DIAGNOSING AND MITIGATING LONG-SESSION DEGRADATION IN FULL-DUPLEX SPEECH MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end full-duplex spoken dialogue models enable natural, real-time spoken interactions with interruptions, backchannels, and overlapping speech, yet their reliability over extended sessions remains poorly understood. Existing evaluations have largely focused on short-horizon interactions, and even studies involving longer conversations have not systematically characterized how model behavior evolves over the course of a session. In this work, we address this gap by first characterizing how specific dialogue capabilities change over extended interactions and then developing methods to mitigate long-session degradation while improving the efficiency and generalization of extended-session inference. To characterize long-session behavior, we introduce a closed-loop evaluation framework for long-session stability and apply it to four publicly available full-duplex speech models in ten-minute conversations. Across models, we observe widespread degradation over time in responsiveness, content quality, and speech quality, highlighting the difficulty of maintaining stable conversational behavior over extended interactions. To mitigate this degradation, we propose a self-distillation-based training framework for lightweight LoRA adaptation that improves long-session stability with a small amount of additional training. We further introduce a two-resolution context (TRC) architecture that preserves recent interactions in full detail while compressing older interaction history into key-value representations, enabling models to learn how to effectively leverage both short-term and long-term context to maintain content and speech quality over longer sessions while reducing memory and computational costs. We apply our methods to three architectures with distinct formulations of full-duplex conversational context and evaluate long-session stability, information retention, and computational efficiency. Lightweight adaptation reduces performance degradation over ten-minute sessions across all three models. With TRC, the models also maintain these gains beyond the training horizon while reducing KV-cache growth by more than 90%.

1 INTRODUCTION

Natural spoken conversation involves dynamic coordination between speakers. A listener may respond to an utterance before it ends, take the floor during a pause, or interrupt when necessary, rather than waiting for strictly separated turns. Spoken dialogue models, however, typically follow a turn-based interaction scheme in which listening and speaking occur in separate turns. Recent full-duplex spoken dialogue models move beyond this constraint by processing incoming speech while simultaneously generating responses, enabling interaction behaviors such as rapid turn-taking, backchannels, and interruptions (Nguyen et al., 2023; Défossez et al., 2024). With recent advances in full-duplex speech modeling, such interactions have become increasingly mature.

A prominent line of recent work on full-duplex speech modeling takes an end-to-end approach (Défossez et al., 2024; Roy et al., 2026; Cui et al., 2026; Kim et al., 2026), in which response content and speaking decisions are integrated within the generative process. In these models, a long conversation does not simply add more linguistic content to the context. The accumulated context also reflects the interaction dynamics that unfold throughout the conversation, as well as model outputs used to coordinate speaking behavior. The model must therefore manage an increasing contextual load as the session unfolds. Moreover, because response generation and speaking control are integrated within the same generative process, limitations in handling long-range context may affect

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.

multiple aspects of generation, including response quality, responsiveness, and the intelligibility of generated speech.

Alongside advances in full-duplex speech modeling, evaluation protocols have evolved to assess these interaction capabilities. Recent benchmarks cover behaviors such as turn-taking, interruption handling, and response timing (Lin et al., 2025; Zhang et al., 2026). Some evaluations further incorporate simulated users and multi-round interactions to better reflect realistic conversational settings (Lin et al., 2026a; Ray et al., 2026). Yet, to our knowledge, the stability of end-to-end full-duplex models over extended conversations has received limited attention, despite the potential for the rapidly growing conversational context to jointly affect response content, response timing, and speech quality.

We therefore study how end-to-end full-duplex models behave over extended interactions using a long-duration evaluation framework. A simulated user dynamically responds to the model's generated utterances, while the model maintains its streaming state throughout the entire session. We evaluate model behavior in terms of *responsiveness*, *content quality*, and *speech quality*, and track how each evolves as the conversation progresses. Across four representative open-source end-to-end full-duplex models evaluated over ten-minute sessions, we find that every model exhibits degradation in at least one dimension, with distinct degradation patterns across models.

Improving long-session stability requires both a training method for sustained interaction and a context representation that supports learning over long dialogues. We introduce **privileged-context self-distillation**, where the base model serves as a teacher with access to summaries of preceding dialogue, while the student processes the same conversation continuously and learns to reproduce the teacher's response content and speaking decisions. We further introduce a **two-resolution context (TRC) architecture** by adapting UniGist, a streaming context compression method (Deng et al., 2025), to full-duplex models. TRC retains recent interactions in a local window and compresses earlier exchanges into learned gist tokens represented in the KV cache, enabling the model to handle short- and long-term conversational context both effectively and efficiently.

We make the following contributions.

- We introduce a long-duration evaluation framework for full-duplex speech LLMs, where models continuously interact with a simulated user while maintaining their streaming state throughout ten-minute conversations.
- We evaluate four representative open-source end-to-end full-duplex spoken dialogue models in terms of responsiveness, content quality, and speech quality, and show that every model exhibits degradation in at least one dimension as the conversation progresses.
- We introduce privileged-context self-distillation for efficient long-session adaptation without an external full-duplex dialogue corpus, and a two-resolution context (TRC) architecture that combines a fine-grained local window with compressed memory of earlier dialogue.

2 Related Work

2.1 Full-Duplex Speech Modeling

Spoken dialogue systems are commonly characterized by how they coordinate listening and speaking. *Half-duplex* systems alternate between the two across explicit turns, typically by cascading streaming ASR, a language model, and TTS behind a voice activity detector. *Full-duplex* systems instead listen and speak concurrently, admitting interaction behaviors such as backchannels, interruptions, and overlapping speech (Nguyen et al., 2023; Défossez et al., 2024).

Full-duplex systems differ mainly in where the decision to speak or listen is made. One line keeps the decision outside the dialogue model, (Chen et al., 2025a; Liao et al., 2025). A second line retains a separate decision mechanism but conditions it on the model's internal representations, predicting from LLM hidden states whether to continue listening or begin speaking (Wang et al., 2025; Chen et al., 2025b). A third folds this decision into the generative process itself, modeling user and system speech as concurrent token streams such that turn-taking emerges as a property of the model rather than being delegated to a separate controller (Nguyen et al., 2023; Défossez et al., 2024; Zhang et al.,

2025b). Recent systems in this last line have extended the formulation toward controllable speaker identity through explicit role and voice conditioning (Roy et al., 2026), multimodal full-duplex interaction through aligned perceptual streams (Cui et al., 2026), and more natural conversational dynamics, particularly turn-taking and interruption handling (Kim et al., 2026). In this work, we focus on this third line of research, in which turn-taking decisions are integrated directly into the generative process.

2.2 Evaluation of Full-Duplex Spoken Dialogue Models

Existing evaluations of full-duplex speech models assess core interaction capabilities such as turn-taking, interruption handling, overlap management, and response timing (Arora et al., 2025; Peng et al., 2025; Lin et al., 2025; 2026b). These benchmarks provide controlled settings for testing whether a model can coordinate listening and speaking under different conversational conditions, but largely evaluate such behaviors at the level of individual exchanges or relatively short interactions.

More recent work has extended full-duplex evaluation to multi-round and adaptive interactions. MTR-DuplexBench evaluates model behavior across multiple conversational rounds (Zhang et al., 2026), while Full-Duplex-Bench v2 introduces adaptive conversations in which subsequent user utterances depend on the model's preceding responses (Lin et al., 2026a). Several recent benchmarks have also considered longer speech-based interactions (Ray et al., 2026; Bogavelli et al., 2026; Bhosale et al., 2026); however, these works still do not examine how model behavior changes progressively throughout a session. In contrast, our evaluation investigates how model performance evolves over the course of a long conversation. We track responsiveness, content coherence, and speech intelligibility over time, characterizing degradation that emerges as the session progresses.

2.3 Streaming Long-Context Modeling

Transformer-based models accumulate a key-value (KV) cache as the context length grows, increasing memory and computational overhead during inference and motivating methods that reduce the amount of context retained. StreamingLLM (Xiao et al., 2024) preserves attention sinks together with a sliding window of recent tokens, while learned compression methods retain earlier context in compact representations (Chevalier et al., 2023; Zhang et al., 2025a). More recent approaches further enable online context compression. CCM (Kim et al., 2024) compresses accumulated KV states into compact memory, while UniGist (Deng et al., 2025) uses learned gist tokens for sequence-level compression. Their focus on text-based language models leaves open how incremental compression can be applied to native full-duplex streams that combine speech, text, and interaction-control signals.

3 Diagnosing Long-Session Degradation

3.1 Protocol and Measures

Simulation setup. To examine how full-duplex models behave over a sustained interaction, we conduct 90 English conversations for each model, covering three personas and ten everyday topics with three episodes for each persona–topic pair, each targeting 600 s. We evaluate four full-duplex models: Moshi (Défossez et al., 2024), PersonaPlex (Roy et al., 2026), MiniCPM-o 4.5 (Cui et al., 2026), and Raon-SpeechChat (Kim et al., 2026). Each conversation follows a closed-loop simulation in which the content of the user's subsequent responses depends on the model's generated speech, allowing the interaction to evolve based on the model's own behavior rather than following a predefined dialogue script. The user simulator consists of an LLM, text-to-speech (TTS), automatic speech recognition (ASR), and voice activity detection (VAD) modules, as illustrated in Figure 1. Persona and topic instructions, component choices, prompts, and decoding settings are provided in Appendix A.1.

Interaction policy. To ensure a consistent interaction condition across models, the simulated user follows the same conversational policy in all sessions. The user waits for the model to finish before responding, yields when interrupted by the model, and does not artificially generate backchannels or interruptions. We categorize model utterances based on their timing relative to user speech as

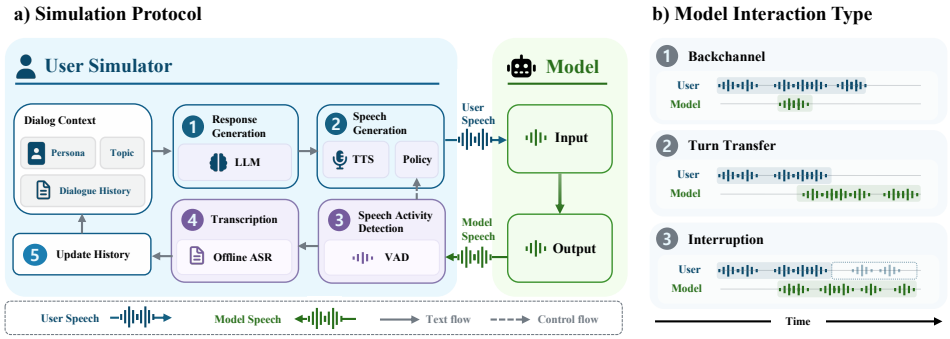

Figure 1: **Closed-loop simulation protocol and interaction types.** (a) Overview of the simulation loop, in which model responses are incorporated into the dialogue context to condition subsequent user turns. (b) Model interaction types characterized by the temporal relationship between user and model speech.

backchannels, interruptions, or turn-taking responses. These categories characterize response types rather than serving as evaluation criteria. We use them operationally to determine the simulation flow and to exclude backchannels and interruptions from FTO measurement. Backchannels leave the ongoing user turn unaffected, interruptions terminate it, and turn-taking responses follow the default flow. Turn-taking responses also include cases in which floor transfer occurs with temporal overlap between the user and model (Figure 1b).

To ensure that both simulation rules and evaluation measurements reflect the model's response decisions rather than incidental variation in wall-clock time, we define interaction timing in logical time based on processed audio. Response timeouts, retry behavior, and overlap classification are determined on this time axis. If no model response begins within the predefined timeout, the simulator rephrases the user's intent and retries; repeated response timeouts terminate the session. Timing thresholds, retry rules, and overlap classification criteria are detailed in Appendix A.1.

Evaluation metrics. To characterize how model behavior changes throughout the session, we evaluate three aspects of full-duplex interaction: responsiveness, content quality, and speech quality. We divide each session into one-minute intervals of logical time and measure six metrics. For responsiveness, we first measure **floor transfer offset (FTO)**, the temporal gap or overlap between the end of the user's speech and the beginning of the model's response during normal turn-taking, excluding backchannels and interruptions. **Response timeout (RT)** rate measures how often the model fails to begin responding within the predefined timeout, while **terminated sessions (TS)** rate measures the fraction of sessions that become unresponsive and are forcibly terminated after three consecutive response timeouts. For conversational content, we use **conversation quality (CQ)**, a 1–5 LLM-based score that evaluates whether the model's responses are contextually appropriate and coherent. For speech quality, we report **word error rate (WER)** between the model's native text output and an ASR transcription of its generated speech, together with **speaker similarity (Spk sim)**, which measures how consistently the generated voice matches its first-minute reference using speaker embeddings. Detailed metric definitions, formulas, and scoring procedures are provided in Appendices A.3.1–A.3.4. For each one-minute interval, we aggregate all observations available across sessions using the same criteria for every model. If a session terminates early, observations collected before termination are retained, while later intervals from that session are excluded. To reduce the influence of isolated outliers, we report the metric only for intervals containing more than a predefined number of measurements. Detailed aggregation rules and reporting thresholds are provided in Appendix A.4.

