DUPLEXLEDGER: BENCHMARKING TRANSACTIONAL MEMORY IN FULL-DUPLEX SPEECH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Full-duplex speech agents must update persistent memory while listening and speaking, yet it remains unclear which events should be stored, revised, or excluded. Existing benchmarks isolate interaction timing from memory outcomes, obscuring the state transitions behind a correct answer. We introduce DuplexLedger, a bilingual benchmark of 1,094 persistent sessions aligning four full-duplex scenarios with event-level WRITE/UPDATE/IGNORE/RETAIN supervision, explicit pre/post memory states, and 6,564 same-session questions and checks. Evaluating five full-duplex models reveals an interaction–memory gap: no system combines strong boundary control with coherent final-state memory, and the best final-state fact coverage is 2.01%. We propose ACT-Mem, an 8.92M-parameter sidecar comprising an event controller, a versioned transactional ledger, and a typed retriever coupled with a frozen `MiniCPM-o` 4.5 backbone. On 219 held-out sessions, ACT-Mem increases latest-value accuracy from 4.11% to 8.68% while reducing stale assertions and distractor adoption. It improves interaction boundaries but slows interruption stopping. These results motivate selective, auditable full-duplex memory rather than indiscriminate transcript retention. Code is available at https://anonymous.4open.science/r/DuplexLedger.

1 INTRODUCTION

Speech is a natural interface for agents that assist people continuously. End-to-end spoken language models now support open-ended voice interaction (Zhang et al., 2023; Tang et al., 2024; Chu et al., 2024; Xie & Wu, 2024), while full-duplex systems listen while speaking, enabling interruptions, backchannels, and overlapping turns (Défossez et al., 2024; Wang et al., 2025c; Roy et al., 2026; Liu et al., 2026). Yet concurrency creates a state-management problem: an utterance may introduce or revise a fact, remain incomplete, serve only as a backchannel, or be addressed to someone else. A dependable agent must decide what was said and whether it should modify persistent memory.

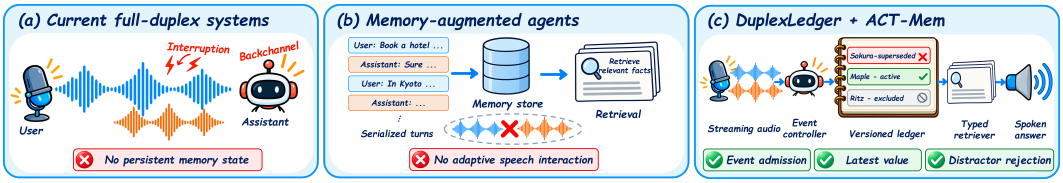


Figure 1: From isolated interaction and memory to transactional full-duplex memory.

Existing evaluation treats interaction and memory largely as separate capabilities. Full-duplex benchmarks emphasize turn taking, pause handling, overlap, and interruption, with recent extensions adding corrections and entity tracking (Lin et al., 2025; 2026b; Wang et al., 2026; Lin et al., 2026c; Modi et al., 2026). Memory methods and benchmarks instead study long-horizon storage, retrieval, and updating from serialized histories (Packer et al., 2023; Maharana et al., 2024; Wu et al., 2025; Xu et al., 2025), while recent speech agents begin to connect persistent memory with streaming interaction (Yao et al., 2025; Xie et al., 2026). The compared resources still do not jointly expose event admission, supersession, distractor exclusion, and memory-grounded spoken answers under adaptive concurrency. Without aligned state transitions, a correct answer cannot show whether the system maintained coherent memory or merely matched its current context.

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.

We formulate this capability as *transactional memory for full-duplex agents*: each event is admitted, rejected, or used to revise a versioned state. We introduce DuplexLedger, a bilingual benchmark of 1,094 persistent sessions aligning pause handling, speech to others, backchannels, and interruptions with WRITE, UPDATE, IGNORE, and RETAIN operations, explicit pre/post states, and same-session spoken questions. We further propose ACT-Mem, coupling an event controller, a versioned transactional ledger, and a typed retriever with a frozen full-duplex backbone. The controller determines admission and operation; the ledger records committed values and supersession through PREPARE, COMMIT, and ABORT; and the retriever supplies active, type-appropriate evidence only for memory QA. Streaming generation remains frozen, making memory transitions explicit and auditable.

Contributions. (1) We formulate full-duplex memory as selective event-level state construction. (2) DuplexLedger aligns adaptive interaction, reference state transitions, and spoken QA across four scenarios, 1,094 bilingual sessions, and 6,564 questions/checks. (3) ACT-Mem combines admission control, versioned memory, and typed retrieval with a frozen backbone, increasing latest-value accuracy from 4.11% to 8.68% while reducing stale assertions and distractor adoption.

2 Related Work

Full-Duplex Dialogue. Early spoken language models established end-to-end speech understanding and generation (Zhang et al., 2023; Tang et al., 2024; Chu et al., 2024; Xie & Wu, 2024). Full-duplex systems subsequently modeled synchronized channels, paired next-token prediction, or parallel audio streams to support concurrent listening and speaking (Nguyen et al., 2023; Veluri et al., 2024; Wang et al., 2025b; Défossez et al., 2024). `Freeze-Omni` retains a frozen language backbone, while `MiniCPM-o` 4.5, `PersonaPlex`, and `Lychee-FD` further target low-latency interaction, role control, and acoustic–semantic coordination (Wang et al., 2025c; Cui et al., 2026; Roy et al., 2026; Liu et al., 2026). These systems improve conversational timing, but generally provide no explicit policy for deciding which overlapping events should enter or revise persistent memory.

Memory-Augmented Agents. Memory-augmented agents manage conversational history through hierarchical storage, temporal updates, structured retrieval, or latent memory (Packer et al., 2023; Zhong et al., 2024; Sarthi et al., 2024; Jiménez Gutiérrez et al., 2024; Xu et al., 2025; Zhang et al., 2026). EgoMem and VoiceMem connect persistent memory to streaming multimodal interaction (Yao et al., 2025; Xie et al., 2026). These methods improve retention and retrieval but process session-level evidence after observation; they do not jointly control event admission, causal supersession, or exclusion of non-addressed and incomplete speech during concurrent interaction.

Full-Duplex and Audio Benchmarks. Full-Duplex-Bench evaluates pauses, backchannels, turn transitions, and interruptions; its overlap-focused extension adds speech to others and background speech (Lin et al., 2025; 2026b). HumDial-FDBench broadens human-recorded coverage, while adaptive benchmarks test correction handling, entity tracking, tool use, and interruption-conditioned state updates (Wang et al., 2026; Lin et al., 2026c;a; Modi et al., 2026). General audio benchmarks assess speech, scene, and paralinguistic understanding (Wang et al., 2025a); Audio Multi-Challenge also evaluates multi-turn inference memory, instruction retention, coherence, and spoken self-correction (Gosai et al., 2026). These protocols broaden audio-interaction evaluation but do not expose an event-level persistent-state ledger under adaptive full-duplex execution.

3 DuplexLedger: Benchmark Design

Full-duplex memory evaluation requires determining event content and whether it should update persistent state. DuplexLedger combines adaptive interaction challenges with event-level memory supervision, enabling joint evaluation of interaction behavior and memory fidelity.

3.1 Task Formulation

A session is an ordered event stream $\mathcal{E} = (e_1, \ldots, e_T)$, where $e_t = (x_t, s_t, a_t, \tau_t)$ specifies speech content, speaker, addressee, and timing. The memory state M_t contains items (k, v, σ) with entity key k, value v, and status $\sigma \in \{\textit{active}, \textit{superseded}\}$. Because one event may affect multiple entities, transitions are defined at the item level:

$$M_t = \mathcal{T}(M_{t-1},\, \{(k_i, v_i, o_i)\}_{i=1}^{n_t}), \qquad o_i \in \{\textsc{Write}, \textsc{Update}, \textsc{Ignore}, \textsc{Retain}\}. \tag{1}$$

Table 1: DuplexLedger versus representative full-duplex and conversational-memory benchmarks.

			Duplex		Memory			
Dataset	Scale	Hrs	Adaptive interaction	FD scenarios	State / Memory eval.	Update QA	Distractor reject	Event state
FDB v1/v1.5	727/499	NR	✗	4/4	✗	✗	✗	✗
HumDial-FDBench	15,698	$>$ 100	✗	9	✗	✗	✗	✗
MultiTalkBench	104	56.5	✗	N/A	✓	✗	✗	✗
FDB v2	200	N/A	✓	4	✓	✗	✗	✗
EchoChain	200	NR	✓	1	✓	✗	✗	✗
LoCoMo	10	N/A	✗	N/A	✓	✗	✗	✗
LongMemEval	500	N/A	✗	N/A	✓	✓	✗	✗
DuplexLedger (ours)	**1,094**	**$\geq$ 35.3**	✓	**4**	✓	✓	✓	✓

Scale denotes stimuli for FDB v1/v1.5, instances for HumDial-FDBench, samples for MultiTalkBench, tasks for FDB v2, interrupted conversations for EchoChain, conversations for LoCoMo, questions for LongMemEval, and sessions for DuplexLedger. *Update QA* denotes explicit evaluation of factual revision or knowledge updating; temporal reasoning alone is not counted. NR: aggregate audio duration is not reported; N/A: no fixed audio corpus or no comparable full-duplex scenario taxonomy.

