{\text{ART}}, \lambda_{\text{FM}}) = (1, 1, 4)$. In two-channel dialogue sequences, periods in which one speaker listens while the other speaks produce a sequence of audio latents representing silence without accompanying text. If λ_{ART} is set to 1, the model becomes biased toward predicting silence. To mitigate this issue, we set λ_{ART} to 0.3 while retaining $\lambda_{\text{text}} = 1$ and $\lambda_{\text{FM}} = 4$. Following CALM (Rouard et al., 2026), we use a head batch multiplier of 8 to train the flow-matching head efficiently.

We use AdamW with a cosine annealing learning-rate schedule. In Stage 1, the learning rates for the LLM and flow-matching head are 5×10^{-5} and 2×10^{-4}, respectively. In Stage 2, we use learning rates of 2.5×10^{-5} and 1×10^{-4}, respectively. We train CSDLM on 48 NVIDIA A100 GPUs with per-GPU batch duration set to 500 seconds for monologue batches and 320 seconds per channel for dialogue batches.

A.2 Inference

For dialogue continuation, we sample discrete tokens from the LM head using top-k sampling with $k = 25$ and a logit temperature of 0.7. For text-conditioned generation, the given text tokens are forced rather than sampled. For boundary decisions, we compute a two-way softmax over the raw logits of the corresponding special token and the audio route token, with a temperature of 0.7. We use an `eos` threshold of 0.01 for text-conditioned generation and a `bos` threshold of 0.1 for dialogue generation.

Unless otherwise specified, we generate each continuous audio latent using 25 Euler steps and a CFG scale of 2.5. For streaming text-to-speech, we use 8 Euler steps per audio latent. The audio noise temperature (Rouard et al., 2026) is set to 0.7.

B Evaluation Details

B.1 Dialogue Continuation

Semantic quality We transcribe each generated audio channel using stable-ts[4] with Whisper-large-v3 (Radford et al., 2023). We serialize the channel transcripts by timestamp and prefix each segment with its speaker tag, `A:` or `B:`. For GPT-Score, we use GPT-5.5 (OpenAI, 2026) to assess the meaningfulness of the dialogue on a scale from 0 to 5 in increments of 0.5, using a prompt adapted from TurnGuide (Cui et al., 2026). We compute PPL using Mistral-Small-24B-Base-2501 (Mistral AI, 2025), providing and scoring only the continuation transcript after removing timestamps.

Turn-taking behavior We follow dGSLM (Nguyen et al., 2023b) and compute the counts and cumulative durations of inter-pausal units (IPUs), pauses, gaps, and overlaps from the generated dialogue using pyannote (Bredin, 2023). For each sample, IPU statistics are averaged across the two channels. We normalize counts per minute and express cumulative durations as percentages of the 90-second continuation, then report the mean of each statistic across the test set.

B.2 Dialogue Generation

We transcribe the generated dialogue and compute WER against the full reference script. For two-channel outputs, we sum the channels into a mono stream before transcription.

For speaker similarity, we compute cpSIM for all models following ZipVoice-Dialog (Zhu et al., 2026). We use speaker diarization to identify individual speaker segments. To apply the same cpSIM protocol to all models, we sum two-channel outputs into a mono stream before diarization. For each of the two possible matchings between voice prompts and generated speakers, we compute

[4] `https://github.com/jianfch/stable-ts`

the average cosine similarity across the two speakers and use the higher of the two scores. For two-channel outputs (ZipVoice-Dialog and CSDLM), the channel-to-speaker correspondence is known. We additionally report SIM for each channel separately after removing non-speech regions.

C Qualitative Examples of Dialogue Continuation

Prompt

A: if they thought that bringing it up like honored their memory or whatnot and if you forgot then that would be like the worst thing you could do then obviously you should make mention of it
B: Yeah.
A: enough that they would think that they would be remembered but if they wanted closure to it and they just wanted no one to talk about it anymore because that'd be the best thing then
B: Yeah.
A: that would be where I would agree with.
B: All right. What do you think about it being like a national holiday?

dGSLM	**CSDLM (Ours)**
A: what was kind of baseball, what kind of do. **B:** Maturity, and it's never so personal. **A:** I think so. **B:** Yeah, how are you doing? A baseball being on and everything. **A:** Yeah. **B:** A little bit of a fan about it. Well, I've had some kind of like a kind of shipper boards, a boss where you live, deal. **A:** All right. **B:** A boss to lodgeable, team center, common areas, like forests, maybe over a sea or minimum wage or whatever. And like I always say about proportional systems, I would say I can't see the beginning. Yeah, I mean, most of the worst, for one reason, my best friend is recording for one of the middle land of systems. You know, I respect that. I don't think that the people absolutely recorded more. But we have to go money. Now, most of the year, weapons, uneducated, defaulter, and different from for us, we might sick portion of Canada. And most of the holiday, I want to go by just out the phone because it was Percy here. She found me heard a lot in the internet. I can buy baseball. But he was on Canada, so it was pretty tough.	**A:** I don't know. I just think it would be better to have it kind of like a National Remembrance Day. You know, like a week after or two weeks after. Just to kind of have everybody remember everybody that's been killed. And not just talk about it, but to have a ceremony and all that. **B:** Well, I don't even think about it to be a national holiday. I think it's a good idea to bring it up, you know, to bring it up, you know, to bring it up and have people talk about it and, you know, let people kind of process what happened and not have it be like a super solemn occasion or anything. **A:** yeah **B:** But yeah, I agree with that. **A:** it's just it's hard to come like with a good **B:** Right. **A:** reason to remember everybody who died in a war and not just focus on one person and say i think they're a bad person or something and just to have people talk about it. I mean, not to glorify it, but just to have people talking about it and remembering the sacrifices they made and whatnot.

Table 6: Dialogue continuation example. Both models continue the same 30-second audio prompt. We show the prompt transcript and the beginning of each generated continuation.

Prompt

B: How'd you get onto this?
A: Craigslist. I'm learning a little on a cash these days,
B: That's where I found it, too.
A: a little bit of extra money. Yeah. Yeah,
B: That's a pretty good list. I'm looking for work.
A: me too.
B: I spent quite a bit of time there.
A: I've been out of work for, what's it?
B: This was kind of interesting.
A: Yeah, I think so. I don't know. I've never done one. Is it like $105?

dGSLM	**CSDLM (Ours)**
B: so **A:** I don't know about it now. And so I don't know how you get around this day too much. **B:** well that's true **A:** Yeah. **B:** it's always a cut of mountain it could be some type of thing but **A:** I mean, it's kind of great or something. **B:** yeah **A:** And, like, if you, like, just become, you know, cold-ass someone, you're just, like, three outside of people in the same row. **B:** yeah exactly **A:** But, you know, it's really great.	**B:** Yeah, I got $25. **A:** Oh, okay. **B:** Yeah. **A:** Yeah, it's pretty under $50. **B:** It's a pretty good deal, so it's to do that. **A:** Try to get a little bit of money. **B:** Yeah, I don't know. I got a friend there in my office. **A:** Wow. **B:** He's doing it, but they're paying $25 an hour, but he's only been building for three weeks. So I think it's a better deal than $20 an hour in February. It's only two days' notice. **A:** Yeah, I know.

Table 7: Dialogue continuation example. Both models continue the same 30-second audio prompt. We show the end of the prompt transcript and the beginning of each generated continuation.