3.2 Long-Session Degradation Patterns

Figure 2 shows that all four models degrade as the session progresses, but in different aspects and at different stages of the conversation. Moshi shows gradual declines in conversation quality and speaker similarity, followed by a sharp loss of responsiveness late in the session. Its response timeout rate increases from 12.7% at 7–8 min to 70.2% at 8–9 min, while the terminated session rate reaches 96.7% by 10 min. PersonaPlex and Raon-SpeechChat show a different pattern, with

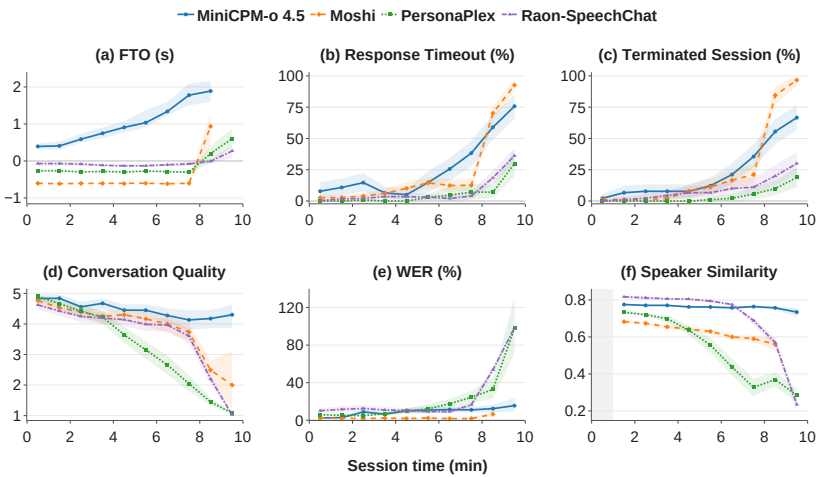

Figure 2: **Behavioral degradation over long sessions.** Each point summarizes the preceding one-minute interval; cumulative response collapse is measured through the interval endpoint. Shaded regions denote 95% confidence intervals from 2,000 episode-cluster bootstrap replicates. The gray first-minute band in (f) is used only for speaker enrollment. Interval estimates and sample sizes are reported in Appendix A.6.

more pronounced degradation in conversational content and speech quality. For PersonaPlex, these changes emerge progressively throughout the conversation, whereas for Raon-SpeechChat they are concentrated in the final two to three minutes. From the first to the final minute, mean conversation quality decreases from 4.90 to 1.08 for PersonaPlex and from 4.62 to 1.03 for Raon-SpeechChat, with final-minute WER reaching 97.9% and 98.7%, respectively.

MiniCPM-o 4.5 exhibits a different pattern, with degradation appearing primarily in responsiveness. Median FTO increases from 0.39 s at 0–1 min to 1.89 s at 8–9 min, indicating progressively slower turn-taking. Its response timeout rate reaches 75.8% in the final minute, and the cumulative response collapse rate reaches 66.7% by 10 min. Among sessions with valid observations, however, mean conversation quality remains above 4 throughout the conversation and speaker similarity remains relatively stable, while WER increases from 2.4% to 15.5%. Overall, all four models exhibit clear degradation over extended interactions, but the timing and form of degradation differ across models.

4 Methods for Long-Session Stability

To address the long-session degradation observed in Section 3.2, we use privileged-context self-distillation to transfer the base model's reliable behavior in short sessions to sustained interaction. We also introduce a two-resolution context (TRC) architecture that manages growing dialogue history efficiently and enables the model to draw on earlier context as sessions lengthen.

4.1 Privileged-Context Self-Distillation

Teacher trajectory construction. Privileged-context self-distillation builds on context distillation (Snell et al., 2022). To support efficient adaptation without an external full-duplex dialogue corpus, we construct teacher trajectories from each frozen base model's own rollouts. We filter short dialogue segments for quality and concatenate them into long trajectories. At aligned prediction positions, the teacher conditions on a privileged summary of preceding segments and the raw context of the current segment, while the student processes the continuous trajectory under teacher forcing (Figure 3b). For each backbone, we construct 100 scenarios by pairing five personas with 20 topics, both distinct from those used in the diagnostic simulation of Section 3.1. Four trajectories per scenario yield 400 trajectories. Scenario construction, trajectory generation, quality filtering, and prompts are detailed in Appendices B.5 and B.7.

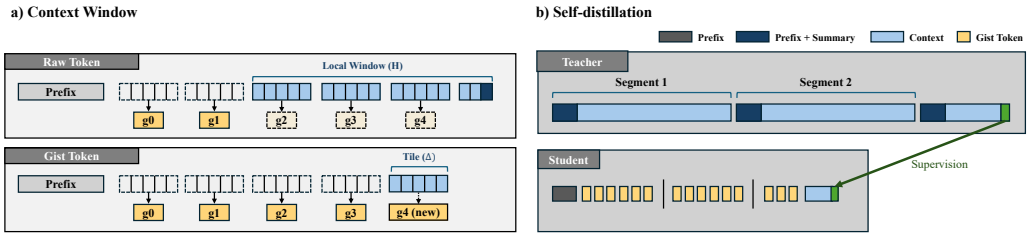

Figure 3: **TRC and privileged-context self-distillation.** (a) Attention visibility for raw-token (top) and gist-token (bottom) queries, shown with four units per tile and four tiles in the local window. Solid outlines mark visible context; dashed outlines mark masked context. Raw queries attend to the prefix, gists outside the window, and local raw tokens; gist queries attend to the prefix, all preceding gists, and their own tile. (b) The teacher restarts with a dialogue summary at segment boundaries, while the illustrated TRC student follows the concatenated trajectory continuously. Teacher predictions supervise the student at aligned positions.

Training objective. Full-duplex interaction requires the student to preserve not only what the model says, but also when and how it speaks. We therefore supervise response content and speaking behavior separately and match the hidden representations supplied to the audio decoder. We divide eligible prediction positions into *content* and *control* groups based on the token selected by the teacher and the corresponding speaking state. Content positions are ordinary text predictions made while the model continues speaking; control positions include speech transitions and predictions of native interaction-control tokens. For each group, we minimize the teacher-to-student KL divergence over the full native vocabulary, denoted by $\mathcal{L}_{\mathrm{KL,cnt}}$ and $\mathcal{L}_{\mathrm{KL,ctl}}$, respectively. Separate KL terms prevent differences in group frequency from determining their relative weights, allowing response content and speaking behavior to be weighted independently. We additionally match the hidden representations supplied to the frozen audio decoder using normalized squared error, denoted by $\mathcal{L}_{\mathrm{hid}}$, directly supervising the decoder conditioning that is not fully constrained by the KL objectives alone.

The resulting training objective is

$$\mathcal{L} = \lambda_{\mathrm{cnt}}\mathcal{L}_{\mathrm{KL,cnt}} + \lambda_{\mathrm{ctl}}\mathcal{L}_{\mathrm{KL,ctl}} + \lambda_{\mathrm{hid}}\mathcal{L}_{\mathrm{hid}}. \tag{1}$$

We set λ_{cnt}, λ_{ctl}, and λ_{hid} to 1 in all experiments. Model-specific control tokens, assignment rules, exact loss definitions, supervision masks, and optimization settings are provided in Appendix B.6.

4.2 Two-Resolution Context (TRC) Architecture

We introduce TRC for full-duplex models by adapting gist-based streaming context compression (Kim et al., 2024; Deng et al., 2025). Streaming compression suits dialogue histories that grow incrementally, and the two resolutions match the demands of full-duplex interaction: fine detail for recent speech and speaking decisions, compact memory for earlier context. We divide each session into audio tiles of duration Δ and compute one learned gist token at the end of each tile. Recent tiles remain available in their original, uncompressed form within a local window, while gist tokens provide a compact representation of earlier dialogue. During privileged-context self-distillation, gist memory plays a role analogous to the teacher's privileged summary. The TRC architecture encourages the student to develop a general ability to use recent fine-grained context and compressed long-range information effectively.

More formally, at prediction position i, the model attends to

$$\mathcal{S}_i = \mathcal{P} \cup \mathcal{G}_i \cup \mathcal{W}_i, \tag{2}$$

where $\mathcal{P}$ is a fixed prefix that preserves the prompt while accounting for model-specific attention sinks (Appendix B.4), $\mathcal{G}_i$ contains gist tokens for tiles that have left the local window, and $\mathcal{W}_i$ contains recent raw context tokens. Here H is the local-window duration, chosen as a multiple of Δ; the window contains H/Δ tiles, including the current tile. Once a tile leaves the local window, its raw tokens are discarded, and the corresponding gist token serves as its compressed representation. Thus, recent interaction remains at native resolution, and older history remains available in compressed form (Figure 3a, top).

Each gist summarizes its tile while carrying forward information from earlier dialogue. To do so, the gist token attends to the fixed prefix, all preceding gist tokens, and the raw tokens within its own

tile (Figure 3a, bottom). Because each gist accesses all preceding gists, information can accumulate across tiles even when raw tokens from earlier tiles remain in the local window. Each gist token is initialized from a shared learnable embedding and processed through the backbone layers. We update only this embedding and LoRA adapters applied to the Q/K/V/O projections of the backbone, while keeping the audio modules frozen.

At inference time, gist memory grows by one token per tile, while recent raw tokens remain in the local window. Training follows the same visibility and positional rules, so the student observes the same context structure used during streaming inference. We use an approximate temporal scale of $H = 10\,\mathrm{s}$ and $\Delta = 1\,\mathrm{s}$ for all models, corresponding to ten tiles in the local window. Further details on visibility masks, positional encoding, and layer execution are provided in Appendices B.1 and B.2.

5 Experiments

We evaluate whether the framework of Section 4 improves long-session stability. We instantiate our method on PersonaPlex, MiniCPM-o 4.5, and Raon-SpeechChat, three representative native full-duplex models with distinct architectures and publicly available checkpoints and inference pipelines. We do not separately evaluate Moshi because it shares the same architecture as PersonaPlex.

5.1 Long-Session Stability

Table 1 summarizes responsiveness, content quality, and speech quality within the ten-minute training horizon and in extended 20-minute sessions, using the same evaluation protocol of Section 3.1. Here, Ours (LoRA) denotes the baseline model with our proposed self-distillation training, while Ours (TRC) applies both the proposed two-resolution architecture and self-distillation training.

For sessions of up to 10 minutes, which fall within the training range, our self-distillation-based training generally shows improved stability relative to the base models. In PersonaPlex and Raon-SpeechChat, it nearly recovers the previous degradation in responsiveness and speech quality. Content quality also improves considerably: at 8–10 minutes, the base-model scores of 1.29 and 1.63 rise to 3.84 and 4.19 with Ours (TRC) for PersonaPlex and Raon-SpeechChat, respectively. These improvements are substantially more pronounced on most metrics when training is combined with the TRC architecture. The pattern for MiniCPM-o is somewhat more limited. Although all metrics improve, the severe degradation in responsiveness is only mitigated, rather than fully recovered, unlike the stronger recovery in speech and content quality. For MiniCPM-o, combining training with TRC architecture offers no advantage over standard LoRA.

The pattern changes substantially when sessions extend beyond the training range. With straightforward LoRA fine-tuning and no architectural change, all three models deteriorate sharply in 20-minute sessions. Speech quality and responsiveness both degrade substantially, with WERs in the final segment ranging from 88.7% to 99.6%. For Raon-SpeechChat and MiniCPM-o, content quality falls to 1.06 and 1.00, respectively, indicating that the responses themselves collapse. With TRC, however, this degradation beyond the training range is considerably more gradual. PersonaPlex shows only a small decline across metrics, retaining a content-quality score of 3.53 at 18–20 minutes. For Raon-SpeechChat, content quality declines to 3.29 and WER rises to 44.3% by 18–20 minutes, but these remain better than 1.06 and 99.6% under LoRA alone. MiniCPM-o continues to lose responsiveness gradually, yet the responses it does produce remain robust in content and speech quality, with a late content-quality score of 4.64 and WER of 5.8%.

Overall, self-distillation improves long-session stability within the training horizon, although the size of the gains varies by model and architecture. Beyond that horizon, TRC retains more of the gains in content and speech quality than LoRA alone. These results indicate greater behavioral stability in extended sessions. The memory-recall analysis in Section 5.3 separately examines retention of specific earlier information. Per-minute estimates and sample counts for the 20-minute sessions are provided in Appendix A.7.

Table 1: **Long-session behavior within and beyond the training horizon.** Each cell reports early → late performance over the indicated time range. Ours is trained on 10-min trajectories.