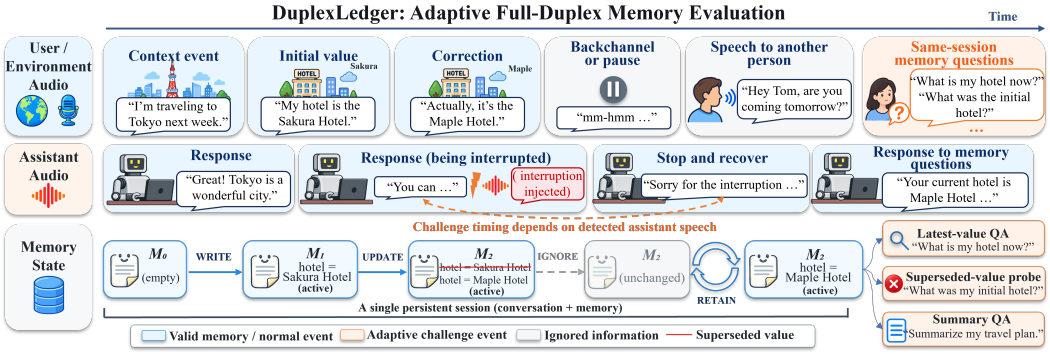


Figure 2: Overview of DuplexLedger. A persistent session aligns adaptively triggered full-duplex challenges, event-level memory transitions, and spoken memory questions.

WRITE introduces a value, UPDATE supersedes an active value, IGNORE rejects inadmissible evidence, and RETAIN preserves the current state. The evaluated model receives only streaming audio and spoken questions; ledger operations and states are reserved for supervision and scoring. The task therefore evaluates selective state construction rather than unrestricted recall.

3.2 Adaptive Full-Duplex Sessions

Each persistent session contains contextual events, two to five challenges, and six evaluation targets, without state resets. Fixed event content follows a model-dependent schedule: interruptions and backchannels require VAD-detected decoded assistant speech, with unmet triggers remaining unrealized. Pause and speech-to-others events use predefined observation windows.

The four scenarios require distinct behavior. An interruption should stop the current response and admit the revised intent; a backchannel should neither terminate the response nor enter memory. A within-turn pause should not trigger premature generation or commitment, whereas speech addressed to another person should produce neither an assistant response nor a memory update.

3.3 Event-Level Supervision and Evidence-Grounded QA

Each ledger entry records the event content and roles, target operation, memory-state transition, and supersession links. Each question is aligned with supporting memory identifiers, forbidden superseded or distractor identifiers, and a reference answer. Two latest-value questions per session test whether revisions replace earlier values; the remaining targets cover factual recall, distractor rejection, final-state summarization, and scenario behavior.

Table 2: Composition and supervision statistics of DuplexLedger.

(a) Session composition

Scenario	ZH	EN	Sess.	Events	Chal.	QA
Pause handling	149	127	276	1,932	552	1,656
Speech to others	150	126	276	2,208	1,104	1,656
User backchannel	146	137	283	2,545	1,130	1,698
User interruption	132	127	259	1,554	518	1,554
Total	**577**	**517**	**1,094**	**8,239**	**3,304**	**6,564**

(b) Ledger and evaluation supervision

Operation	Count	%	Target	Count	/sess.
WRITE	3,982	27.1	Latest value	2,188	2
UPDATE	1,212	8.3	Factual recall	1,094	1
RETAIN	1,941	13.2	Distractor reject.	1,094	1
IGNORE	7,546	51.4	Final summary	1,094	1
Total	**14,681**	**100**	Behavior check	1,094	1

Notes. Events are executable user or bystander inputs; ledger counts additionally include 6,442 aligned assistant-shadow entries. Fixed-input and event-source audio total 35.33 and 24.22 h, respectively. Release dev/test splits contain 237/857 sessions; method train/validation/calibration/held-out splits contain 660/175/40/219 sessions.

Combined with system traces, this alignment supports diagnosis of admission, supersession, exclusion, retrieval, and generation failures. Table 2 summarizes dataset composition and supervision.

3.4 Construction and Evaluation Protocol

DuplexLedger comprises controlled bilingual sessions generated specifically for this benchmark. We preserve source waveforms, derive event playback windows from timestamps and forced-alignment boundaries, and compile immutable manifests specifying event order, response requirements, challenge conditions, and same-session questions. Gold operations, memory states, and reference answers are retained for supervision, scoring, and audit, but withheld from the evaluated speech model and its answer generator. Appendix A details construction, quality controls, and the data card.Adaptive evaluation distinguishes planned from realized challenges. Interruption and backchannel events are triggered by decoded assistant speech, whereas pause and speech-to-others conditions use scheduled observation windows. For session s, the realized challenge rate (RCR) and its macro-average are defined as

$$\mathrm{RCR}_s = \frac{N^{(s)}_{\text{realized}}}{N^{(s)}_{\text{planned}}}, \qquad \overline{\mathrm{RCR}} = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \mathrm{RCR}_s, \tag{2}$$

where $\mathcal{S}$ contains sessions with at least one planned challenge.

Other session-level ratios are also macro-averaged unless stated otherwise. Conditional interaction metrics use eligible realized or observable events and are reported alongside RCR and validity indicators. All memory answers are scored by the same deterministic bilingual lexical scorer. Memory questions retain fixed question denominators, so silence and unanswered cases receive no credit. Latest-value answers must assert the active value without asserting a superseded alternative, and silence does not count as distractor rejection. Exact scheduling rules and metric denominators are specified in Appendices B and C; the released artifact structure is documented in Appendix J.

4 Performance of Current Full-Duplex Models

4.1 Experimental Setup and Reporting Protocol

We evaluate five full-duplex systems using identical source audio, event order, questions, and scheduler logic. Since model architectures and response tendencies differ, cross-model results are descriptive. Memory questions retain fixed denominators, while conditional interaction metrics follow their eligible-event denominators. RCR is therefore treated as challenge exposure rather than a performance score. The execution and scoring protocols are detailed in Appendices B and C; model-specific inference settings and complete baseline results appear in Appendices D and H.

4.2 Benchmark Results

[Takeaway 1] No model jointly dominates full-duplex interaction and memory fidelity. Table 3 separates the two capabilities. `Lychee-FD` leads in conditional full-duplex performance and latest-value accuracy. `MiniCPM-o 4.5` achieves the lowest early-onset and pre-end overlap rates, together

Table 3: Full-duplex interaction and memory fidelity on DuplexLedger.

(a) Full-duplex interaction

Model	RCR†	cFD↑	Early↓	Pre-end↓	Win-Intr.↓	Stop@0.8↑	Stop (s)↓
MiniCPM-o 4.5	73.51	40.23	**18.70**	**44.38**	59.47	55.69	1.032
Freeze-Omni	86.17	28.40	49.76	91.82	73.64	45.75	1.239
Moshi	44.63	37.36	66.59	72.25	**45.11**	60.79	0.777
Lychee-FD	75.64	**47.32**	24.68	57.14	53.89	**90.12**	**0.312**
PersonaPlex	93.54	22.74	46.64	72.73	90.62	16.40	1.689

(b) Memory fidelity

Model	Resp.↑	Latest↑	Stale↓	D-Adopt↓	D-Rej.↑	Final↑
MiniCPM-o 4.5	98.90	4.30	0.59	**14.17**	**85.19**	**2.01**
Freeze-Omni	**99.94**	11.61	3.38	65.90	34.10	0.00
Moshi	54.31	2.19	**0.32**	44.15	8.32	0.04
Lychee-FD	92.41	**25.50**	1.65	48.17	51.55	1.06
PersonaPlex	96.40	5.85	2.29	79.71	17.46	0.24

Bold and underlined: best and second best. Values are percentages except Stop (s). † RCR is diagnostic; cFD is conditional. Win-Intr.: windowed speech intrusion; D-Adopt/D-Rej.: distractor adoption/rejection; Final: final-state coverage.

with the strongest distractor rejection and final-state coverage. Moshi pairs the lowest stale rate with only 54.31% response coverage. PersonaPlex realizes the most challenges but obtains the lowest cFD. These contrasting profiles preclude a meaningful scalar ranking.

4.3 Full-Duplex Coordination under Adaptive Challenges

[Takeaway 2] Challenge exposure, boundary control, and interruption recovery are distinct capabilities. Lychee-FD combines substantial exposure with the highest cFD, 90.12% stopping success within 800 ms, and the shortest stopping latency. Its remaining pre-end overlap and window intrusion nevertheless show that fast interruption recovery does not ensure correct turn boundaries. MiniCPM-o 4.5 instead minimizes early onset and pre-end overlap, while PersonaPlex pairs the highest RCR with the lowest cFD, greatest window intrusion, and slowest stopping. RCR must therefore be interpreted as exposure rather than interaction quality.

4.4 Transactional Memory Failures and Interaction–Memory Misalignment

[Takeaway 3] Local value revision does not yield a coherent persistent state. Lychee-FD reaches 25.50% latest-value accuracy but covers only 1.06% of final-state facts and adopts 48.17% of distractors. MiniCPM-o 4.5 is more selective but correctly answers only 4.30% of update questions. Freeze-Omni responds almost universally but retains high stale-value and distractor errors. Moshi similarly combines a low stale rate with limited response coverage and latest-value accuracy.

Even the best final-state coverage is only 2.01%. Current systems can coordinate speech, recover isolated revisions, or reject irrelevant content, yet none jointly supports event admission, supersession, exclusion, and state consolidation. Maintaining an evolving state during full-duplex interaction is therefore the central limitation motivating ACT-Mem in Section 5.

5 ACT-Mem: Transactional Memory for Full-Duplex Agents

5.1 Design Motivation and Overview

The failures in Section 4.4 motivate explicit control over memory admission and revision. We introduce ACT-Mem (**A**uditable and **C**ausal **T**ransactional Memory), an external module coupled with a frozen MiniCPM-o 4.5 backbone (Cui et al., 2026). As shown in Figure 3, streaming audio enters both the frozen full-duplex backbone and a frozen ASR/VAD module. The latter supplies transcripts and timing to a trainable event controller, which manages a transactional ledger implemented as versioned memory. A typed retriever constructs evidence cards for memory questions. Only these cards enter the backbone through QA-only text prefill; all speech-generation components remain frozen. Architectural and runtime details are provided in Appendix E.

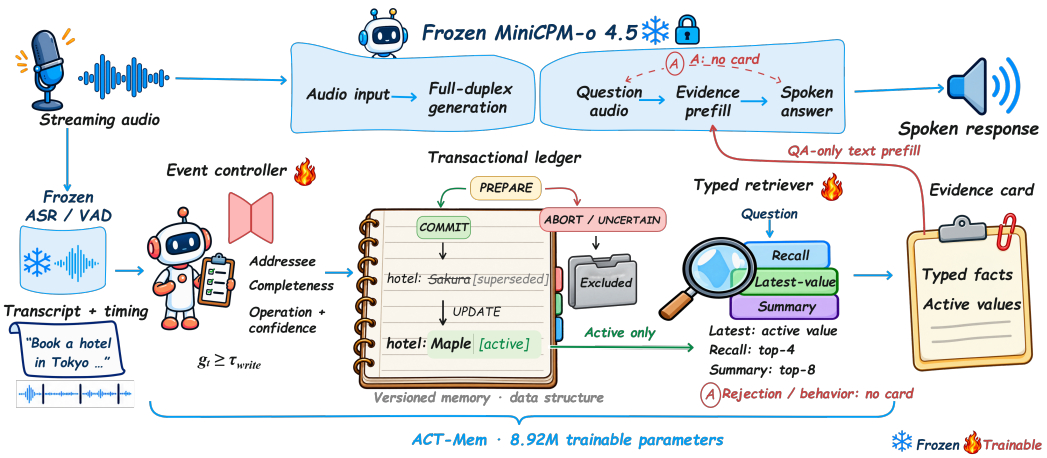


Figure 3: Overview of the ACT-Mem framework for transactional memory in full-duplex agents.

5.2 Event Admission and Transaction Lifecycle

[Admission] **Persistence requires explicit evidence admission.** A deterministic parser extracts candidate entity–value slots from the event transcript, while the learned controller predicts event-level addressee, completeness, operation, and confidence for admission. Each proposal enters PREPARE with score

$$g_t = p_t^{\text{addr}} p_t^{\text{complete}} \big[p_t(\text{WRITE}) + p_t(\text{UPDATE})\big] c_t, \tag{3}$$

where p_t^{addr} and p_t^{complete} estimate assistant-directedness and semantic completeness, and $c_t \in [0, 1]$ is the controller confidence.

A proposal enters COMMIT when $g_t \geq \tau_{\text{write}}$. Non-addressed or incomplete proposals enter ABORT; ambiguous or low-confidence proposals remain UNCERTAIN. Neither is retrievable. Pause and silence create no persistent transaction. WRITE creates an active fact, UPDATE replaces an active value, IGNORE rejects the proposal, and RETAIN preserves the ledger. Committed entries retain their source event, predicted roles, order, completeness, and confidence for audit.

5.3 Causal Supersession and Versioned Memory

[Supersession] **Recency alone cannot overwrite an active value.** The transactional ledger extends the state in Section 3.1 with provenance and version links. A candidate m_j may supersede an active item m_i only when

$$k_j = k_i, \qquad t_j > t_i, \qquad \widehat{o}_j = \text{UPDATE}, \qquad g_j \geq \tau_{\text{write}}, \tag{4}$$

where k_i is the entity key and t_i denotes source-event order. A later same-key mention is therefore insufficient without an admissible update.

A successful transaction marks m_i as *superseded*, activates m_j, and records their correction link. Both versions remain auditable, but only active values are exposed to the typed retriever. Superseded, uncertain, aborted, and ignored content cannot enter an evidence card.

5.4 Typed Retrieval and Memory Conditioning

[Typed retrieval] **Question type controls evidence scope and card generation.** The typed retriever maps each question to one of four behaviors. A *latest-value* question selects the target entity's active value, a *recall* question retrieves up to four active facts, and a *summary* question aggregates up to eight facts across types and entities. Rejection and behavior questions produce no card, preventing inadmissible content from being disclosed to the backbone.

For retrieval-enabled questions, ranking combines dense similarity (Karpukhin et al., 2020), lexical overlap, recency, controller confidence, and stale/distractor penalties. The resulting card contains only typed facts and active values; insufficient evidence yields no card.

ACT-Mem conditions generation without modifying the decoder (Lewis et al., 2020). At a stable-silence QA boundary, the backbone processes the question audio, receives one evidence card through QA-only text prefill, and generates a spoken answer. Context events and full-duplex challenges continue through the native audio interface without memory prefill.

5.5 Training, Calibration, and Model Complexity

Training uses frozen-ASR transcripts from the session-disjoint training split. The event controller learns operation, addressee, completeness, and confidence prediction. Slot extraction and entity-key normalization use non-trainable rules, and the ledger applies version transitions. Its objective is

$$\mathcal{L}_{\text{ctrl}} = \mathcal{L}_{\text{op}}^{\text{wCE}} + 0.3\mathcal{L}_{\text{addr}}^{\text{CE}} + 0.3\mathcal{L}_{\text{complete}}^{\text{CE}} + 0.2\mathcal{L}_{\text{conf}}^{\text{BCE}},$$
$$\mathcal{L}_{\text{retr}} = \mathcal{L}_{\text{BCE}}(\mathbf{s}, \mathbf{y}) + \frac{0.5}{|\mathcal{P}||\mathcal{N}|} \sum_{p \in \mathcal{P}} \sum_{n \in \mathcal{N}} [0.25 - s_p + s_n]_+ . \tag{5}$$

Weighted operation cross-entropy uses class weights $(1.0, 1.5, 1.4, 0.8)$ for WRITE, UPDATE, IGNORE, and RETAIN. $\mathbf{s}$ contains retrieval scores, $\mathbf{y}$ gives binary relevance labels, and $\mathcal{P}$ and $\mathcal{N}$ contain positive and forbidden, hard-negative, or distractor items in the QA context. Joint training updates the controller and retriever; adaptation freezes the controller and updates only the retriever.

We use the session-disjoint splits in Section 3.4. Gold states and evidence labels provide supervision only; future events never enter the current prefix. Calibration on 40 sessions gives $\tau_{\text{write}} = 0.35$ without gradient updates. ACT-Mem adds 8.92M trainable parameters: 4.59M in the event controller and 4.33M in the typed retriever. The backbone and ASR/VAD module remain frozen. Training uses 8 NVIDIA A100 GPUs; optimization and augmentation details are provided in Appendix F.

6 Evaluation of ACT-Mem

We evaluate ACT-Mem on the same 219 held-out sessions used for the baseline study, containing 1,314 questions/checks and 438 latest-value questions. The controlled comparison retains the frozen `MiniCPM`-o 4.5 backbone; no held-out session is used for training, checkpoint selection, or calibration. Evaluation follows Sections 3.4 and 4.1. Relative changes use unrounded scores, and pointwise 95% confidence intervals for latest-value accuracy and final-state coverage are obtained from 20,000 paired bootstrap resamples of sessions. Further details appear in Appendices C and G.

6.1 Main Results

Figure 4 compares the complete ACT-Mem configuration with its frozen `MiniCPM`-o 4.5 backbone.

[Takeaway 1]* ACT-Mem *substantially improves latest-value recovery, while complete final-state reconstruction remains difficult. Latest-value accuracy increases from 4.11% to 8.68% (+111.2%; +4.57 percentage points, 95% CI $[1.60, 7.53]$), whereas final-state coverage rises from 1.27% to 2.00% (+57.5%; +0.73 points, 95% CI $[-0.49, 1.99]$). The latest-value interval excludes zero, providing evidence that ACT-Mem more reliably recovers active committed values after revision. Final-state coverage, however, remains low and its interval crosses zero. The controlled evidence therefore supports improved local state revision, but not yet complete reconstruction of the session-wide memory state.

[Takeaway 2]* *The memory gains are concentrated in selective commitment rather than uniformly improving every outcome. Stale assertions and distractor adoption decrease by 0.22 and 2.28 percentage points, respectively, showing that the latest-value gain is accompanied by fewer conflicting and incorrectly admitted facts. Distractor rejection and response rate decline by 2.28 and 1.45 points, although the latter remains high at 97.41%. Because distractor adoption and rejection are separately credited under fixed denominators, they are not complementary; their joint decline means that ACT-Mem produces fewer explicit adoptions but also fewer credited rejections. The principal benefit is therefore more selective state construction, while robust distractor rejection remains improvable.