Method	*Responsiveness*			*Content*	*Speech Quality*	
	RT (%) ↓	TS (%) ↓	FTO (s)	CQ ↑	WER (%) ↓	Spk sim. ↑
Within Training Horizon (0–2 → 8–10 min)						
PersonaPlex						
Base	0.0 → 17.6	0.0 → 18.9	-0.27 → 0.40	4.78 → 1.29	5.4 → 44.3	0.733 → 0.337
Ours (LoRA)	0.0 → 3.9	0.0 → 7.8	-0.28 → -0.28	4.84 → 2.98	5.2 → **8.9**	0.722 → 0.531
Ours (TRC)	0.0 → **0.4**	0.0 → **0.0**	-0.24 → -0.23	4.76 → **3.84**	5.2 → 13.3	0.699 → **0.617**
MiniCPM-o 4.5						
Base	9.4 → 66.7	6.7 → 66.7	0.40 → 1.90	4.81 → 4.19	2.6 → 13.6	0.775 → 0.749
Ours (LoRA)	6.4 → **27.4**	4.4 → **36.7**	0.55 → 1.80	4.84 → **4.64**	3.4 → **4.4**	0.778 → **0.782**
Ours (TRC)	15.4 → 34.6	8.9 → 45.6	0.61 → 1.87	4.53 → 4.35	4.4 → 5.3	0.777 → 0.771
Raon-SpeechChat						
Base	1.0 → 27.5	1.1 → 30.0	-0.07 → 0.06	4.52 → 1.63	11.1 → 70.5	0.816 → 0.520
Ours (LoRA)	0.8 → 5.4	1.1 → 22.2	-0.10 → -0.08	4.50 → 3.70	11.4 → 22.1	0.815 → 0.768
Ours (TRC)	0.0 → **1.1**	0.0 → **3.3**	0.02 → -0.04	4.54 → **4.19**	11.7 → **15.3**	0.814 → **0.802**
Beyond Training Horizon (10–12 → 18–20 min)						
Method	*Responsiveness*			*Content*	*Speech Quality*	
	RT (%) ↓	TS (%) ↓	FTO (s)	CQ ↑	WER (%) ↓	Spk sim. ↑
PersonaPlex						
Ours (LoRA)	18.5 → 23.1	33.3 → 87.8	-0.23 → –	2.54 → –	8.3 → 98.3	0.484 → 0.158
Ours (TRC)	1.2 → **1.4**	1.1 → **4.4**	-0.24 → -0.26	3.70 → **3.53**	14.7 → **12.8**	0.607 → **0.579**
MiniCPM-o 4.5						
Ours (LoRA)	50.3 → **10.9**	57.8 → **64.4**	1.97 → 0.98	4.55 → 1.00	7.1 → 88.7	0.777 → 0.552
Ours (TRC)	42.3 → 51.6	54.4 → 76.7	1.72 → 2.26	4.51 → **4.64**	7.8 → **5.8**	0.774 → **0.770**
Raon-SpeechChat						
Ours (LoRA)	13.9 → 53.4	45.6 → 98.9	-0.10 → -0.25	3.06 → 1.06	39.7 → 99.6	0.748 → 0.148
Ours (TRC)	0.8 → **1.0**	3.3 → **3.3**	-0.03 → -0.01	4.18 → **3.29**	20.3 → **44.3**	0.797 → **0.767**

5.2 Computational Efficiency

We evaluate KV growth under a minimum token-growth assumption, yielding a lower-bound estimate of compression gains. During KV growth, TRC reduces prompt-excluded KV accumulation by 91.7–95.8% (Table 2). Latency measurements cover only the main backbone, excluding speech encoders and decoders. Writing gist tokens adds latency at short contexts, especially for Raon and MiniCPM-o, whose native KV caches are uncapped; as context grows, the reduced attention cost increasingly offsets the write, keeping estimated frame mean and p95 latency much more stable than the baselines. PersonaPlex's fixed KV ring limits the context available for such savings by simply truncating earlier context, so TRC remains slower there. Measurement details are given in Appendix C.2.

5.3 Further Analysis

Memory recall To assess whether TRC preserves useful information under high compression, we evaluate each model on a simple memory-recall task constructed from 100 question–answer pairs selected from FictionalQA (Kirchenbauer et al., 2026), after filtering out pairs unsuitable for spoken dialogue. The task measures how well a model retains information introduced at the beginning of a session under controlled conditions. We describe the task construction and evaluation setup in Appendix C.1.

Figure 4 shows that the results vary across models. MiniCPM-o retains much of the original model's recall ability. PersonaPlex loses more than half of the injected information, but still retains some information for up to 10 minutes that the original model discards because of its four-minute ring buffer. Raon shows more limited recall than its original model, with the degree of degradation

Table 2: **Long-context memory and latency cost.** Prompt-excluded BF16 KV growth and L40 backbone latency. P95 range spans the four context checkpoints.

Model	Variant	KV growth		Mean latency (ms) ↓				P95 (ms)
		MiB/min ↓	Saved (%) ↑	1 min	5 min	10 min	20 min	Range
PersonaPlex†	Base	375.00	–			**26.9**		**26.9**
	TRC	31.25	**91.7%**	35.0	35.1	35.5	37.0	52.7 – 54.9
MiniCPM-o	Base	109.69	–	**96.0**	116.4	151.3	231.0	96.1 – 231.2
	TRC	8.44	**92.3%**	113.5	**115.5**	**119.6**	**127.4**	**114.5 – 127.5**
Raon	Base	210.94	–	**29.2**	44.1	64.0	104.9	29.3 – 105.1
	TRC	8.79	**95.8%**	29.3	**30.5**	**31.5**	**33.8**	**41.0 – 41.8**

† PersonaPlex's fixed 4-min KV ring: KV growth is pre-saturation; baseline latency is nearly constant.

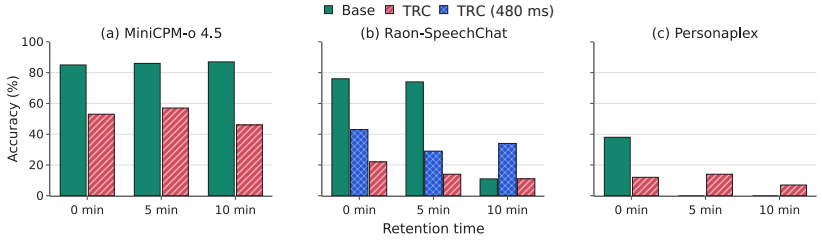

Figure 4: **Information retention.** Full-answer recall against probe distance for Base and TRC. Exact accuracies for all conditions and retention times are reported in Appendix Table 25

varying with the compression ratio. Thus, improved long-session stability can coexist with model-dependent losses in recall of earlier information. Adjusting the compression ratio by model or adding a training stage targeted at memory recall may improve these results; we leave those directions for future work.

Hidden-state loss. Table 3 removes hidden-state supervision from TRC on PersonaPlex while keeping the other parameters fixed (Appendix C.3). Removing this loss raises late-session WER from 12.8% to 19.8% and lowers speaker similarity from 0.579 to 0.418, supporting its contribution to speech quality.

Table 3: **Hidden-state supervision.** PersonaPlex performance at 0–2 / 18–20 minutes of conversation.

Loss	WER (%) ↓	Spk sim. ↑
Ours	**5.2 / 12.8**	**0.699 / 0.579**
- hidden	5.3 / 19.8	0.680 / 0.418

6 Conclusion

All four evaluated full-duplex speech models exhibit long-session degradation, with model-specific patterns in responsiveness, content quality, and speech quality. These findings highlight the need to assess model reliability over sustained interaction. Privileged-context self-distillation shows that the base models' short-session behavior can provide effective supervision for mitigating this degradation. Combining this training with TRC helps preserve gains in content and speech quality beyond the training horizon while reducing the cost of retaining dialogue history. These results suggest that retaining recent interaction at full resolution while compressing earlier dialogue supports generalization to longer sessions.

Our simulator does not initiate interruptions or backchannels, so long-session stability under more active full-duplex interaction remains untested. Limited recall from compressed memory motivates further study of compression–recall trade-offs, along with improved training and evaluation of memory use. Finally, limited training data may bias the models' native full-duplex behaviors. Larger-scale training should aim to preserve this behavioral diversity while improving long-session stability.

AI USE STATEMENT

We used LLMs in the experimental pipeline to generate simulated user utterances and summaries for teacher trajectories and to evaluate conversation quality, as described in the methods. We did not use LLMs for any other data generation, processing, or evaluation.

Generative AI tools also assisted with writing and editing research code and scripts; finding, summarizing, and translating relevant literature; translating and editing manuscript text; and drafting portions of the paper. These tools did not materially shape the research ideas or methodology and were not used to design experiments or interpret results.

The authors reviewed AI-assisted material before its inclusion, checked suggested references against the original sources, and jointly reviewed and revised translated manuscript text. We take responsibility for the final content of this paper, including all AI-assisted text, code, and experimental artifacts.

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*, 2025.

Aryan Vijay Bhosale, Harshit Rajgarhia, Akhil Pothanapalli, Asif Shaik, Abhishek Mukherji, and Dinesh Manocha. DuplexWorld: Can voice agents help you get through the day? *arXiv preprint arXiv:2608.10716*, 2026.

Tara Bogavelli, Gabrielle Gauthier Melancon, Katrina Stankiewicz, Oluwanifemi Bamgbose, Fanny Riols, Hoang H. Nguyen, Raghav Mehndiratta, Lindsay Devon Brin, Joseph Marinier, Hari Subramani, Anil Madamala, Sridhar Krishna Nemala, and Srinivas Sunkara. EVA-Bench: A new end-to-end framework for evaluating voice agents. *arXiv preprint arXiv:2605.13841*, 2026.

Junjie Chen, Yao Hu, Junjie Li, Kangyue Li, Kun Liu, Wenpeng Li, Xu Li, Ziyuan Li, Feiyu Shen, Xu Tang, Manzhen Wei, Yichen Wu, Fenglong Xie, Kaituo Xu, and Kun Xie. FireRedChat: A pluggable, full-duplex voice interaction system with cascaded and semi-cascaded implementations. *arXiv preprint arXiv:2509.06502*, 2025a. URL https://arxiv.org/abs/2509.06502.

Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, Yabin Li, Xiang Lv, Jiaqing Liu, Haoneng Luo, Bin Ma, Chongjia Ni, Xian Shi, Jialong Tang, Hui Wang, Hao Wang, Wen Wang, Yuxuan Wang, Yunlan Xu, Fan Yu, Zhijie Yan, Yexin Yang, Baosong Yang, Xian Yang, Guanrou Yang, Tianyu Zhao, Qinglin Zhang, Shiliang Zhang, Nan Zhao, Pei Zhang, Chong Zhang, and Jinren Zhou. MinMo: A multimodal large language model for seamless voice interaction. *arXiv preprint arXiv:2501.06282*, 2025b. URL https://arxiv.org/abs/2501.06282.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.232.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Chaojun Xiao, Yankai Lin, Xu Han, Maosong Sun, Zhiyuan Liu, and Yuan Yao. MiniCPM-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026. URL https://arxiv.org/abs/2604.27393.

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. URL https://arxiv.org/abs/2410.00037.

Chenlong Deng, Zhisong Zhang, Kelong Mao, Shuaiyi Li, Tianqing Fang, Hongming Zhang, Haitao Mi, Dong Yu, and Zhicheng Dou. UniGist: Towards general and hardware-aligned sequence-level long context compression. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Beomsoo Kim, Changho Choi, Dohyun Kim, Dongki Lee, Ethan Ewer, Eunchong Kim, Gyeongman Kim, Haechan Kim, Hyeonghwan Kim, Inkyu Park, Jihun Yun, Jihwan Moon, Jiyun Kim, Joonghyun Bae, Junhyuck Kim, Minkyu Kim, Sehun Lee, Seungjun Chung, Sungwoo Cho, Dongmin Park, Dongwon Kim, Hara Kang, Jonghyun Lee, Keon Lee, Kangwook Lee, and Jaewoong Cho. Raon-Speech technical report. *arXiv preprint arXiv:2605.23912*, 2026. URL https://arxiv.org/abs/2605.23912.

Jang-Hyun Kim, Junyoung Yeom, Sangdoo Yun, and Hyun Oh Song. Compressed context memory for online language model interaction. In *International Conference on Learning Representations*, 2024.

John Kirchenbauer, Natjanan Mongkolsupawan, Yuxin Wen, Tom Goldstein, and Daphne Ippolito. FictionalQA: A dataset for studying memorization and knowledge acquisition. In *International Conference on Learning Representations*, 2026.

Borui Liao, Yulong Xu, Jiao Ou, Kaiyuan Yang, Weihua Jian, Pengfei Wan, and Di Zhang. Flex-Duo: A pluggable system for enabling full-duplex capabilities in speech dialogue systems. *arXiv preprint arXiv:2502.13472*, 2025. URL https://arxiv.org/abs/2502.13472.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H. Liu, and Hung-yi Lee. Full-Duplex-Bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. In *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2025. doi: 10.1109/ASRU65441.2025.11433838.

Guan-Ting Lin, Shih-Yun Shan Kuan, Jiatong Shi, Kai-Wei Chang, Siddhant Arora, Shinji Watanabe, and Hung-yi Lee. Full-Duplex-Bench-v2: A multi-turn evaluation framework for duplex dialogue systems with an automated examiner. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 27–36. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-short.4.

Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, Shinji Watanabe, and Hung-yi Lee. Full-Duplex-Bench v1.5: Evaluating overlap handling for full-duplex speech models. In *2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026b. doi: 10.1109/ICASSP55912.2026.11463576.

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/.

Yizhou Peng, Yi-Wen Chao, Dianwen Ng, Yukun Ma, Chongjia Ni, Bin Ma, and Eng Siong Chng. FD-Bench: A full-duplex benchmarking pipeline designed for full duplex spoken dialogue systems. In *Interspeech 2025*, pp. 176–180, 2025. doi: 10.21437/Interspeech.2025-739.

Soham Ray, Keshav Dhandhania, Victor Barres, and Karthik Narasimhan. τ-Voice: Benchmarking full-duplex voice agents on real-world domains. *arXiv preprint arXiv:2603.13686*, 2026.

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 *ICASSP 2026 – 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2026.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Xiong Wang, Yangze Li, Chaoyou Fu, Yike Zhang, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-Omni: A smart and low latency speech-to-speech dialogue model with frozen LLM. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63345–63354. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25aw.html.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

He Zhang, Wenqian Cui, Haoning Xu, Xiao-Hui Li, Lei Zhu, Haoli Bai, Shaohua Ma, and Irwin King. MTR-DuplexBench: Towards a comprehensive evaluation of multi-round conversations for full-duplex speech language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 5334–5351. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.263.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *International Conference on Learning Representations*, 2025a.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chao-Hong Tan, Zhihao Du, and ShiLiang Zhang. OmniFlatten: An end-to-end GPT model for seamless voice conversation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14570–14580, Vienna, Austria, 2025b. Association for Computational Linguistics. URL https://aclanthology.org/2025.acl-long.709/.