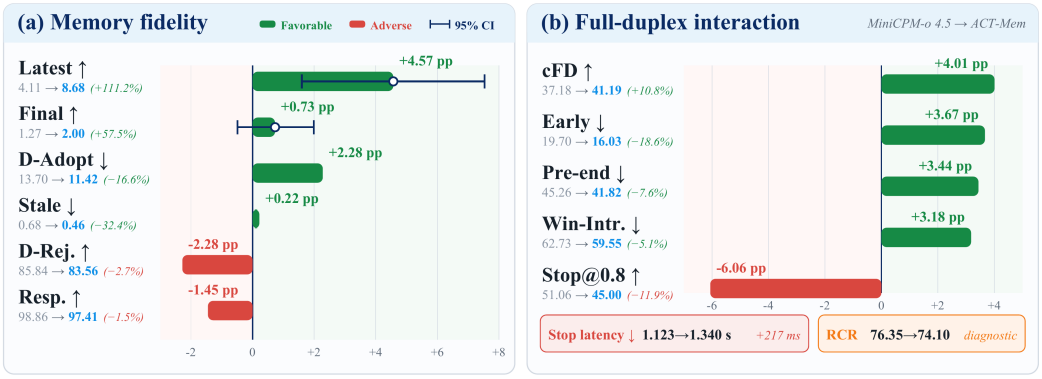


Figure 4: ACT-Mem versus its frozen MiniCPM-o 4.5 backbone on 219 held-out sessions; error bars show 95% paired-bootstrap CIs for Latest and Final.

Table 4: Ablations on 219 sessions. Best and second-best scores are bold and underlined.

	Full-duplex interaction			Memory fidelity			
Variant	**cFD↑**	**Early↓**	**Pre-end↓**	**Latest↑**	**Stale↓**	**D-Adopt↓**	**Final↑**
FIFO memory	34.63	37.21	48.65	7.08	2.28	<u>12.33</u>	1.61
w/o addressee gating	40.14	31.48	46.36	6.39	<u>0.91</u>	17.35	1.75
Untyped retrieval	<u>40.33</u>	<u>29.98</u>	<u>44.61</u>	**9.59**	1.60	19.63	<u>1.82</u>
ACT-Mem	**41.19**	**16.03**	**41.82**	<u>8.68</u>	**0.46**	**11.42**	**2.00**

All values are percentages.

6.2 Effects on Full-Duplex Interaction

[Takeaway 3]* ACT-Mem *improves most interaction-boundary metrics but slows interruption stopping. cFD increases by 10.8%, while early response, pre-end speech, and windowed intrusion decrease by 18.6%, 7.6%, and 5.1%, respectively. Because cFD is conditioned on realized challenges and RCR decreases from 76.35% to 74.10%, this improvement should be interpreted as better behavior on exposed challenges rather than uniformly greater challenge coverage. The main trade-off is interruption stopping: Stop@0.8 falls from 51.06% to 45.00%, and median stopping latency increases from 1.123 to 1.340 seconds. Thus, ACT-Mem improves observed onset and overlap control, while rapid interruption recovery remains its principal interaction weakness.

6.3 Ablation Studies

We evaluate three inference-time variants with fixed memory-model weights. *FIFO memory* replaces the versioned ledger with a 32-slot queue and removes supersession and latest-only selection. *Without addressee gating* removes speaker/addressee rejection and the learned gate. *Untyped retrieval* applies top-4 retrieval to every question type, including rejection questions.

***[Takeaway 4]* The full configuration balances value recovery with selective memory use.** ACT-Mem leads all three reported interaction metrics and three of four memory metrics. Compared with FIFO memory, its stale-assertion rate is 79.8% lower; compared with removing addressee gating, distractor adoption is 34.2% lower. Untyped retrieval achieves 9.59% latest-value accuracy, but stale assertions and distractor adoption reach 1.60% and 19.63%. ACT-Mem thus trades 0.91 percentage points of latest-value accuracy for fewer stale assertions and distractor adoptions. Interaction differences remain descriptive across runs: QA-only retrieval does not establish a causal effect on preceding interaction boundaries.

6.4 Results by Language and Scenario

***[Takeaway 5]* Improvements are bilingual but scenario-dependent.** Latest-value accuracy increases from 5.83% to 13.11% on English sessions and from 2.59% to 4.74% on Chinese sessions. The largest absolute gains occur for backchannels (+10.53 points) and speech to others (+6.36), followed by pause handling (+1.82). Under user interruption, however, latest-value accuracy decreases from 3.85% to 2.88%; backchannel final-state coverage also falls from 2.34% to 1.10%. These descriptive breakdowns identify interruption-time updating as the principal remaining weakness.



Figure 5: ACT-Mem cases across four full-duplex scenarios, showing controller decisions, memory transitions, and paired outcomes.

6.5 Case Study

Figure 5 connects interaction events with admission decisions, committed states, and QA outcomes. The controller governs admission; the ledger records supersession and exclusion; typed retrieval supplies active, question-relevant evidence. DuplexLedger's reference transitions and ACT-Mem's transaction traces distinguish correct operation predictions from successful slot resolution, supersession, or exclusion (Appendix I, Table 21). These records support failure localization beyond answer scores by separating predicted operations, committed changes, and downstream QA outcomes.

6.6 Admission-Threshold Sensitivity

Table 5 summarizes admission sensitivity on 40 calibration sessions with fixed model weights. Event-level admission precision remains 100% at all tested thresholds; 0.35 and 0.45 share the highest F1. Increasing the threshold from 0.35 to 0.75 reduces recall from 55.26% to 48.95% and F1 from 71.19% to 65.72%. These results indicate local stability across two adjacent settings and a recall cost from stricter admission, with conclusions limited to calibration-set admission behavior.

Table 5: Threshold sensitivity on 40 calibration sessions.

τ_{write}	0.35	0.45	0.50	0.55	0.65	0.75
Recall (%)	55.26	55.26	54.74	53.68	50.53	48.95
F1 (%)	71.19	71.19	70.75	69.86	67.13	65.72

7 Conclusion

We frame persistent memory during concurrent speech as selective state management. DuplexLedger aligns adaptive challenges, event-level operations, explicit reference states, and spoken QA across 1,094 bilingual sessions. This alignment supports joint evaluation of interaction and memory, while reference transitions and system traces help diagnose admission, supersession, exclusion, retrieval, and generation failures. Across five systems, no model dominates both interaction and memory fidelity. ACT-Mem combines an event controller, versioned ledger, and typed retriever with a frozen backbone. On 219 held-out sessions, latest-value accuracy rises from 4.11% to 8.68%, with fewer stale assertions and distractor adoptions. Several interaction-boundary metrics also improve, although interruption stopping becomes slower. These findings motivate explicit, auditable state management and further validation on natural conversations, additional languages, longer timescales, and other backbones.

Ethics Statement

DuplexLedger contains AI-generated dialogues and synthetic speech rather than recordings of human participants or private conversations. Its controlled scenarios may nevertheless encode generation-model biases and should not be treated as representative of unrestricted real-world dialogue.

Reproducibility Statement

Dataset construction, execution protocols, metric denominators, baseline configurations, ACT-Mem implementation and training, statistical procedures, complete baseline results, qualitative traces, and artifact organization are documented in Appendices A–J.

AI Use Statement

Generative models were used to construct synthetic benchmark dialogues as documented in Appendix A. The authors specified the schemas, validation rules, evaluation protocol, and analyses, and remain responsible for the resulting content and claims.

References

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen2-Audio technical report, 2024. URL https://arxiv.org/abs/2407.10759.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Jiancheng Gui, Luoyuan Zhang, Xian Sun, Fuwei Huang, Moye Chen, Zhuo Lin, Hanyu Liu, Qingxin Gui, Qingzhe Han, Yuyang Wen, Huiping Liu, Rongkang Wang, Yaqi Zhang, Hongliang Wei, Chi Chen, You Li, Kechen Fang, Jie Zhou, Yuxuan Li, Guoyang Zeng, 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, 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.

Advait Gosai, Tyler Vuong, Utkarsh Tyagi, Steven Li, Wenjia You, Miheer Bavare, Arda Uçar, Zhongwang Fang, Brian Jang, Bing Liu, and Yunzhong He. Audio MultiChallenge: A multi-turn evaluation of spoken dialogue systems on natural human interaction. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35740–35770. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1654. URL https://aclanthology.org/2026.acl-long.1654/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract.html.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

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 *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2025. URL https://arxiv.org/abs/2503.04721.

Guan-Ting Lin, Chen Chen, Zhehuai Chen, and Hung-yi Lee. Full-duplex-bench v3: Evaluating tool use under speech disfluency, 2026a. URL https://arxiv.org/abs/2604.04847.

Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, and Hung-yi Lee. Full-duplex-bench v1.5: Evaluating overlap handling for full-duplex spoken dialogue models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2026b. URL https://arxiv.org/abs/2507.23159.

Guan-Ting Lin et al. Full-duplex-bench v2: An adaptive benchmark for full-duplex spoken dialogue models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026c. URL https://arxiv.org/abs/2510.07838.

Zhenyu Liu, Xuanyu Zhang, Yunxin Li, Qixun Teng, Shenyuan Jiang, Haolan Chen, Mingjun Zhao, Fanbo Meng, Yu Xu, Yancheng He, Baotian Hu, Haizhou Li, and Min Zhang. Hierarchical acoustic-semantic modeling: Modality separation and semantic coherence for full-duplex SLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9264–9280. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.419. URL https://aclanthology.org/2026.acl-long.419/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Smit Nautambhai Modi, Gandharv Mahajan, Marc Wetter, and Randall Welles. Echochain: Evaluating state updates under interruption in full-duplex spoken dialogue, 2026. URL https://arxiv.org/abs/2604.16456.

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/.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. URL https://arxiv.org/abs/2310.08560.

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, 2026. URL https://arxiv.org/abs/2602.06053.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. SALMONN: Towards generic hearing abilities for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=t8EF6AAImY.