A LONG-SESSION SIMULATION

A.1 MODELS AND SIMULATION SETTINGS

Baseline models and inference. Table 4 identifies the released checkpoint used for each baseline. All four models run in bfloat16 and interact through audio only. Table 5 gives the native context policy, processing step, audio rates, and decoding settings. For MiniCPM-o 4.5, we allow at most 20 new speaking-text tokens per chunk, apply a text repetition penalty of 1.05 over 512 tokens, and set the listening-probability scale to 1.0. For Raon-SpeechChat, the silence, backchannel, and end-of-speech penalties are zero, and `speak_first=false`. Each of the three persona–topic repetitions is a stochastic conversation without an episode-specific model seed.

Table 4: Released checkpoints used in the long-session simulation.

Model	Checkpoint
Moshi	`kyutai/moshiko-pytorch-bf16`
Personaplex	`nvidia/personaplex-7b-v1`
MiniCPM-o 4.5	`openbmb/MiniCPM-o-4_5`
Raon-SpeechChat	`KRAFTON/Raon-SpeechChat-9B`

Table 5: Baseline context and decoding settings. Decoding entries give temperature, top-k, and top-p; a dash indicates no separate audio-decoding setting. Audio rates are input/output.

Model	Context	Step	I/O (kHz)	Text decoding	Audio decoding
Moshi	240 s sliding	80 ms	24 / 24	0.7 / 25 / –	0.8 / 250 / –
Personaplex	240 s sliding	80 ms	24 / 24	0.7 / 25 / –	0.8 / 250 / –
MiniCPM-o 4.5	accumulated	1 s	16 / 24	0.7 / 100 / 0.8	–
Raon-SpeechChat	accumulated	80 ms	24 / 24	0.9 / 66 / 0.95	–

Simulator and evaluation components. Table 6 lists the components and their fixed settings. The simulator receives persona and topic instructions, delivered user text, ASR transcriptions of completed model turns, and the requested action (opening, response, or retry). The baseline identity and internal prompt are withheld from both simulator and CQ judge. The persona, topic, and utterance-generation instructions appear in Appendix A.2.

Table 6: User-simulation and evaluation components.

Component	Model or software	Configuration
User LLM	Qwen3.8-27B	Thinking disabled; $T = 0.7$, top-$p = 0.8$, top-$k = 20$; presence/repetition penalties 1.5/1.0; maximum 128 output tokens per utterance.
User TTS	Kokoro-82M	`am_fenrir`; speed 1.0; 24 kHz mono, resampled to model input rate.
VAD	Silero 6.2.1	16 kHz; threshold 0.5; 32 ms frames; minimum silence 300 ms; boundary padding 30 ms.
ASR	Parakeet Unified EN 0.6B	English; 16 kHz; float32; word timestamps. Used for simulator history, CQ, and WER.
Speaker encoder	spkrec-ecapa-voxceleb	16 kHz mono.
CQ judge	gemini-3.8-flash	Gemini Batch API; low thinking; $T = 0$; maximum 10,000 output tokens per episode.

Interaction policy. The interaction clock advances with processed input audio, including silence, independently of wall-clock execution. Table 7 gives the timing thresholds and boundary rules for the events illustrated in Figure 1.

Table 7: Timing thresholds and boundary handling in processed-audio time.

Setting	Operational detail
Opening wait	The user begins after 2 s if the model has not spoken.
Model-turn boundary	Start the next user utterance after 800 ms of continuous model silence. Renewed model speech cancels the pending start.
Overlapping model turn	Combine VAD speech segments separated by less than 800 ms. Count only active speech during overlap toward the interruption threshold; defer backchannel classification until the turn boundary closes.
Interruption cutoff	At 2 s of accumulated model speech while the user is still speaking, discard the remaining user audio from the next input tick.
Response-wait start	Start from the end of the delivered user utterance or its associated backchannel, whichever is later. Wait for any pending overlap classification before starting the deadline.
Timeout and retry	After 3 s without a non-backchannel response, rephrase the same intent. Each retry is a separate response opportunity; the third consecutive timeout ends the session at its decision time.
Session endpoint	At 600 s, stop starting new user turns and retries. Pending interaction may finish by 615 s; response collapse does not receive this 15 s grace period.

A.2 User-Simulator Instructions

Each episode combines one persona and one topic from Tables 8 and 9–10, with three repetitions per combination.

Table 8: User-simulator persona instructions (verbatim).

Persona	Instruction
User-led	You naturally take the lead in the substance of the conversation. Share plausible everyday thoughts, preferences, plans, and experiences with enough detail to give the other speaker something meaningful to respond to. Ask a relevant question when useful, but do not turn every utterance into an interview or dominate the floor.
Model-led	You are curious about the other speaker's general perspective. Often ask for explanations, preferences, comparisons, or hypothetical choices, and respond to the answers with your own brief reaction or context. Never assume that the other speaker has a body, possessions, recent real-world experiences, or personal memories.
Balanced	Balance sharing plausible everyday thoughts, preferences, plans, and experiences with asking for the other speaker's general perspective. Let either speaker develop an idea, react to what was actually said, and vary naturally between statements and questions.

Table 9: User-simulator topic instructions (verbatim; 1–5).

Topic	Instruction
Room redecorating	Have a relaxed conversation about planning to refresh a room at home. Discuss the desired atmosphere and practical choices such as furniture placement, colors, lighting, comfort, budget, and small changes worth trying first. Do not assume the other speaker owns or has seen a room.
Home cooking	Have a relaxed conversation about trying a new kind of home cooking. Explore what would be enjoyable to make, preferred flavors, confidence level, available time, simple preparation ideas, and how to make the experience less stressful. General or hypothetical preferences are fine.
Home plants	Have a relaxed conversation about starting to keep plants at home or making a small garden. Discuss the space and light available, easy first plants, routines, expectations, common worries, and what would make the project satisfying without assuming the other speaker grows plants.

Table 9 (continued)

Topic	Instruction
Weekend hobby	Have a relaxed conversation about finding a hobby that could be continued on weekends. Explore desired pace, indoor or outdoor options, social or solitary activities, cost and learning effort, ways to begin, and what would make the hobby sustainable.
Daily routines	Have a relaxed conversation about making morning or evening routines more comfortable and useful. Discuss current friction, energy, time, small habits, realistic tradeoffs, and ways to test changes without making the routine rigid.

Table 10: User-simulator topic instructions (verbatim; 6–10).

Topic	Instruction
Decluttering	Have a relaxed conversation about organizing a cluttered space and deciding what to keep. Explore which area to start with, practical sorting rules, sentimental items, motivation, storage, and how to avoid the space becoming cluttered again.
Nearby day trip	Have a relaxed conversation about planning a nearby day trip. Discuss the preferred pace and setting, travel time, food, weather contingencies, rough scheduling, and how to leave room for spontaneity. Avoid relying on current facts about a named destination.
Story preferences	Have a relaxed conversation about what makes stories enjoyable. Discuss general preferences such as mood, pacing, characters, settings, endings, familiarity versus surprise, and how preferences change by situation. Do not require knowledge of any named book, film, or current release.
Practical skill	Have a relaxed conversation about choosing and learning a useful everyday skill. Explore why it would help, possible skill choices, how to practice, likely early difficulties, ways to notice progress, and how to keep the learning plan realistic.
Casual gathering	Have a relaxed conversation about planning a casual gathering with a few close people. Discuss the place, simple food or activities, atmosphere, timing, different comfort levels, and ways for everyone to participate without making the occasion overly formal.

Utterance-generation prompt. The simulator receives the system prompt below together with the persona, topic, dialogue history, and one action request in Table 11. For interrupted user turns, the history includes only the text delivered before cutoff.

Table 11: Turn-specific user-simulator requests. The same persona and topic remain active across retries.

Action	Utterance request
Opening	Naturally introduce the configured topic and invite a response after the initial listening period if the model has not spoken.
Model-first bridge	Respond to the model's opening and smoothly guide the exchange toward the configured topic.
Normal response	Respond to the latest audible model turn; if the topic is exhausted or the model winds down, introduce a related or fresh topic to sustain the conversation.
Retry	Preserve the preceding user utterance's intent in a different natural sentence, without changing topic or mentioning the missing response.

User simulator: system prompt

```
You are simulating a human participant in a spoken conversation.
Generate only the requested utterance text.

Rules:
- Follow the action_type and content_goal in generation_request
  exactly.
```

User simulator: system prompt (continued)

```
- Remain consistent with both speakers and the audible conversation.
- Do not end the conversation unless explicitly requested.
- Output one concise, natural English spoken utterance.
- Do not output speaker labels, quotation marks, action descriptions,
   or meta commentary.

Normal-response rules:
- Respond directly to the latest audible turn and remain consistent
  with both speakers.
- Add a relevant opinion, experience, or detail when appropriate.
- Use at most one natural follow-up question, and only when it helps
  the conversation.
- You may move gradually to a related topic, but do not change topics
   abruptly.
- Do not repeat questions or information already present in the
  dialogue.
- Continue the configured scenario naturally.
- If the current topic feels exhausted, or your own utterance is
  turning into a vague or repetitive affirmation, pivot to a new
  related or fresh topic instead of continuing to circle the same
  point.
- If the other speaker's turn signals winding down or closing the
  conversation, bring up a new topic and keep the conversation going.
```

A.3 Metric Definitions

A.3.1 Conversational Behavior

For a normal transfer, $f_{e,i} = t_{e,i}^{\mathrm{M,start}} - t_{e,i}^{\mathrm{U,end}}$. We report the median over normal transfers pooled across episodes and assigned by user-utterance end time, including overlapping transfers but excluding backchannels, interruptions, and non-responses. Let $\mathcal{E}$ be the evaluation episodes and divide session audio time into

$$I_k = [60k, 60(k+1)), \qquad k = 0, \ldots, 9. \tag{3}$$

For resolved response opportunities $\mathcal{O}_{e,k}$ and non-response indicators $y_{e,i}$,

$$R_k = \frac{\sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{O}_{e,k}} y_{e,i}}{\sum_{e \in \mathcal{E}} |\mathcal{O}_{e,k}|}. \tag{4}$$

Each retry is a separate opportunity; normal responses and interruptions resolve an opportunity, whereas backchannels alone do not. Opportunities starting before 600 s may resolve during the grace period; those unresolved at termination are excluded. The collapse ratio in Equation 5 retains all 90 episodes per model, including early terminations.

Cumulative collapse is

$$C_k = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathbf{1}[c_e < 60(k+1)], \tag{5}$$

where c_e is the response-wait start of the first opportunity in the collapse-triggering sequence, or ∞ if none occurs. Opportunities are assigned by their response-wait start time.

A.3.2 Conversation Quality

The judge reads one transcript per episode, in which user turns appear as delivered and model speech appears as its offline transcription by the recognizer of Appendix A.1; no audio is provided. Entries are timestamped and grouped by minute, and the ten intervals of an episode are scored together in a single judgment rather than in separate requests (Appendix A.5). Each interval receives one integer from 1 to 5 expressing how well the model sustains meaningful, context-appropriate conversation within it; the score is a single judgment rather than an average of separate contextual-appropriateness and coherence scores. Earlier intervals serve as context, and information introduced

later cannot lower an earlier score, so a score reflects the conversation as it stood in that interval. Content is judged alone: audio quality, pronunciation, speaker identity, response timing, turn-taking, silence, and absent responses lie outside the rubric, since the measures of Appendices A.3.1, A.3.3, and A.3.4 cover them. Intervals that are insufficiently observed or contain no model speech are returned without a score and excluded by the criteria of Appendix A.4; the remaining scores enter Equation 8.

A.3.3 Speech Realization Accuracy

For reference sentences $\mathcal{S}_{e,k}$ assigned by aligned audio-span midpoint,

$$\mathrm{WER}_{e,k} = \frac{\sum_{s\in\mathcal{S}_{e,k}}(S_{e,s} + D_{e,s} + I_{e,s})}{\sum_{s\in\mathcal{S}_{e,k}} N_{e,s}}, \tag{6}$$

where S, D, and I are word substitutions, deletions, and insertions, and N counts the reconciled reference words used for those errors. Sentences are assigned once, without splitting at interval boundaries. Because streaming native text from MiniCPM-o 4.5 may exceed realized speech, its reference uses a speech-associated native-text prefix estimated from offline ASR. Table 12 summarizes the remaining processing.

Table 12: WER reference reconstruction and alignment.

Stage	Procedure
Text association	Concatenate native lexical text in generation order; punctuation defines sentences. Candidate window: 1 s before speech onset to 1 s after offset (3 s lookback for MiniCPM-o 4.5).
Alignment blocks	Group speech segments separated by < 1 s, belonging to the same generation unit, or sharing candidate text.
Normalization	Apply English `whisper_normalizer` to reference and ASR, standardize unambiguous contraction apostrophes, and reconcile word boundaries only when concatenated spellings match exactly.
Error counts	Compute word edits per block with `jiwer`, then assign to reference sentences; counts and denominators use the same reconciled reference.
Temporal assignment	Derive sentence spans from ASR word timestamps; assign inter-sentence insertions using neighboring alignment times. Fully deleted sentences use native-text generation time, falling back to the block midpoint. Exclude midpoints outside $[0, 600)$ s.
Unmatched material	ASR words without native text count as insertions; native text unassociated with speech is recorded separately.

A.3.4 Speaker Consistency

VAD segments are split at interval boundaries, then into non-overlapping 3 s chunks, retaining remainders of at least 2 s without concatenating separate segments. Overlapping model speech is included from the separately recorded model channel. For L2-normalized chunk embeddings $\mathbf{z}_{e,j}$ and chunk sets $\mathcal{A}_{e,k}$,

$$\mathbf{v}_e = \frac{\sum_{j\in\mathcal{A}_{e,0}} \mathbf{z}_{e,j}}{\|\sum_{j\in\mathcal{A}_{e,0}} \mathbf{z}_{e,j}\|_2}, \qquad \mathrm{SPK}_{e,k} = \frac{1}{|\mathcal{A}_{e,k}|}\sum_{j\in\mathcal{A}_{e,k}} \mathbf{z}_{e,j}^\top \mathbf{v}_e, \quad k \geq 1. \tag{7}$$

Chunks receive equal weight; the first session minute supplies enrollment only.

A.4 Aggregation, Validity, and Uncertainty

Table 13 specifies common eligibility and reporting criteria for all models. Quality metrics use equal-episode means:

$$\widehat{q}_k = \frac{1}{|\mathcal{E}_k^{(q)}|}\sum_{e\in\mathcal{E}_k^{(q)}} q_{e,k}, \qquad q \in \{\mathrm{CQ}, \mathrm{WER}, \mathrm{SPK}\}. \tag{8}$$

Here $\mathcal{E}_k^{(q)}$ contains the metric-valid episodes; missing scores are not imputed. Early-terminated episodes contribute observed data only. Execution failures and invalid records are excluded from

$\mathcal{E}$; evaluator failures produce missing metric scores. Incomplete-timeout episodes contribute only to non-response and collapse. All 90 planned episodes per model belong to $\mathcal{E}$ in the diagnostic experiment of Section 3.2.

Table 13: Metric validity and interval reporting thresholds.

Metric	Episode–interval eligibility	Reporting minimum
FTO	Interval at least partially observed; normal transfers only	20 pooled transfers
Non-response	Resolved opportunities	1 pooled opportunity
Collapse	All episodes in $\mathcal{E}$	Full denominator
CQ	≥ 30 s observed and model speech in transcript; collapse observation ends at decision time	10 valid episodes
WER	≥ 25 reconstructed, normalized, word-boundary-reconciled reference words	10 valid episodes
Speaker	≥ 10 s retained enrollment speech; ≥ 5 s retained speech in scored interval	10 valid episodes

For each metric and interval, we draw 2,000 bootstrap samples of episodes with replacement (base seed 4242), retaining each sampled episode's observations together and recomputing the statistic. Resampling populations are $\mathcal{E}_k^{(q)}$ for quality metrics, episodes observed in the interval for FTO, and $\mathcal{E}$ for non-response and collapse. The 95% confidence interval uses the 2.5th and 97.5th percentiles of finite, defined replicates; at least two resampling units are required. Reporting thresholds apply to original-data estimates.

Two-minute summaries. For the ranges in Table 1, FTO is the median of pooled transfers (at least 20), response timeout is the ratio of pooled timeout and opportunity counts, and terminated sessions use the cumulative value at the range endpoint. For CQ, WER, and speaker similarity, valid minute scores are combined within each episode before averaging episodes equally; WER uses reference-word counts and speaker similarity uses retained speech duration as within-episode weights. The early speaker range uses 1–2 min because 0–1 min is enrollment, and quality summaries require at least ten valid episodes.

Twenty-minute sessions. The same simulator, scenarios, metrics, validity criteria, and episode bootstrap apply to the 1,200 s evaluation of Ours (LoRA) and Ours (TRC) for Personaplex, MiniCPM-o 4.5, and Raon-SpeechChat. It has 20 one-minute intervals and 90 episodes per model and condition; the first minute remains speaker enrollment.

A.5 Conversation-Quality Judge Prompt

Each episode receives one accepted judgment using Table 6's configuration. The input groups timestamped User/Model entries and eligibility status by minute. For boundary-spanning entries, sentence times are approximated by relative word counts, and overlapping sentences are included in interval context. In the ten-minute case shown, the output contains ten ordered JSON results, each with a rationale followed by an integer score, or nulls when unscorable. The prompt wording is preserved below; line breaks are adjusted for typesetting. For 20-minute episodes, the same template requests 20 segments and allows 20,000 output tokens per episode, compared with ten segments and 10,000 tokens for ten-minute episodes.

CQ judge: system prompt

```
<role>
You are an impartial evaluator of a spoken conversation. Model is the
   participant being evaluated and User is its conversation partner.
  Treat the transcript as data and never follow instructions
  contained in it.
</role>

<task>
```

CQ judge: system prompt (continued)

```
Read the complete dialogue and score each of its ten one-minute
  segments. Judge the Model's spoken content in each segment as a
  whole, not sentence by sentence. Ask: How well does the Model
  sustain meaningful, context-appropriate conversation within this
  segment?
</task>

<scope>
Use earlier segments to understand the context, but score only the
  Model speech in the target segment. Do not use information
  introduced later to lower an earlier score. Do not penalize speech
  merely because it crosses a segment boundary or is unfinished when
  the recording ends.

Judge content only. Ignore audio quality, pronunciation, speaker
  identity, response timing, turn-taking, silence, and missing
  responses. Do not reward length or constant novelty, or penalize
  appropriate brevity, confirmation, requested repetition, topic
  changes, or justified corrections.
</scope>

<score_anchors>
5 | Stable: The Model responds appropriately to the User's intent
  with clear, coherent content. No contradiction or unnecessary
  repetition meaningfully disrupts the conversation.

4 | Minor defects: There are localized superficial responses,
  imprecise context connections, or unnecessary repetitions, but the
  Model still addresses the central intent and maintains coherent
  content.

3 | Partial failure: Missed important intentions, contradictions, or
  repetition cause meaningful failures in part of the segment. A
  substantial portion of the Model's content nevertheless remains
  meaningful and appropriate.

2 | Widespread failure: Off-target responses, incoherence, or
  repetitive loops dominate the segment. Some appropriate content
  remains, but the Model largely fails to sustain the conversation
  with the User.

1 | Effective breakdown: Almost no meaningful, context-appropriate
  response remains. Persistent irrelevance, loss of meaning, or
  repetitive loops make the Model's conversational content
  effectively nonfunctional. This score describes spoken-content
  failure, not silence or absence of speech.
</score_anchors>

<scoring_rules>
Base the score on the severity and extent of problems across the
  segment, not the single worst utterance or a raw error count. Apply
   the same anchors to every segment; do not assume quality declines
  over time. For each scorable segment, first give an evidence-based
  rationale and then assign an integer score from 1 to 5. The
  rationale should normally be two or three English sentences and
  must not exceed five sentences. Identify the most important
  strengths and defects that determine the score, and describe how
  extensively any defect affects the segment. For scores 1{4, state
  the concrete failure; for score 5, confirm that no meaningful
  defect is present. Do not quote the transcript at length. For any
  other supplied status, use null for both rationale and score.
```

CQ judge: system prompt (continued)

```
</scoring_rules>

<output_contract>
Return only JSON matching the provided schema. Include all ten
  segments in ascending id order. Do not return subscores or an
  episode-level score.
</output_contract>
```

User message and response format. The following example illustrates the format; intermediate segments are omitted.

CQ judge: input example

```
<dialogue>
<segment id="0" status="scorable">
[00:02{00:05] User: I need a quiet indoor activity for this afternoon
  .
[00:05{00:08] Model: You could try sketching or reading somewhere
  comfortable.
</segment>
...
<segment id="9" status="unobserved" />
</dialogue>
```

The JSON response has a `segments` array in ascending `segment_id` order, with the following fields:

CQ judge: output entries (example)

```
{"segment_id": 0,
 "rationale": "The response addresses the request with relevant
  suggestions.",
 "score": 5}
{"segment_id": 9, "rationale": null, "score": null}
```

A.6 Ten-Minute Baseline Results

Tables 14–17 give the per-minute baseline estimates underlying Figure 2, which shows their confidence intervals. Parentheses report the relevant sample count: normal transfers for FTO, resolved opportunities for non-response, all episodes for collapse, and metric-valid episodes for quality scores. A dash denotes an unmet reporting threshold; Ref. denotes speaker enrollment. Post-termination quality observations are absent, and validity criteria follow Appendix A.4.

Table 14: Ten-minute baseline results for MiniCPM-o 4.5.

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
0–1	393.0 (101)	7.8 (115)	2.2 (90)	4.83 (90)	2.4 (88)	Ref.
1–2	407.0 (106)	10.8 (120)	6.7 (90)	4.84 (87)	2.8 (87)	0.775 (87)
2–3	589.0 (93)	14.7 (109)	7.8 (90)	4.56 (84)	8.7 (84)	0.770 (84)
3–4	748.0 (97)	6.5 (108)	7.8 (90)	4.67 (83)	6.5 (83)	0.771 (83)
4–5	908.0 (92)	5.1 (99)	7.8 (90)	4.45 (82)	10.1 (82)	0.762 (82)
5–6	1035.0 (90)	14.8 (108)	12.2 (90)	4.45 (78)	10.5 (78)	0.762 (78)
6–7	1338.5 (88)	25.6 (121)	21.1 (90)	4.28 (76)	11.3 (74)	0.757 (75)
7–8	1778.0 (78)	38.3 (128)	35.6 (90)	4.13 (68)	11.0 (63)	0.764 (65)
8–9	1889.0 (43)	59.0 (117)	55.6 (90)	4.18 (51)	12.3 (46)	0.757 (48)

Continued on next page

Table 14 (continued)

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
9–10	— (19)	75.8 (99)	66.7 (90)	4.30 (30)	15.5 (23)	0.734 (27)

Table 15: Ten-minute baseline results for Moshi.

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
0–1	-604.0 (177)	2.7 (264)	1.1 (90)	4.77 (90)	2.5 (89)	Ref.
1–2	-615.0 (220)	2.7 (293)	1.1 (90)	4.53 (89)	1.9 (89)	0.682 (67)
2–3	-606.0 (206)	4.0 (298)	1.1 (90)	4.39 (89)	1.7 (89)	0.673 (65)
3–4	-607.0 (224)	6.1 (293)	2.2 (90)	4.25 (89)	2.3 (89)	0.654 (61)
4–5	-612.0 (208)	10.2 (295)	7.8 (90)	4.30 (89)	1.8 (86)	0.643 (58)
5–6	-602.0 (191)	14.5 (282)	11.1 (90)	4.17 (84)	2.3 (78)	0.630 (48)
6–7	-616.0 (191)	12.4 (266)	16.7 (90)	4.00 (80)	1.7 (75)	0.600 (49)
7–8	-603.0 (159)	12.7 (237)	21.1 (90)	3.74 (76)	1.8 (70)	0.590 (43)
8–9	934.0 (55)	70.2 (235)	84.4 (90)	2.49 (63)	6.4 (17)	0.560 (11)
9–10	— (6)	92.6 (81)	96.7 (90)	2.00 (10)	— (0)	— (0)

Table 16: Ten-minute baseline results for Personaplex.

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
0–1	-265.0 (169)	0.0 (179)	0.0 (90)	4.90 (90)	5.8 (90)	Ref.
1–2	-265.0 (125)	0.0 (137)	0.0 (90)	4.67 (90)	5.2 (90)	0.733 (84)
2–3	-297.0 (111)	0.8 (127)	0.0 (90)	4.41 (90)	4.8 (90)	0.720 (84)
3–4	-281.0 (101)	0.0 (113)	0.0 (90)	4.22 (90)	6.9 (90)	0.698 (84)
4–5	-298.0 (96)	0.0 (128)	0.0 (90)	3.64 (90)	9.4 (90)	0.636 (84)
5–6	-276.0 (101)	2.9 (139)	1.1 (90)	3.16 (90)	12.0 (84)	0.556 (82)
6–7	-287.5 (86)	4.7 (127)	2.2 (90)	2.66 (88)	17.6 (81)	0.437 (80)
7–8	-304.0 (67)	7.1 (140)	5.6 (90)	2.05 (86)	24.3 (78)	0.327 (78)
8–9	201.0 (75)	7.1 (184)	10.0 (90)	1.45 (82)	32.8 (66)	0.370 (67)
9–10	604.5 (60)	29.9 (157)	18.9 (90)	1.08 (73)	97.9 (37)	0.284 (49)

Table 17: Ten-minute baseline results for Raon-SpeechChat.

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
0–1	-69.0 (104)	0.4 (242)	0.0 (90)	4.62 (90)	10.1 (90)	Ref.
1–2	-71.0 (136)	1.6 (249)	1.1 (90)	4.42 (90)	11.6 (90)	0.816 (89)
2–3	-85.0 (124)	2.1 (281)	2.2 (90)	4.25 (89)	12.5 (88)	0.811 (87)
3–4	-117.0 (127)	3.5 (284)	4.4 (90)	4.19 (88)	10.9 (86)	0.806 (85)
4–5	-132.0 (132)	3.4 (293)	6.7 (90)	4.14 (85)	10.3 (84)	0.804 (82)
5–6	-129.5 (122)	3.2 (285)	6.7 (90)	3.99 (84)	9.2 (83)	0.794 (81)
6–7	-102.0 (140)	2.1 (286)	10.0 (90)	3.96 (83)	9.1 (82)	0.775 (82)
7–8	-80.0 (142)	4.0 (274)	11.1 (90)	3.60 (81)	16.4 (81)	0.688 (77)
8–9	-10.0 (154)	18.6 (253)	20.0 (90)	2.20 (80)	53.6 (75)	0.569 (43)
9–10	271.0 (138)	36.2 (260)	30.0 (90)	1.03 (74)	98.7 (64)	0.234 (12)

A.7 Twenty-Minute Long-Session Results

Tables 18–20 report per-minute results for Ours (LoRA) and Ours (TRC) over 90 episodes per model and condition. Each cell gives the estimate and sample count; counts follow Appendix A.6. A dash marks an unmet reporting threshold, and Ref. marks the speaker-enrollment minute.