Bandhav Veluri, Benjamin N. Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. Beyond turn-based interfaces: Synchronous LLMs as full-duplex dialogue agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21390–21402. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1192. URL https://aclanthology.org/2024.emnlp-main.1192/.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy F. Chen. AudioBench: A universal benchmark for audio large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4297–4316. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.naacl-long.218. URL https://aclanthology.org/2025.naacl-long.218/.

Chengyou Wang, Hongfei Xue, Guojian Li, Zhixian Zhao, Shuiyuan Wang, Shuai Wang, Xin Xu, Hui Bu, and Lei Xie. HumDial-FDBench: A human-recorded benchmark for full-duplex spoken dialogue, 2026. URL https://arxiv.org/abs/2604.21406.

Qichao Wang, Ziqiao Meng, Wenqian Cui, Yifei Zhang, Pengcheng Wu, Bingzhe Wu, Irwin King, Liang Chen, and Peilin Zhao. NTPP: Generative speech language modeling for dual-channel spoken dialogue via next-token-pair prediction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63912–63928. PMLR, 2025b. URL https://proceedings.mlr.press/v267/wang25by.html.

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, 2025c. URL https://proceedings.mlr.press/v267/wang25aw.html.

Di Wu et al. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Zhifei Xie and Changqiao Wu. Mini-Omni: Language models can hear, talk while thinking in streaming, 2024. URL https://arxiv.org/abs/2408.16725.

Zhifei Xie et al. Voicemem: Streaming memory for long-term voice interaction, 2026. URL https://arxiv.org/abs/2608.26005.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.12110.

Yiqun Yao et al. Egomem: Lifelong memory agent for full-duplex omnimodal models, 2025. URL https://arxiv.org/abs/2509.11914.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15757–15773. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.1055. URL https://aclanthology.org/2023.findings-emnlp.1055/.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. MemGen: Weaving generative latent memory for self-evolving agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vI56m4Iu4e.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

A DuplexLedger Construction and Data Card

A.1 Construction Principles

DuplexLedger was constructed to evaluate whether a spoken dialogue agent can maintain a coherent memory state while simultaneously respecting full-duplex interaction boundaries. We designed each session around three coupled requirements: (i) the dialogue must contain facts that evolve through additions and corrections; (ii) some audible content must be inadmissible to persistent memory; and (iii) the interaction must contain a concurrency event whose correct handling depends on the model's runtime speaking state. These requirements led to four scenarios: user interruption, user backchannel, within-turn pause, and speech addressed to another person.

All dialogue content is synthetic. We used GPT-5.5 to generate the underlying Chinese and English task descriptions, conversational utterances, factual values, corrections, distractors, and reference answers under scenario-specific instructions. Generation was constrained by a structured schema specifying speaker roles, event order, memory dependencies, challenge types, and evaluation targets. The generated content was subsequently normalized and instantiated by deterministic construction programs. The language model was used only during dataset creation and is not part of the benchmark execution or scoring pipeline.

A.2 Task and Dialogue Generation

We constructed sessions from a broad collection of everyday task templates, including scheduling, planning, customer support, troubleshooting, study assistance, household coordination, travel, document preparation, creative work, and low-pressure personal organization. The Chinese and English construction pools contain 73 and 72 base task templates, respectively, covering 72 high-level domains in each language.

Each base task specifies two revisable attributes and several persistent facts. A typical task contains:

- an initial value and a corrected value for a primary attribute;
- an initial value and a corrected value for a secondary attribute;
- a stable goal or constraint that should remain active;
- an additional task-specific detail;
- a named person and an associated responsibility;
- a plausible but irrelevant distractor;
- an expected consolidated final state.

To reduce exact repetition, each task was combined with a scenario-specific identifier, one of four discourse-style variants, and one of twelve instance-level modifiers. The modifiers vary properties such as budget conservatism, scheduling slack, communication requirements, fallback plans, execution complexity, and family or workplace coordination. Timing, challenge position, pause length, overlap duration, and short-feedback form were sampled using fixed construction seeds.

We also varied how the user concludes a session. A final turn may fully restate the task, omit the recap, provide only a partial recap, contain a self-correction, or explicitly request that the assistant not summarize. The intended proportions were 40%, 25%, 20%, 10%, and 5%, respectively. The released corpus contains 434 complete recaps, 268 sessions without a complete final recap, 218 incomplete recaps, 118 recaps containing a correction, and 56 explicit no-summary requests. This variation prevents the final memory state from being recoverable solely by copying the last user turn.

A.3 Scenario-Specific Interaction Design

The same underlying memory requirements are combined with four distinct full-duplex phenomena.

User interruption. The user begins with a provisional task state and later interrupts an active assistant response to correct one or more values. The interruption should terminate the obsolete response trajectory, and the corrected values should supersede their earlier versions. A second interruption may clarify that a nearby detail is irrelevant and must not enter persistent memory.

User backchannel. Short acknowledgements such as "yes," "okay," or their Chinese equivalents are inserted while the assistant is speaking. These utterances provide listener feedback but introduce no new task fact. A correct system should continue its response without interpreting the backchannel as an interruption or memory write.

Pause handling. The user produces an incomplete utterance, pauses, and then resumes with a completion or correction. The pause is represented explicitly rather than as a completed turn. A correct system should wait for the continuation and must not commit a provisional value merely because a long silence occurred.

Speech addressed to another person. The session contains both user speech directed to a bystander and audible speech from that bystander. Such content is topically plausible but is not addressed to the assistant. It therefore serves as a controlled distractor: the model should neither respond to it nor incorporate it into the official task state. The user later returns explicitly to the assistant and continues the admissible task.

A.4 Structured Event and Memory Annotation

A session is represented as an ordered list of speech and silence events. Each event records its identifier, speaker, addressee, dialogue act, planned time interval, audio-rendering status, memory dependencies, and candidate memory writes. Events spoken by the primary user or bystander are audible; reference assistant events specify the intended conversational trajectory but are not rendered into the benchmark input.

Memory is annotated at the item level. The annotation vocabulary covers `goal`, `schedule`, `constraint`, `preference`, `episodic`, `named_entity`, `task_state`, and `correction`. Each item contains a stable memory identifier, semantic type, textual content, and source event. Corrections additionally identify the earlier item that they supersede. Distractors are annotated separately and are excluded from the oracle final state.

For evaluation, the event annotations are mapped to four memory operations: WRITE, UPDATE, IGNORE, and RETAIN. Ordinary admissible facts produce WRITE; explicit corrections and clarifications produce UPDATE; backchannels, side conversations, and bystander speech produce IGNORE; and pauses without confirmed content produce RETAIN. This mapping supplies the event-level supervision used to distinguish admission errors from supersession, retrieval, and answer-generation errors.

The construction procedure enforces two provenance constraints. First, every item in the oracle final state must be supported by an audible user event. Muted reference-assistant events cannot introduce a gold fact. Second, a superseding item must refer to an earlier item in the same session; future or cross-session references are disallowed.

A.5 Memory Questions

Every session contains six spoken evaluation targets:

1. two latest-value questions, one for each corrected attribute;
2. one factual-recall question about a stable requirement;
3. one distractor-rejection question;
4. one final-state summary question;

5. one scenario-specific behavioral check.

Each question is aligned with its expected answer, supporting memory identifiers or events, and, when applicable, forbidden obsolete or distractor items. The latest-value questions require the corrected value and prohibit credit when the obsolete value is asserted. The distractor question tests whether the system rejects inadmissible content rather than rewarding silence alone. The summary question requires coverage of multiple active facts and therefore evaluates state consolidation beyond isolated retrieval.

Questions are synthesized as audio and presented after all contextual and challenge events, without resetting the model state. The question audio, reference text, and gold evidence annotations are stored separately. Reference answers and memory annotations are used only for scoring and are not exposed to the evaluated system.

A.6 Speech Synthesis and Temporal Alignment

All speech is generated using text-to-speech synthesis; the dataset contains no recordings from human participants. Audible event segments are rendered at 24 kHz and assembled according to the structured event timeline. Scenario-specific speaker assignments distinguish the primary user from the bystander in multi-speaker sessions. The reference assistant remains muted, because the assistant audio during evaluation must be produced by the system under test.

We apply forced alignment to the synthesized speech to obtain word-level English boundaries and character-level Chinese boundaries. Punctuation is removed for alignment, while the original text is retained in the source annotation. The aligned event window is defined by the first and last aligned units with 120 ms of boundary padding. For each event, we retain the source sampling rate, number of samples, first and last aligned boundaries, leading and trailing silence, playback sample range, source duration, SHA-256 checksum, and quality-control flags.

The fixed mixed waveform is retained as a compatibility artifact, but it is not the authoritative input for adaptive evaluation. The final protocol instead replays aligned event segments according to a closed-loop stage manifest. This distinction is necessary because a full-duplex challenge should occur when the evaluated model is actually speaking, rather than at a timestamp derived from a prerecorded reference trajectory.

A.7 Adaptive Session Materialization

For each session, we compile a chronological execution manifest containing ordinary context playback, adaptive challenges, recovery periods, and same-session memory QA. Ordinary user events are streamed in real time. Before an interruption or backchannel, the scheduler waits for decoded-audio VAD to confirm that the evaluated model is speaking. It then injects the corresponding user event. If the required speaking state is not observed within the timeout, the challenge remains unrealized and the timeout is recorded.

Pause intervals are represented as explicit wait stages. Speech directed to another person is played through a dedicated side-talk stage and is marked as not addressed to the assistant. After an ordinary response-required event, the scheduler observes response onset and subsequent silence before advancing. The six audio questions are asked only after the interaction stages have finished.