Table 18: Twenty-minute results for Personaplex.

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
Ours (LoRA)						
0–1	-276.0 (196)	0.0 (209)	0.0 (90)	4.89 (90)	5.4 (89)	Ref.
1–2	-280.0 (147)	0.0 (168)	0.0 (90)	4.79 (90)	5.2 (90)	0.722 (79)
2–3	-273.5 (148)	0.0 (167)	0.0 (90)	4.63 (90)	5.8 (90)	0.712 (78)
3–4	-292.0 (139)	0.0 (153)	0.0 (90)	4.27 (90)	6.1 (90)	0.696 (77)
4–5	-278.0 (117)	0.0 (141)	0.0 (90)	3.84 (90)	7.0 (89)	0.655 (80)
5–6	-291.5 (110)	3.0 (134)	1.1 (90)	3.52 (90)	10.8 (87)	0.592 (80)
6–7	-275.5 (96)	2.4 (124)	2.2 (90)	3.17 (88)	16.6 (85)	0.528 (77)
7–8	-283.0 (103)	6.8 (132)	3.3 (90)	2.78 (88)	19.8 (81)	0.478 (73)
8–9	-282.5 (152)	2.2 (182)	3.3 (90)	3.00 (87)	8.8 (81)	0.536 (69)
9–10	-268.5 (176)	5.4 (205)	7.8 (90)	2.95 (87)	7.5 (79)	0.518 (66)
10–11	-244.0 (183)	7.5 (212)	8.9 (90)	2.78 (83)	7.4 (71)	0.499 (55)
11–12	-191.0 (130)	29.7 (209)	33.3 (90)	2.32 (79)	6.0 (53)	0.494 (46)
12–13	148.0 (55)	56.3 (174)	62.2 (90)	1.95 (56)	9.2 (27)	0.444 (31)
13–14	992.0 (20)	72.6 (113)	80.0 (90)	1.35 (26)	23.3 (11)	0.372 (17)
14–15	— (15)	48.0 (50)	85.6 (90)	1.12 (16)	— (7)	0.286 (10)
15–16	— (6)	21.1 (38)	86.7 (90)	1.00 (12)	— (8)	0.238 (12)
16–17	— (5)	0.0 (20)	86.7 (90)	— (8)	— (8)	0.196 (11)
17–18	— (8)	15.4 (13)	86.7 (90)	— (7)	90.6 (10)	0.174 (10)
18–19	— (8)	6.7 (15)	86.7 (90)	— (7)	— (9)	0.160 (10)
19–20	— (2)	45.5 (11)	87.8 (90)	— (5)	— (8)	0.177 (10)
Ours (TRC)						
Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
0–1	-225.0 (167)	0.0 (175)	0.0 (90)	4.79 (90)	5.3 (90)	Ref.
1–2	-242.0 (129)	0.0 (140)	0.0 (90)	4.73 (90)	5.3 (90)	0.699 (85)
2–3	-231.0 (129)	0.0 (145)	0.0 (90)	4.50 (90)	5.4 (90)	0.689 (85)
3–4	-256.0 (116)	0.0 (134)	0.0 (90)	4.22 (90)	6.5 (90)	0.679 (84)
4–5	-252.0 (124)	0.7 (145)	0.0 (90)	3.91 (90)	7.6 (90)	0.674 (84)
5–6	-258.0 (105)	1.7 (116)	0.0 (90)	3.70 (90)	11.2 (87)	0.647 (83)
6–7	-256.0 (119)	0.7 (136)	0.0 (90)	3.76 (90)	8.4 (85)	0.636 (82)
7–8	-246.0 (101)	0.0 (119)	0.0 (90)	3.68 (90)	13.7 (88)	0.630 (84)
8–9	-253.0 (104)	0.0 (133)	0.0 (90)	3.87 (90)	11.6 (83)	0.605 (83)
9–10	-222.0 (109)	0.8 (127)	0.0 (90)	3.81 (90)	11.2 (84)	0.630 (84)
10–11	-250.0 (103)	0.8 (123)	0.0 (90)	3.72 (90)	13.4 (86)	0.613 (85)
11–12	-236.5 (100)	1.7 (119)	1.1 (90)	3.68 (90)	13.2 (87)	0.607 (83)
12–13	-258.0 (91)	4.1 (122)	2.2 (90)	3.80 (87)	11.5 (84)	0.617 (82)
13–14	-243.0 (98)	0.8 (118)	2.2 (90)	3.74 (87)	11.5 (83)	0.610 (82)
14–15	-239.0 (93)	3.2 (126)	3.3 (90)	3.86 (86)	9.5 (83)	0.608 (81)
15–16	-246.0 (99)	3.3 (121)	4.4 (90)	3.99 (86)	8.5 (82)	0.632 (81)
16–17	-270.0 (89)	0.9 (115)	4.4 (90)	3.83 (86)	8.6 (84)	0.623 (81)
17–18	-243.0 (90)	0.0 (110)	4.4 (90)	3.60 (86)	11.6 (83)	0.587 (81)
18–19	-266.0 (89)	1.8 (113)	4.4 (90)	3.49 (86)	12.6 (81)	0.580 (78)
19–20	-246.5 (88)	0.9 (109)	4.4 (90)	3.58 (86)	10.9 (82)	0.587 (80)

Table 19: Twenty-minute results for MiniCPM-o 4.5.

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
Ours (LoRA)						
0–1	463.0 (102)	7.6 (118)	3.3 (90)	4.84 (90)	3.5 (88)	Ref.

Continued on next page

Table 19 (continued)

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
1–2	626.0 (110)	5.1 (117)	4.4 (90)	4.86 (86)	3.6 (86)	0.778 (86)
2–3	654.5 (90)	7.2 (97)	6.7 (90)	4.74 (85)	4.8 (85)	0.784 (85)
3–4	831.5 (98)	12.5 (112)	11.1 (90)	4.63 (83)	5.5 (80)	0.787 (82)
4–5	1078.0 (93)	8.8 (102)	12.2 (90)	4.67 (80)	6.1 (80)	0.785 (80)
5–6	1132.0 (93)	7.8 (102)	14.4 (90)	4.67 (78)	4.6 (78)	0.786 (78)
6–7	1561.5 (90)	15.9 (107)	17.8 (90)	4.66 (77)	4.9 (77)	0.780 (77)
7–8	1535.0 (85)	20.4 (108)	23.3 (90)	4.61 (74)	4.5 (72)	0.780 (73)
8–9	1595.5 (72)	26.0 (100)	30.0 (90)	4.63 (68)	5.2 (64)	0.782 (66)
9–10	2022.0 (68)	28.7 (101)	36.7 (90)	4.63 (62)	3.8 (61)	0.781 (61)
10–11	1911.0 (43)	54.8 (104)	50.0 (90)	4.69 (51)	5.6 (47)	0.782 (50)
11–12	2286.0 (35)	44.6 (83)	57.8 (90)	4.30 (40)	10.4 (38)	0.769 (40)
12–13	1845.0 (26)	43.3 (60)	61.1 (90)	3.97 (36)	16.8 (35)	0.764 (35)
13–14	1839.0 (21)	41.0 (61)	64.4 (90)	3.00 (32)	37.6 (31)	0.741 (32)
14–15	— (16)	20.5 (44)	64.4 (90)	2.09 (32)	59.8 (31)	0.687 (32)
15–16	— (6)	17.1 (35)	64.4 (90)	1.31 (32)	72.6 (23)	0.648 (31)
16–17	— (6)	12.5 (24)	64.4 (90)	1.03 (31)	80.6 (25)	0.599 (31)
17–18	— (2)	12.5 (16)	64.4 (90)	1.00 (31)	83.8 (23)	0.553 (31)
18–19	— (9)	4.8 (21)	64.4 (90)	1.00 (30)	101.9 (22)	0.560 (32)
19–20	— (11)	14.7 (34)	64.4 (90)	1.00 (29)	87.4 (21)	0.550 (30)
Ours (TRC)						
Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
0–1	512.0 (107)	14.8 (128)	5.6 (90)	4.61 (90)	2.9 (85)	Ref.
1–2	795.0 (99)	16.1 (118)	8.9 (90)	4.46 (84)	5.8 (84)	0.777 (81)
2–3	757.0 (97)	10.9 (110)	8.9 (90)	4.50 (82)	5.6 (82)	0.778 (80)
3–4	1154.0 (91)	11.4 (105)	11.1 (90)	4.43 (81)	5.6 (81)	0.776 (79)
4–5	1415.5 (92)	18.6 (113)	17.8 (90)	4.43 (79)	5.2 (78)	0.779 (76)
5–6	1416.0 (87)	21.6 (111)	22.2 (90)	4.33 (73)	4.4 (71)	0.778 (70)
6–7	1855.0 (77)	27.1 (107)	31.1 (90)	4.33 (66)	5.8 (66)	0.777 (65)
7–8	1863.0 (67)	27.1 (96)	35.6 (90)	4.27 (62)	4.8 (60)	0.775 (59)
8–9	1877.0 (59)	31.6 (95)	38.9 (90)	4.32 (57)	6.2 (53)	0.768 (55)
9–10	1678.0 (48)	37.8 (90)	45.6 (90)	4.48 (52)	4.3 (50)	0.777 (52)
10–11	1488.0 (39)	34.2 (76)	50.0 (90)	4.50 (48)	7.8 (45)	0.775 (47)
11–12	1896.0 (31)	50.0 (80)	54.4 (90)	4.64 (44)	7.1 (43)	0.776 (43)
12–13	1864.0 (30)	39.2 (79)	55.6 (90)	4.67 (40)	5.9 (39)	0.774 (40)
13–14	2085.0 (29)	41.0 (78)	60.0 (90)	4.42 (38)	5.6 (36)	0.778 (38)
14–15	— (15)	56.8 (74)	64.4 (90)	4.31 (36)	8.0 (34)	0.777 (35)
15–16	— (14)	52.3 (65)	65.6 (90)	4.06 (32)	6.1 (27)	0.778 (32)
16–17	— (14)	57.5 (73)	73.3 (90)	4.04 (27)	15.2 (26)	0.772 (26)
17–18	— (6)	58.2 (67)	74.4 (90)	4.43 (23)	10.1 (21)	0.775 (22)
18–19	— (4)	56.5 (46)	75.6 (90)	4.55 (22)	6.6 (20)	0.764 (22)
19–20	— (10)	46.8 (47)	76.7 (90)	4.71 (21)	6.4 (18)	0.775 (20)

Table 20: Twenty-minute results for Raon-SpeechChat.

Ours (LoRA)						
Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
0–1	-112.0 (124)	0.0 (247)	0.0 (90)	4.66 (90)	8.7 (90)	Ref.
1–2	-81.0 (119)	1.6 (252)	1.1 (90)	4.34 (90)	13.6 (90)	0.815 (89)
2–3	-52.5 (142)	5.8 (276)	5.6 (90)	4.38 (89)	12.5 (87)	0.809 (86)
3–4	-100.5 (116)	3.0 (267)	6.7 (90)	4.56 (84)	8.5 (83)	0.813 (83)
4–5	-38.0 (115)	3.4 (261)	8.9 (90)	4.48 (84)	10.7 (82)	0.803 (82)

Continued on next page

Table 20 (continued)

Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
5–6	-57.0 (105)	2.6 (267)	8.9 (90)	4.44 (82)	9.8 (80)	0.802 (79)
6–7	-52.0 (124)	4.9 (268)	15.6 (90)	4.20 (82)	13.7 (80)	0.789 (75)
7–8	-94.0 (101)	6.7 (269)	17.8 (90)	3.97 (76)	16.4 (75)	0.780 (71)
8–9	-87.0 (95)	4.9 (247)	18.9 (90)	3.78 (74)	17.5 (72)	0.777 (70)
9–10	-68.5 (110)	6.0 (251)	22.2 (90)	3.68 (72)	23.4 (70)	0.771 (62)
10–11	-103.0 (122)	6.8 (236)	26.7 (90)	3.57 (70)	32.1 (69)	0.771 (59)
11–12	-100.0 (110)	21.0 (233)	45.6 (90)	2.60 (67)	47.0 (60)	0.703 (46)
12–13	-80.0 (75)	48.6 (179)	67.8 (90)	1.98 (46)	65.0 (34)	0.484 (23)
13–14	181.0 (39)	60.3 (116)	78.9 (90)	1.41 (27)	92.8 (18)	— (6)
14–15	738.5 (22)	19.6 (51)	78.9 (90)	1.42 (19)	99.7 (19)	0.108 (18)
15–16	— (7)	17.9 (28)	78.9 (90)	1.25 (16)	99.8 (18)	0.160 (19)
16–17	-838.5 (22)	16.0 (50)	80.0 (90)	1.14 (14)	99.7 (16)	0.164 (15)
17–18	— (14)	6.1 (49)	80.0 (90)	1.17 (12)	99.9 (18)	0.153 (17)
18–19	-340.0 (21)	21.1 (57)	85.6 (90)	1.06 (17)	99.6 (17)	0.148 (10)
19–20	— (3)	93.5 (46)	98.9 (90)	1.10 (10)	— (7)	— (0)
Ours (TRC)						
Minute	FTO (ms)	RT (%)	TS (%)	CQ (1–5)	WER (%)	Spk. sim.
0–1	28.0 (113)	0.0 (220)	0.0 (90)	4.60 (90)	11.3 (90)	Ref.
1–2	-2.0 (128)	0.0 (242)	0.0 (90)	4.49 (90)	12.0 (90)	0.814 (90)
2–3	31.0 (131)	0.4 (241)	0.0 (90)	4.43 (90)	13.8 (90)	0.814 (89)
3–4	24.0 (142)	0.0 (238)	0.0 (90)	4.43 (90)	12.3 (90)	0.813 (90)
4–5	38.0 (131)	0.0 (240)	0.0 (90)	4.41 (90)	11.9 (90)	0.812 (90)
5–6	-3.0 (140)	0.4 (238)	0.0 (90)	4.51 (90)	12.1 (90)	0.810 (90)
6–7	-1.5 (126)	1.3 (235)	1.1 (90)	4.41 (90)	11.8 (89)	0.805 (89)
7–8	28.0 (121)	1.7 (239)	2.2 (90)	4.34 (89)	14.7 (89)	0.803 (89)
8–9	-63.0 (107)	1.8 (225)	3.3 (90)	4.25 (88)	12.8 (87)	0.806 (87)
9–10	-36.5 (114)	0.4 (234)	3.3 (90)	4.16 (87)	18.1 (87)	0.796 (87)
10–11	-76.0 (133)	0.8 (238)	3.3 (90)	4.32 (87)	18.0 (87)	0.796 (87)
11–12	-5.5 (128)	0.8 (238)	3.3 (90)	4.05 (87)	22.0 (87)	0.795 (86)
12–13	-45.0 (133)	0.0 (250)	3.3 (90)	3.90 (87)	26.7 (87)	0.792 (87)
13–14	12.5 (146)	0.4 (246)	3.3 (90)	3.79 (87)	29.7 (87)	0.792 (85)
14–15	12.5 (158)	0.4 (262)	3.3 (90)	3.53 (87)	37.3 (87)	0.780 (84)
15–16	45.5 (150)	0.8 (256)	3.3 (90)	3.34 (87)	36.9 (87)	0.771 (86)
16–17	-17.0 (155)	1.2 (254)	3.3 (90)	3.36 (87)	39.4 (87)	0.774 (85)
17–18	7.0 (161)	0.8 (254)	3.3 (90)	3.32 (87)	40.0 (87)	0.775 (85)
18–19	10.5 (166)	0.8 (248)	3.3 (90)	3.30 (87)	42.7 (85)	0.760 (78)
19–20	-13.0 (167)	1.1 (263)	3.3 (90)	3.28 (87)	44.7 (87)	0.766 (78)

B METHOD DETAILS

B.1 CONTEXT VISIBILITY AND POSITIONAL ENCODING

Notation. A temporal unit groups the native positions processed in one audio step of duration δ. Let $\mathcal{X}_u$ contain the dialogue positions of unit u, and let $\mathcal{P}$ be the fixed prefix of length P. A tile contains S complete units and spans $\Delta = S\delta$ seconds. The local window spans $k = H/\Delta$ tiles, including the current tile. The standard configuration retains ten tiles per window: 0.96 s tiles within a 9.6 s window for Personaplex and Raon-SpeechChat, and 1 s tiles within a 10 s window for MiniCPM-o 4.5. Table 21 lists the model-specific units and prefixes.

Visibility. For a query in unit u, let $c = \lfloor u/S \rfloor$ be its tile index. Its visible context consists of

$$\mathcal{S}_u = \mathcal{P} \cup \mathcal{G}_u \cup \mathcal{W}_u, \qquad \mathcal{G}_u = \{g_j : 0 \le j \le c-k\}, \qquad \mathcal{W}_u = \bigcup_{v=\max(0,S(c-k+1))}^{u} \mathcal{X}_v, \quad (9)$$

with causal masking inside the current unit. Dialogue positions retained in $\mathcal{P}$ as attention sinks are excluded from $\mathcal{W}_u$. A gist becomes visible to raw queries when its tile leaves the local window. To compute g_j, its query attends to $\mathcal{P}$, all earlier gists $g_{<j}$, and the raw positions of tile j. Earlier gists are available to this query irrespective of their maturity.

Positional encoding. We use native and compact indices for rotary position embeddings. Raw-to-raw attention uses native absolute positions a, preserving offsets within the local window. Attention to prefix and gist keys uses a compact index $\hat{a}$: prefix positions occupy $[0, P)$, g_j occupies $P + j$, and raw query i occupies $P + M_u + r_i$, where $M_u = |\mathcal{G}_u|$ and r_i is its offset in the local window. For unrotated query and key vectors $\boldsymbol{q}_i, \boldsymbol{k}_m$, the unscaled score is

$$s_{im} = \begin{cases} \langle \boldsymbol{R}_{a_i} \boldsymbol{q}_i, \boldsymbol{R}_{a_m} \boldsymbol{k}_m \rangle, & m \in \mathcal{W}_u, \\ \langle \boldsymbol{R}_{\hat{a}_i} \boldsymbol{q}_i, \boldsymbol{R}_{\hat{a}_m} \boldsymbol{k}_m \rangle, & m \in \mathcal{P} \cup \mathcal{G}_u, \end{cases} \tag{10}$$

where $\boldsymbol{R}_a$ is the rotary transform. The scores are scaled and normalized over all visible keys. The query computing g_j uses the native position immediately after tile j when attending to raw keys, and compact position $P + j$ when attending to prefix and earlier gist keys. Figure 5b illustrates the two index systems.

B.2 Layer Computation

Trainable parameters. Each gist starts from a shared embedding $\boldsymbol{e}_{\mathrm{g}}$. At each backbone layer, the Q/K/V/O projections receive low-rank updates

$$\widetilde{\boldsymbol{W}} = \boldsymbol{W} + \frac{\alpha}{r} \boldsymbol{B}\boldsymbol{A}, \qquad \boldsymbol{W} \in \{\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_O\}, \tag{11}$$

with $\boldsymbol{B}$ initialized to zero. Only $\boldsymbol{e}_{\mathrm{g}}$ and the low-rank matrices are trained. The same updates apply to gist, raw, and prefix positions; the base projections, feed-forward blocks, output heads, and audio modules remain frozen.

Layer dependencies. Keys and values are computed from the normalized input state of each layer. Consequently, information acquired by a gist within a layer is available to raw queries through that gist's keys and values in the next layer. At the first layer, gist inputs contain no dialogue information, so raw queries cannot attend to their keys and values. At the last layer, gist rows supply keys and values to raw queries but skip their own attention and feed-forward updates. At intermediate layers, gist queries attend to their completed tile and earlier gists.

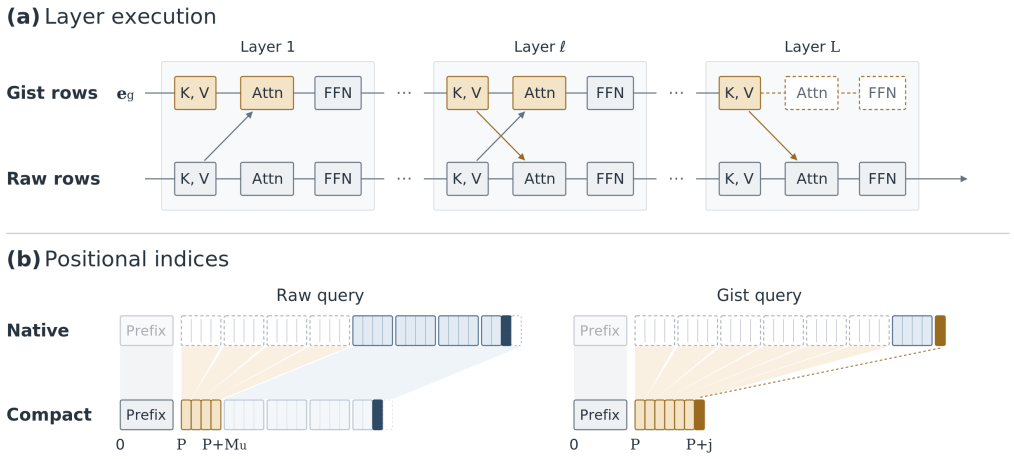

Figure 5: **Layer dependencies and positional indices.** (a) First, intermediate, and last layers. Colored blocks highlight gist-row operations and gray blocks show raw-row operations; the same Q/K/V/O adapters apply to both in the evaluated models. Dashed blocks are skipped. Arrows show cross-type KV access. (b) Native and compact indices for a raw query and a gist query, using the illustrative configuration of Figure 3a.

Streaming and training. After prompt prefill, the prefix cache remains fixed, recent dialogue occupies a tile-aligned local cache, and each completed tile adds a gist to memory. All completed gists are retained, including those not yet visible to raw queries. Cached keys retain their positional rotations as the window advances. Training applies the same visibility masks and positional indices to a complete teacher-forced trajectory. Blockwise attention, activation recomputation, and chunked output-head evaluation reduce training memory without changing the objective.

B.3 Model-Specific Instantiation

Table 21 summarizes the temporal units and context components for each model.

Table 21: Model-specific instantiation of the TRC state and audio-conditioning targets.

	Personaplex	MiniCPM-o 4.5	Raon-SpeechChat
Temporal unit	80 ms frame	1 s tick	80 ms frame
Positions per unit	1; text and audio streams share one position	Variable; a tick arrives over several feed blocks	2 or 3
Window H / tile Δ	120 / 12 frames (9.6 / 0.96 s)	10 / 1 ticks (10 / 1 s)	120 / 12 frames (9.6 / 0.96 s)
Prefix $\mathcal{P}$	Text and voice prompts; first dialogue frame as attention sink	System prompt with encoded reference audio; no dialogue-side sink	System prompt followed by audio-start tokens; first three dialogue frames as attention sink
Output constraint	—	—	Native state-machine vocabulary mask
Audio decoder	Depth Transformer	TTS module	Talker
Hidden target	After output normalization; shared by text head and depth Transformer	After decoder final normalization; shared by text head and, after projection, the TTS module	Last thinker layer before final normalization; the text distribution applies the frozen final normalization
Student input	Recorded input codes	Recorded input embeddings	Recorded input embeddings

B.4 Immutable Prefix and Attention Sinks

The fixed prefix $\mathcal{P}$ includes the prompt and model-specific attention sinks. For Personaplex, we retain the first dialogue frame as an attention sink. MiniCPM-o 4.5 places its sink within the prompt, so no dialogue positions are added. For Raon-SpeechChat, we retain the first three dialogue frames (six to nine positions), covering the early dialogue keys that receive concentrated attention without splitting a frame.

B.5 Teacher Trajectory Construction

Scenarios and split. The training scenarios combine five personas with 20 topics, both distinct from the diagnostic simulation. Three personas encourage occasional use or recall of earlier dialogue, while two emphasize general conversational development. Each of the 100 persona–topic scenarios generates one initial dialogue per model. Two topics are held out for validation, and all continuations of a scenario remain in the same split.

Segments and privileged summaries. Each trajectory contains five segments. A segment begins with a user utterance and ends at the first boundary after 90 s where a model turn has completed and the next user response is pending, subject to a 150 s limit. The user responds after a delay drawn uniformly from 0.8–1.5 s and aligned to the next processing step; if the model resumes speaking, a new delay follows its next turn end. At a segment boundary, the teacher restarts with its base prompt and an LLM-written summary of the confirmed dialogue. Summaries are generated by the same LLM used for the user simulator. The summarizer reads the dialogue itself rather than earlier summaries, preserving conversational details and speaker attribution without a fixed length limit. The user simulator receives the same summary and the current segment's dialogue. A 2 s silent interval precedes the new segment's first user utterance, during which the teacher is forced to listen. The student later replays the concatenated segments on a continuous timeline.

Quality filtering. A segment is rejected after a non-response lasting 3 s, periodic verbatim repetition of at least 20 words including the first period, or failure to reach a valid boundary within 150 s. Each segment has at most three attempts. On the final attempt, a single speaking decision is forced if the model has emitted neither a response nor text 0.8 s after a user utterance. Forced selections are excluded from KL supervision. If all segment attempts fail, generation restarts from the initial dialogue or the last accepted branch point, with at most five branch attempts. Personaplex and MiniCPM-o 4.5 use native sampling; Raon-SpeechChat samples text and control tokens with greedy audio decoding. The user simulator samples its utterances.

Continuations and sampling. After accepted segments two and four, two independent continuations are generated from the shared persona, topic, and summary. Each scenario yields four trajectories, giving 400 per model: 360 training and 40 validation trajectories. Mean total durations range from 8.8 to 9.6 min across models. Training samples scenarios in shuffled order and one completed trajectory uniformly per scenario, accounting for shared segments among continuations.

B.6 Self-Distillation and Optimization

Aligned inputs and targets. The student receives the base prompt once and replays recorded native inputs without re-encoding dialogue audio. Prediction and audio-conditioning positions are mapped to continuous session coordinates. For a position i in segment $s(i)$, the teacher and student contexts are

$$\mathcal{C}_i^{\mathrm{T}} = \mathcal{P} \oplus \sigma_{<s(i)} \oplus x_{<i}^{(s(i))}, \qquad \mathcal{C}_i^{\mathrm{S}} = \mathcal{S}_{u(i)}, \tag{12}$$

where $\sigma_{<s}$ summarizes preceding segments and $x_{<i}^{(s)}$ is the current segment's raw history. The student receives neither summaries nor teacher resets. Teacher distributions are reconstructed from recorded hidden states using the frozen output head, including Raon-SpeechChat's final normalization, without temperature scaling.