The complete session uses one persistent model state:

- the model is reset before a new session;
- it is not reset between context and challenges;
- it is not reset between challenges and memory QA;
- it is not reset between individual QA items.

The execution protocol uses 80-ms input chunks. Interruption and backchannel triggers require at least 500 ms and 800 ms of active assistant speech, respectively. Question input retains 400 ms of post-speech tail silence, and answer windows are determined dynamically. Runtime failures, trigger timeouts, and missing responses are recorded rather than silently discarded.

Table 6: Composition of DuplexLedger. Fixed-input duration refers to the compatibility waveform; event-source duration is the total duration of the individual audible speech segments used by the adaptive protocol.

Quantity	Value
Sessions	1,094
Chinese / English	577 / 517
Interruption / backchannel sessions	259 / 283
Pause / talking-to-other sessions	276 / 276
Development / test sessions	237 / 857
Memory questions	6,564
Latest-value questions	2,188
Full-duplex challenges	3,304
Audible speech and silence events	8,239
Speech-event waveforms	7,687
Fixed compatibility audio	35.328 h
Event-source audio	24.221 h
Session duration, mean / median	116.25 s / 114.93 s
Session duration, minimum / maximum	86.64 s / 176.70 s
MFA-aligned speech events	7,589
Missing event waveforms	0

A.8 Validation and Quality Control

We apply validation at three levels.

Structural validation. Before synthesis, each session must contain all required fields, unique event and memory identifiers, chronologically valid timestamps, valid memory dependencies, and consistent overlap annotations. Memory reads may reference only previously introduced items. Supersession links must point to existing earlier items, and every final-state fact must trace to audible user speech.

Audio and alignment validation. The construction pipeline verifies the existence and duration of every waveform, preserves source-file checksums, and records missing alignments, leading silence, and trailing residuals. The released corpus contains no missing event-audio files. Of 7,687 speech events, 7,589 have forced-alignment boundaries. The remaining events are retained with explicit missing-alignment flags rather than removed. In addition, 3,007 aligned events contain more than 0.5 s of trailing residual audio; these cases remain visible through per-event quality-control metadata.

Execution validation. Each session must contain a nonempty ordered stage list, unique stage identifiers, relative and resolvable audio paths, an adaptive state-wait stage before every interruption or backchannel injection, and a final same-session QA stage. Question windows must satisfy

$$0 \leq n_{\text{start}} < n_{\text{speech-end}} \leq n_{\text{end}}.$$

A full validation pass over all 1,094 sessions reports no hard structural errors.

A.9 Dataset Composition

Table 6 summarizes the released corpus.

The public release uses a fixed split of 237 development and 857 test sessions. The split assignment, source text, source waveforms, question order, and oracle annotations are held fixed in the adaptive release. The separate method experiments use session-disjoint training, validation, calibration, and held-out subsets; no persistent session is shared across those subsets.

A.10 Intended Use and Access Restrictions

The baseline setting of DuplexLedger evaluates spoken agents through streamed audio events and spoken questions within a persistent session. Source text, annotation timestamps, event roles, gold

memory operations, oracle states, evidence links, and reference answers are evaluator-side resources rather than baseline-model inputs. For augmented systems, access by auxiliary modules must be distinguished from access by the speech backbone; ACT-Mem's observation and conditioning interfaces are described in Appendix E.

Access to gold transcripts, annotated speaker/addressee labels, question types, oracle states, or supporting-memory identifiers must be declared for the complete evaluated system, including auxiliary modules. Such access constitutes an annotation-assisted or oracle setting, depending on the information supplied, and must be distinguished from end-to-end audio-only evaluation.

A.11 Limitations and Responsible Release

DuplexLedger is fully synthetic and therefore does not reproduce the complete distribution of spontaneous human speech. In particular, TTS speech underrepresents natural disfluency, accent variation, emotional prosody, microphone conditions, environmental noise, and unplanned multi-party interaction. The sessions last approximately one to three minutes and do not establish performance on hour-scale or cross-day memory.

Although the task includes selective admission, correction, and distractor rejection, it does not currently contain an explicit memory-deletion operation. It should therefore not be interpreted as a complete benchmark for privacy-aware memory management. The absence of real-user recordings reduces direct privacy risk, but public release still requires documentation of the generation-model and speech-synthesis licenses, the exact generation prompts, construction seeds, source and derived checksums, and all validation reports.

B Adaptive Full-Duplex Execution Protocol

B.1 Protocol Objective and Execution Unit

The evaluation unit is a complete interaction session associated with one DuplexLedger sample. A session contains an annotated context, one or more full-duplex challenges, a recovery phase, and a set of memory questions. The audio content and event annotations are fixed before evaluation, whereas the wall-clock time at which a challenge is delivered is determined online from the decoded assistant waveform. Consequently, the protocol distinguishes between a *planned* challenge, a *trigger-satisfied* challenge, and a *realized* challenge. A planned challenge is an event specified in the sample manifest. A trigger is satisfied only when the required assistant state is observed within the prescribed timeout. A challenge is realized only when the corresponding user audio is subsequently delivered to the model. This separation prevents a silent or non-responsive model from being credited with having successfully experienced an interruption that was never actually applied.

Each sample is evaluated in one persistent model session. The session is reset before the first context event, but it is not reset between context, challenge, recovery, or memory-question stages. In particular, the memory questions are asked after the interaction challenges in the same conversational state. The default lifecycle is therefore

$$\textsc{Reset} \rightarrow \textsc{Context} \rightarrow \textsc{Challenge/Recovery} \rightarrow \textsc{SameSessionQA} \rightarrow \textsc{Finalize}.$$

No reset is performed between individual memory questions.

B.2 Compilation of Ledger Events into Execution Stages

The event-level representation is compiled into an ordered stage manifest before model inference. The manifest contains event identifiers, source sample ranges, speaker and addressee labels, dialogue acts, semantic turn boundaries, response requirements, and the expected memory operation. The scheduler executes only this manifest; it does not infer the event ordering from a concatenated waveform at runtime.

For ordinary response-required context events, the manifest inserts a response-start wait followed by a response-silence wait before the next context event. Adaptive interruption and backchannel events instead insert a state-trigger wait immediately before the challenge audio. This ordering ensures that the challenge is aligned to the model's actual speaking state rather than to a fixed timestamp in the source recording.

Table 7: Execution stages generated from annotated ledger events.

Stage	Purpose	Execution semantics
play_audio	Context delivery	Plays the bounded audio window associated with a user event and preserves the event's speaker, addressee, and dialogue-act metadata.
wait_for_assistant_state	State-triggered synchronization	Waits for decoded assistant speech to satisfy a target state and a minimum active-speech duration. The stage terminates on satisfaction or timeout.
inject_interruption	Interruption challenge	Delivers the interruption audio only after the preceding speaking-state trigger has been satisfied.
inject_backchannel	Backchannel challenge	Delivers a backchannel only after the assistant has produced the required amount of active speech.
play_side_talk	Non-addressee speech	Plays speech addressed to another participant and then observes whether the assistant responds during the observation interval.
wait	Annotated pause or observation	Advances the scheduler for a fixed duration without fabricating user speech. A pause inside a turn is not treated as a semantic turn boundary.
end_user_event	Event boundary	Records the semantic turn boundary and whether an assistant response is required; it does not reset the session.
wait_for_assistant_silence	Recovery synchronization	Waits for stable assistant silence after a response-required event.
run_same_session_memory_qa	Memory evaluation	Opens the dynamically timed question-and-answer windows without clearing the preceding conversational state.

B.3 Audio Windows and Clock Semantics

For each speech event, the scheduler stores an actual source range $[s_i, e_i)$ in samples. The range is derived from the annotated speech window and expanded by a fixed padding of $120\,\mathrm{ms}$, subject to the available source boundaries. The original waveform is never modified in place. A concatenated waveform may be retained as a compatibility artifact, but it is not the authority for adaptive execution; the stage manifest and its sample ranges are authoritative.

The protocol maintains two clocks. *Media time* measures the amount of user and assistant audio advanced by the scheduler, whereas *wall time* is measured with a monotonic runtime clock. Adaptive sessions are therefore allowed to have different wall-clock durations for different models. Runtime records preserve both clocks, the decoded assistant VAD state, and the timestamps reported by the native audio bridge. Decisions about whether the assistant is speaking are made from decoded PCM audio rather than packet arrival. Input audio is processed in $80\,\mathrm{ms}$ chunks, and the scheduler polls state at $20\,\mathrm{ms}$ intervals.

B.4 Adaptive Challenge Triggering

Interruption and backchannel challenges use a closed-loop trigger:

$$\text{wait for assistant speech} \;\rightarrow\; \text{verify VAD duration} \;\rightarrow\; \text{inject challenge audio.}$$

An interruption requires at least $500\,\mathrm{ms}$ of continuous decoded assistant speech, while a backchannel requires at least $800\,\mathrm{ms}$. Both trigger waits have a $10\,\mathrm{s}$ assistant-start timeout. If the condition is not satisfied before the timeout, the challenge audio is not sent and the event is recorded as an unrealized challenge with a trigger timeout. In particular, codec activity without VAD-confirmed speech does not satisfy the trigger.

The remaining challenge types use different execution semantics. A pause is implemented as a scheduler wait over its annotated duration and does not force the assistant to speak. Speech ad-

dressed to another participant is played as audio and followed by a $1.2\,\mathrm{s}$ observation interval. Its structured addressee label is retained in the evaluator's event record; whether an auxiliary module also accesses this field is a separate input configuration described in Appendix E. Thus, these events test pause intrusion and target-speaker selectivity, respectively, rather than using the same speech-trigger rule as interruption and backchannel events.

After an event whose semantic turn has ended and which requires a response, the scheduler waits for at least $160\,\mathrm{ms}$ of assistant speech. It then waits for stable silence for up to $12\,\mathrm{s}$, with a $2\,\mathrm{s}$ silence criterion. Failure to observe response onset is recorded as a response-initiation failure; it is not converted into a zero-duration response.

B.5 Recovery and Stop-Latency Measurement

For an interruption, the assistant's decoded waveform is monitored from the time at which challenge audio is injected. The runtime record stores whether the assistant was speaking before injection, whether speech continued after injection, the first observed speech offset, and whether the response ended within the configured silence window. Stop latency is computed only when a valid assistant speech interval and a valid interruption injection are both available. If the assistant does not stop before the response timeout, the event remains in the record with an explicit timeout or failure flag.

Backchannels are not treated as interruptions: the protocol records whether the assistant continues, pauses, or restarts after the feedback audio. For non-addressee speech, the observation interval is retained even when the assistant remains silent, since silence is a valid selectivity outcome. All runtime failures are recorded under a "record-and-continue" policy whenever the session remains executable. A server crash, unsupported cancellation, or unrecoverable protocol violation terminates the sample and preserves the partial runtime artifacts.

B.6 Same-Session Memory Questions

The final stage evaluates memory without resetting the model. Before each question, the controller waits for $1.2\,\mathrm{s}$ of stable assistant silence, with a maximum pre-input drainage timeout of $60\,\mathrm{s}$. When supported by the model interface, an outstanding response is cancelled through the native cancellation mechanism without clearing conversational state; the controller then allows an additional $10\,\mathrm{s}$ recovery interval before sending the question.

For each question, the execution order is

stable-silence check $\rightarrow$ question playback $\rightarrow$ answer-onset wait $\rightarrow$ answer collection $\rightarrow$ silence/turn-end closure.

The answer-onset timeout and answer-collection timeout are both $30\,\mathrm{s}$. An answer is closed by a detected turn end, a $2\,\mathrm{s}$ silence interval, or the answer timeout. Late speech after a timeout is held in a $10\,\mathrm{s}$ quarantine interval and is not retroactively assigned to the expired question. If no answer is observed, the question is marked `no_response`; text from a previous question is never reused.

Memory questions and reference targets are fixed by the planned session. If a speech-triggered challenge is not delivered, these targets are not reconstructed from the realized playback history. Fixed-denominator memory scores therefore measure end-to-end task completion and can reflect both missed evidence delivery and subsequent memory or response errors. They do not isolate retention conditional on receiving the supporting evidence.

B.7 Validity Gates and Failure Accounting

The protocol reports validity at multiple levels. First, a session is *protocol-valid* only if its manifest, audio windows, runtime clocks, and required artifacts pass structural validation. Second, a valid session is *behaviorally observable* only if decoded assistant audio is available for the relevant decision windows. Third, each planned challenge is labeled as realized, unrealized, or timed out according

Table 8: Default timing and execution parameters.

Parameter	Value	Parameter	Value
Input chunk	80 ms	Scheduler poll	20 ms
Event padding	120 ms	Assistant-start timeout	10 s
Interruption trigger speech	500 ms	Backchannel trigger speech	800 ms
Minimum response speech	160 ms	Assistant-end timeout	12 s
Post-event silence	240 ms	Stable assistant silence	2 s
Side-talk observation	1.2 s	QA pre-input quiet	1.2 s
QA start timeout	30 s	QA answer timeout	30 s
QA end silence	2 s	Late-response quarantine	10 s
Maximum session duration	900 s	Maximum input stream RTF	1.25

to its trigger result. These states are not collapsed:

attempted session
↓
protocol valid vs. protocol invalid
↓
behaviorally observable vs. not observable
↓
challenge realized vs. unrealized/timeout.

Consequently, conditional behavioral metrics are computed only on realized and observable challenges, while challenge-realization and trigger-timeout rates are reported with the planned-challenge denominator. Runtime failures, response-initiation failures, no-response questions, skipped challenge audio, and timeout events remain in the released execution records rather than being silently discarded.

B.8 Reproducible Protocol Configuration

All parameters in Table 8, including VAD thresholds, trigger conditions, timeout behavior, and session-reset rules, are frozen before evaluation. The runtime output records the effective configuration, stage identifiers, media and wall-clock timestamps, VAD decisions, trigger outcomes, timeout flags, and failure types, thereby making each adaptive execution auditable and reproducible.

C Metric Definitions and Denominators

This section defines the aggregate metrics used to evaluate full-duplex interaction and same-session memory behavior. Every metric is first computed independently for each sample and is then aggregated across samples using a macro-average. The resulting values therefore give every valid sample equal weight, regardless of the number of events or the duration of its interaction.

C.1 Aggregation and Missing-Value Policy

Let m_i denote the value of metric m for sample i, and let $\mathcal{I}_m$ be the set of samples for which m_i is defined. The reported aggregate is

$$\overline{m} = \frac{1}{|\mathcal{I}_m|} \sum_{i \in \mathcal{I}_m} m_i. \tag{6}$$

If $\mathcal{I}_m$ is empty, the metric is reported as unavailable. Missing values are not replaced with zero. An unanswered memory question is not a missing observation: it remains in the fixed question denominator and receives no credit. By contrast, an interaction metric without eligible observations is undefined and is excluded from that metric's mean. An unrealized challenge remains in the planned-challenge denominator for RCR but does not supply an observation for delivery-conditioned metrics.

All ratios are computed using sample-specific denominators before aggregation. The evaluation therefore does not pool all events, questions, or audio segments across the complete test set. This rule also applies to latency statistics: the median and the 90th percentile are first computed

within each sample and are then averaged across samples. Thus, `stop_latency_p50_s` and `stop_latency_p90_s` are macro-averaged sample-level quantiles, rather than quantiles computed from one pooled latency distribution.

C.2 Runtime Validity and Behavioral Observability

Protocol validity and behavioral observability are reported separately. For sample i, protocol validity is defined as

$$v_i = \mathbf{1}[\text{runtime timestamps are valid} \wedge \text{no runtime failure is recorded} \wedge \text{a decoded assistant-audio artifact exists}]. \tag{7}$$

The aggregate `protocol_valid` is the mean of v_i.

Behavioral observability is defined as

$$o_i = \mathbf{1}\left[d_i^{\text{asst}} > 0 \wedge n_i^{\text{raw}} > 1\right], \tag{8}$$

where d_i^{asst} is the decoded assistant-speech duration and n_i^{raw} is the number of samples in the raw assistant track. The aggregate `behaviorally_observable` is the mean of o_i. Consequently, a session can be protocol-valid while remaining behaviorally unobservable if the model produces no usable decoded speech.

C.3 Adaptive-Challenge Metrics

Let $\mathcal{P}_i$ be the set of distinct challenge identifiers planned in the manifest of sample i, and let $\mathcal{R}_i$ be the subset recorded as realized in the runtime trace. For an interruption or backchannel, realization requires successful audio injection after the corresponding assistant-speaking trigger has been satisfied. For wait-based challenges, realization follows the corresponding runtime observation rule. The sample-level challenge realization rate is

$$r_i = \frac{|\mathcal{R}_i|}{|\mathcal{P}_i|}. \tag{9}$$

Samples without a planned challenge do not contribute to this ratio.

Let T_i denote the number of challenge-trigger waits that time out before the required assistant state is observed. The trigger-timeout rate is

$$t_i = \frac{T_i}{|\mathcal{P}_i|}. \tag{10}$$

When a trigger times out, the corresponding interruption or backchannel audio is not injected and the challenge is not counted as realized.

The conditional full-duplex score, denoted by c_i, summarizes behavioral correctness on challenge instances for which the corresponding scenario can be evaluated. The scenario-specific correctness criteria are:

- timely stopping after an interruption;
- continuation of the ongoing response after a backchannel;
- absence of assistant intrusion during an annotated pause; and
- target-speaker selectivity during speech addressed to another participant.

The aggregate conditional score is

$$\texttt{conditional_cfd} = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} c_i, \tag{11}$$

where $\mathcal{I}_c$ contains samples with a defined conditional score. Because this score is conditioned on behaviorally scorable challenges, it must be interpreted together with `challenge_realization_rate`, `trigger_timeout_rate`, and `behaviorally_observable`.

The aggregate `adaptive_fd` is the macro-average of the corresponding sample-level adaptive full-duplex scores. It is an auxiliary summary that combines challenge opportunity and challenge behavior. It is not recomputed as the product of separately averaged aggregate quantities.

C.4 Ordinary-Turn Timing Metrics

Let $\mathcal{U}_i$ be the set of response-required semantic turns in sample i, and let $\mathcal{O}_i \subseteq \mathcal{U}_i$ be the subset satisfying the ordinary-turn eligibility conditions. An eligible ordinary turn must have a valid semantic boundary, valid speech-alignment markers, usable native audio-clock metadata, and an input stream within the prescribed real-time limit.