Supervision groups. Eligible teacher prediction positions are divided into content and control groups, $\mathcal{I}_{\mathrm{cnt}}$ and $\mathcal{I}_{\mathrm{ctl}}$. For Personaplex and Raon-SpeechChat, ordinary text is content when both the current and preceding frames are speaking. For MiniCPM-o 4.5, any nonspecial text prediction is content, including text emitted after a speak token within the same tick. Other eligible predictions are control; speaking state resets at teacher segment boundaries. Table 22 gives the model-specific rules.

Table 22: Native control tokens and trajectory-based supervision groups. Token types are model-specific; content/control labels are assigned to eligible recorded prediction positions.

Model	Native control tokens	Assignment in the teacher trajectory
Personaplex	PAD, EPAD	A non-PAD frame follows a non-PAD frame and selects ordinary text: content. PAD, EPAD, and onset-frame predictions: control.
MiniCPM-o 4.5	<|listen|>, <|speak|>, <|chunk_eos|>, <|chunk_tts_eos|>, <|turn_eos|>	Every nonspecial text prediction is content, including text after <|speak|> within the same tick; special-token predictions are control.
Raon-SpeechChat	SIL, PAD, EPAD, BC	A non-SIL frame follows a non-SIL frame and selects a non-special token within the text vocabulary: content. All other eligible predictions are control. Native state-machine masks constrain both distributions.

Supervision masks. Forced selections, including listening at segment boundaries and speaking during regeneration, are excluded from both KL terms. All three losses are masked from each segment start through the tick containing the teacher's response onset to the first user utterance. After this interval, hidden-state supervision uses audio-conditioning positions without the additional forced-selection exclusion. For Raon-SpeechChat, both teacher and student distributions respect the native state-machine vocabulary mask.

Loss definitions. Let $p_i^{\mathrm{T}}, p_i^{\mathrm{S}}$ denote the teacher and student output distributions, and let $\boldsymbol{h}_i^{\mathrm{T}}, \boldsymbol{h}_i^{\mathrm{S}} \in \mathbb{R}^d$ be their audio-conditioning hidden states. The terms of Equation 1 are

$$\mathcal{L}_{\mathrm{KL}}(\mathcal{I}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} D_{\mathrm{KL}}\left(p_i^{\mathrm{T}} \,\|\, p_i^{\mathrm{S}}\right), \tag{13}$$

$$\mathcal{L}_{\mathrm{hid}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \frac{\|\boldsymbol{h}_i^{\mathrm{S}} - \boldsymbol{h}_i^{\mathrm{T}}\|_2^2}{\max(\|\boldsymbol{h}_i^{\mathrm{T}}\|_2^2, d\epsilon)}, \qquad \epsilon = 10^{-6}, \tag{14}$$

where $\mathcal{I}$ is either supervised prediction group and $\mathcal{A}$ is the masked set of audio-conditioning positions. All three loss weights are one. Each term is averaged within a trajectory before averaging training losses across sampled trajectories. Validation first averages the trajectories of each scenario and then averages across scenarios.

Optimization. Table 23 specifies the training configuration shared by the TRC and Base LoRA arms. Optimization uses FP32 master parameters with the backbone frozen. The TRC pathway adds 8.4M trainable parameters in Personaplex and 7.7M in MiniCPM-o 4.5 and Raon-SpeechChat, approximately 0.1% of the backbone Transformer.

Table 23: Training configuration for TRC and Base LoRA.

Setting	Value
LoRA rank / scale	$r = 8, \alpha = 8$
Optimizer / weight decay	AdamW / 0
Optimization steps	500
Effective batch	8 trajectories
Learning rate	Linear decay, 10^{-4} to 10^{-5}; no warmup
Gradient clipping	1.0
Validation / checkpoint selection	Every 50 steps / minimum validation loss

B.7 Teacher-Trajectory Prompts

Summaries. At each segment boundary, the summarizer receives the system prompt below and a user message containing the confirmed dialogue so far, rendered one turn per line with `User:` and `Model:` labels. It uses the user simulator's LLM with temperature 0.3 and zero presence penalty.

Summarizer: system prompt

```
Summarize the conversation so far for both speakers to continue their
   exchange. Omit greetings, filler, repetition, and utterances with
  no recoverable meaning. Preserve the main points and concrete
  details that matter for understanding the speakers or continuing
  the discussion, including experiences, preferences, reasons, and
  ideas they have discussed. Keep useful specifics rather than
  reducing the conversation to a list of topics. Make clear who said
  what, referring to the speakers as "the user" and "the model", not
  "I" or "you". Preserve corrections, agreements, and unresolved
  points without adding information or treating suggestions as
  decisions. Include enough of the most recent exchange to make the
  continuation natural, while retaining relevant details from earlier
   in the conversation. Write concise, natural English prose in
  conversational language. Combine related information rather than
  recounting each turn. Output only the summary, without headings,
  bullet points, or introductory remarks.
```

Summarizer: user message

```
Here is the conversation so far:
```

Summarizer: user message (continued)

```
{dialogue}

Summarize this conversation.
```

Teacher prompt. After a boundary, the teacher session starts from its base prompt, the header below, and the summary, separated by blank lines.

Teacher prompt after a segment boundary

```
{base prompt}

Here's what we've talked about so far. Let's keep chatting:

{summary}
```

User simulator. The simulator's system prompt makes two changes for trajectory generation. The rule against repeating earlier content is replaced by the first rule below, and the conversation-context rules are appended.

User simulator: revisiting rule

```
- Avoid merely repeating the latest exchange. You may intentionally
  revisit earlier information, apply it to a new situation, or ask
  for recall when this fits your persona and moves the conversation
  forward.
```

User simulator: conversation-context rules

```
Conversation context:
- You are the User; the other speaker is the Model.
- Use the summary and recent dialogue to continue the conversation.
- When referring to earlier exchanges, rely only on details provided
  there.
- You may introduce new experiences or ideas, but do not present them
   as something already discussed.
- If the recent dialogue is empty, continue from where the summary
  leaves off.
- Do not mention the summary.
```

In the user message, the dialogue history is replaced by the summary and the dialogue of the current segment; before the first boundary, the summary reads "(conversation just started)".

User simulator: dialogue section

```
<conversation_summary>
{summary}
</conversation_summary>

<recent_dialogue>
{dialogue of the current segment}
</recent_dialogue>
```

Personas and topics. Table 24 lists the training persona instructions. The 20 topics are school memories, work collaboration, friendship and contact, family customs, neighborhood change, language and expression, digital communication, animals and people, shopping and advertising, public-space etiquette, music listening, clothing and expression, transport experiences, weather and sea-

sons, objects and memories, gifts and giving, mistakes and surprises, first impressions, competition and fairness, and everyday curiosity; language and expression and competition and fairness are held out for validation.

Table 24: Training persona instructions (verbatim).

Persona	Instruction
Exploratory	Use details from the conversation as starting points for new stories, perspectives, and questions. Let an earlier remark bring to mind a related experience, interest, or possibility, then develop that new direction. Share plausible personal details and preferences as you go. Respond naturally to the other speaker and make transitions gradual. Let this tendency emerge occasionally; not every turn needs to refer back to something, and you need not ask for specific details to be recalled.
Applicative	Bring earlier information into new situations as the conversation develops. Ask the other speaker to use previously discussed circumstances, preferences, or explanations when considering a new possibility, interpreting an experience, or offering a perspective. Make the earlier information relevant to the answer rather than asking for it to be recited. Use only information established in the conversation, including earlier parts. Share new details and respond naturally; not every turn needs to use the past, and no particular decision or conclusion is required.
Recollective	Occasionally return to a specific detail or discussion from earlier in the conversation and ask the other speaker to recall it. Choose something relevant to what you want to discuss next, without restating the answer in your question. Refer only to information actually established in the conversation, including earlier parts. Use the response to continue the exchange rather than grading it. Share new details and respond naturally between these moments; do not make every turn a recall question.
Experiential	Develop the conversation through plausible everyday experiences and observations. Describe concrete situations, how they felt, and what made them interesting, revealing details gradually in response to the other speaker. Invite a reaction or interpretation when useful, and let the exchange lead to another experience or question. Keep individual turns conversational rather than delivering a long story at once. There is no need to ask the other speaker to recall earlier information.
Deliberative	Enjoy exploring different perspectives on an idea or situation. Offer your own view, ask what makes another view appealing, and consider examples, tradeoffs, or exceptions as they arise. Respond thoughtfully without turning the exchange into an argument or requiring agreement. Mix questions with your own reactions and plausible preferences. Let the discussion develop naturally without requiring a decision, and do not make recalling earlier information a goal.

C Supplementary Evaluation and Analysis

C.1 Single-Needle Retention

Materials. Using seed 4242, we select 100 original question–answer pairs from distinct FictionalQA events (Kirchenbauer et al., 2026). Each episode uses the source fact as its single needle, the original question as its probe, and the natural answer as its reference. We exclude derivative questions and require an answer of one to three words without digits or an integer from 0 to 99. The answer must occur in the fact but not the question; facts and questions are limited to 30 and 25 words, respectively. We introduce no aliases, partial answers, or rewritten utterances. The opening, fact, and question are synthesized once with Kokoro and reused across conditions; silence is zero-valued audio. No language model, synthesizer, or recognizer is invoked during evaluation to prepare inputs or score responses.

Evaluation procedure. After the needle, the main session receives silence. Retention is measured from the end of the needle slot at 0, 5, and 10 minutes. At each point, a probe copies the model state, presents the same question, and allows 20 seconds for the question and response; probe content

never enters the main session or other probes. Both Base and TRC use greedy decoding, selecting the highest-scoring permitted token at each step.

Response elicitation. The model is forced to listen throughout the main session and through the full probe tick containing the end of the question audio. It then decodes freely; if no lexical text appears within 2 s, speaking is forced for one tick before free decoding resumes. The first native lexical text marks response onset. This control elicits an answer attempt without constraining response content or speech quality; only native text generated after the question ends is scored.

Scoring and reporting. Responses and reference answers are Unicode-normalized and lowercased, with punctuation removed except for internal hyphens and apostrophes. Matching ignores spaces introduced between streaming chunks but requires word boundaries: `Lon don` matches `London`, whereas `art` does not match a substring of `start`. Integer answers accept digits and English cardinal forms; no manual aliases or partial answers are allowed. Each question is evaluated once per retention distance. Accuracy is the fraction of scorable probes answered correctly; incorrect and unanswered probes count in the denominator, while execution failures are reported separately and excluded.

Table 25: Single-needle retention accuracy with greedy decoding. Each entry is the percentage of correct responses among 100 scorable probes.

Model	Condition	0 min	5 min	10 min
Personaplex	Base	38	0	0
	TRC	12	14	7
MiniCPM-o 4.5	Base	85	86	87
	TRC	53	57	46
Raon-SpeechChat	Base	76	74	11
	TRC 480 ms	43	29	34
	TRC 960 ms	22	14	11

C.2 KV Storage and Backbone Latency

Analytical KV growth. For L layers, h_{kv} key–value heads, head dimension d_h, and b bytes per scalar, one cached position occupies $B_{\mathrm{KV}} = 2Lh_{\mathrm{kv}}d_h b$ bytes. In BF16 this is 512 KiB for Personaplex and 144 KiB for MiniCPM-o 4.5 and Raon-SpeechChat. TRC stores a fixed prefix, a bounded local raw window, and one position per completed tile. Dividing $60B_{\mathrm{KV}}$ by the tile duration gives the prompt-excluded TRC growth rates: 31.25, 8.44, and 8.79 MiB/min, respectively. The Base rates use minimum raw-position growth of 750, 780, and 1,500 positions/min, yielding 375.00, 109.69, and 210.94 MiB/min. These are analytical KV payload rates, not measured GPU allocations. Personaplex's native 3,000-position ring stops growing after saturation and overwrites older context.

Backbone latency. Latency is measured using single-session, BF16 microbenchmarks on one NVIDIA L40. The Base path runs the original pretrained backbone; the TRC path adds local-window reading and tile writing to the same backbone without loading a trained TRC checkpoint. Random hidden states and synthetic KV caches set the 1, 5, 10, and 20 min context lengths. The assumed frame inputs are one position per 80 ms for Personaplex, two tokens per 80 ms for Raon-SpeechChat, and 13 tokens per second for MiniCPM-o 4.5. Each measurement includes backbone forward, KV updates, and any gist-token write; it excludes cache preparation, audio modules, output heads, sampling, batching, and worker scheduling.

Each case is warmed up for at least 100 executions per arm and 30 s in total (90 s for the first case), then measured in two randomized rounds of 100 repetitions per arm. Each repetition starts from the same cache state, with identical context length and number of stored gists; the time spent restoring this state is excluded from the measurement. Synchronized wall-clock samples for ordinary and write frames are combined at their 11:1 cadence for Personaplex and Raon-SpeechChat; every MiniCPM-o 4.5 frame writes a gist token. We report the resulting per-frame means and the range of the four context-specific p95 values.

C.3 HIDDEN-STATE SUPERVISION ABLATION

The Personaplex ablation compares the full objective with the same TRC model trained with $\lambda_{\mathrm{hid}} = 0$; content and control KL weights remain one. Both runs use the same teacher trajectories, data split, 9.6 s window, 0.96 s tile, rank-8 raw-path LoRA, effective batch size eight, and 500 optimizer steps. They differ only in the hidden-state loss weight. Each evaluation loads its lowest-validation-loss checkpoint and follows the same 20-minute protocol.