Response coverage is computed over $\mathcal{U}_i$:

$$\texttt{response_coverage}_i = \frac{\sum_{u \in \mathcal{U}_i} \mathbf{1}[\text{assistant speech is observed for } u]}{|\mathcal{U}_i|}. \tag{12}$$

The remaining ordinary-turn metrics use $\mathcal{O}_i$:

$$\texttt{premature_onset_rate}_i = \frac{\sum_{u \in \mathcal{O}_i} \mathbf{1}[\text{assistant onset precedes user end}]}{|\mathcal{O}_i|}, \tag{13}$$

$$\texttt{carry_in_rate}_i = \frac{\sum_{u \in \mathcal{O}_i} \mathbf{1}[\text{assistant is speaking at input onset}]}{|\mathcal{O}_i|}, \tag{14}$$

$$\texttt{overlap_before_end_rate}_i = \frac{\sum_{u \in \mathcal{O}_i} \mathbf{1}[\text{assistant overlaps user speech before user end}]}{|\mathcal{O}_i|}. \tag{15}$$

For pause handling, let $\mathcal{W}_i$ be the annotated pause-wait events. Pause intrusion is defined as

$$\texttt{pause_intrusion_rate}_i = \frac{\sum_{w \in \mathcal{W}_i} \mathbf{1}[\text{assistant speech is observed during } w]}{|\mathcal{W}_i|}. \tag{16}$$

These definitions distinguish a new premature response from speech carried over from a previous turn and use only turns with valid timing evidence.

C.5 Interruption Stop Metrics

Let $\mathcal{L}_i$ be the set of valid stop latencies for realized interruption events in sample i. A stop latency is included only when the interruption was actually injected and the decoded assistant waveform contains a valid speech-offset measurement. The sample-level latency statistics are

$$\ell_i^{50} = \text{Quantile}_{0.50}(\mathcal{L}_i), \tag{17}$$

$$\ell_i^{90} = \text{Quantile}_{0.90}(\mathcal{L}_i), \tag{18}$$

$$s_i^{800} = \frac{\sum_{\ell \in \mathcal{L}_i} \mathbf{1}[\ell \le 0.8\ \text{s}]}{|\mathcal{L}_i|}. \tag{19}$$

The macro-averages of these quantities are reported as `stop_latency_p50_s`, `stop_latency_p90_s`, and `stop_success_at_800ms`, respectively. Samples without a valid stop-latency observation are excluded from the corresponding aggregate and are not assigned a latency of zero.

The metric `assistant_speech_duration_s` is the macro-average of the total decoded assistant-speech duration per sample. It is a descriptive quantity rather than a quality measure.

C.6 Memory-Answer Normalization

Memory answers are obtained from the transcript associated with each question-specific answer window. When available, the frozen transcript associated with the answer window is used; otherwise, the corresponding answer artifact is used. Text is normalized with Unicode NFKC normalization, lowercasing, and removal of punctuation and symbols. English text is tokenized by whitespace-delimited words, while Chinese text is tokenized into CJK characters and alphanumeric units.

For token sequences A and B, multiset precision and recall are

$$P(A,B) = \frac{|A \cap B|}{|B|}, \qquad R(A,B) = \frac{|A \cap B|}{|A|}, \tag{20}$$

Table 9: Aggregate runtime metrics and their sample-level denominators.

Metric	Direction	Denominator or interpretation
protocol_valid	↑	Attempted samples.
behaviorally_observable	↑	Attempted samples; requires usable decoded assistant speech.
challenge_realization_rate	↑	Distinct planned challenge identifiers.
conditional_cfd	↑†	Challenge instances eligible for conditional behavioral scoring.
adaptive_fd	↑†	Sample-level adaptive full-duplex score; auxiliary summary.
trigger_timeout_rate	↓	Distinct planned challenges.
response_coverage	↑	Response-required semantic turns.
premature_onset_rate	↓	Eligible ordinary response turns.
carry_in_rate	↓	Eligible ordinary response turns.
overlap_before_end_rate	↓	Eligible ordinary response turns.
pause_intrusion_rate	↓	Annotated pause-wait events.
stop_latency_p50_s	↓	Valid interruption stop latencies, summarized within each sample.
stop_latency_p90_s	↓	Valid interruption stop latencies, summarized within each sample.
stop_success_at_800ms	↑	Valid interruption stop-latency observations.
assistant_speech_duration_s	–	Total decoded assistant-speech duration; descriptive only.

† Conditional or auxiliary measures that must be interpreted with challenge realization, trigger timeout, and observability statistics.

and the token overlap score is

$$F_1(A,B) = \frac{2P(A,B)R(A,B)}{P(A,B)+R(A,B)}. \tag{21}$$

A value is considered present when its normalized form is an exact substring of the answer or when token F_1 reaches 0.80 for values of at most two tokens and 0.60 for longer values.

C.7 Memory QA Metrics

Let $\mathcal{Q}_i$ be all memory questions in sample i, and let $\mathcal{Q}_i^+$ be the subset with a non-empty response that is not marked as `no_response`. Response and no-response rates are

$$\texttt{response_rate}_i = \frac{|\mathcal{Q}_i^+|}{|\mathcal{Q}_i|}, \tag{22}$$

$$\texttt{no_response_rate}_i = 1 - \texttt{response_rate}_i. \tag{23}$$

Let $\mathcal{Q}_i^{\text{latest}}$ denote the latest-value questions. For question q, let a_q indicate that an accepted updated value is present and let f_q indicate that a forbidden superseded value is present. Then

$$\texttt{latest_value_accuracy}_i = \frac{\sum_{q\in\mathcal{Q}_i^{\text{latest}}} \mathbf{1}[q \text{ answered} \wedge a_q \wedge \neg f_q]}{|\mathcal{Q}_i^{\text{latest}}|}, \tag{24}$$

$$\texttt{superseded_value_error}_i = \frac{\sum_{q\in\mathcal{Q}_i^{\text{latest}}} \mathbf{1}[q \text{ answered} \wedge f_q]}{|\mathcal{Q}_i^{\text{latest}}|}. \tag{25}$$

Accuracy conditioned on producing an answer is

$$\texttt{answer_accuracy_given_response}_i = \frac{\sum_{q\in\mathcal{Q}_i^{\text{latest}}\cap\mathcal{Q}_i^+} \mathbf{1}[a_q \wedge \neg f_q]}{|\mathcal{Q}_i^{\text{latest}} \cap \mathcal{Q}_i^+|}. \tag{26}$$

This metric separates incorrect answers from failures to answer.

Let $\mathcal{Q}_i^{\text{dist}}$ denote the distractor-ignore questions. A distractor is considered adopted when the answer is non-empty and does not explicitly reject the distractor, or when the answer directly matches distractor content without rejection. A distractor is considered rejected only when the model responds and the adoption condition is false:

$$\texttt{distractor_adoption_rate}_i = \frac{\sum_{q\in\mathcal{Q}_i^{\text{dist}}} \mathbf{1}[\text{distractor adopted}]}{|\mathcal{Q}_i^{\text{dist}}|}, \tag{27}$$

$$\texttt{distractor_rejection_accuracy}_i = \frac{\sum_{q\in\mathcal{Q}_i^{\text{dist}}} \mathbf{1}[q \text{ answered} \wedge \text{distractor not adopted}]}{|\mathcal{Q}_i^{\text{dist}}|}. \tag{28}$$

Table 10: Aggregate same-session memory metrics and their sample-level denominators.

Metric	Direction	Denominator or interpretation
response_rate	↑	All memory questions.
no_response_rate	↓	All memory questions; complementary to response rate.
latest_value_accuracy	↑	All latest-value questions, including unanswered questions.
superseded_value_error	↓	All latest-value questions.
distractor_adoption_rate	↓	All distractor-ignore questions.
distractor_rejection_accuracy	↑	All distractor-ignore questions; rejection additionally requires a response.
summary_fact_coverage	↑	Atomic reference facts in final-summary questions.
answer_accuracy_given_response	↑	Answered latest-value questions only.
legacy_token_f1	↑	All memory questions; retained as a compatibility measure.

These two measures are reported together with response_rate, since silence can reduce distractor adoption without demonstrating correct rejection.

For a final-summary question, the reference summary is split into atomic facts using sentence-final punctuation, semicolons, question marks, and line breaks. A fact is considered covered when its token F_1 with the answer is at least 0.50. If $\mathcal{F}_q$ is the reference fact set for question q, then

$$\text{coverage}(q) = \frac{\sum_{f \in \mathcal{F}_q} \mathbf{1}[F_1(f, \widehat{a}_q) \geq 0.50]}{|\mathcal{F}_q|}. \tag{29}$$

The sample-level summary_fact_coverage is the mean coverage over final-summary questions.

The metric legacy_token_f1 is the mean token F_1 between each answer and its full reference answer. It is retained for compatibility with earlier result tables, but it is not used as the sole measure of memory quality.

C.8 Denominator Reporting and Interpretation

For each evaluated model, the results report the total number of attempted samples, the number of protocol-valid samples, and the number of behaviorally observable samples. The total number of memory questions and the number of latest-value questions are also reported. Challenge, ordinary-turn, pause, and stop-latency denominators are retained in the sample-level records used for aggregation.

Conditional metrics must always be reported together with their coverage variables. In particular, conditional_cfd is accompanied by challenge realization and trigger-timeout rates; stop-latency measures are accompanied by the number of valid latency observations; and conditional memory accuracy is accompanied by response rate. A low error rate from a nearly silent model is therefore not interpreted as evidence of strong full-duplex or memory behavior. Stratified analyses by scenario or language reuse the definitions above and do not introduce additional metrics.

D Baseline Models and Inference Configurations

D.1 Evaluation Scope and Comparison Protocol

We evaluate five publicly available full-duplex speech models: MiniCPM-o 4.5 (Cui et al., 2026), Freeze-Omni (Wang et al., 2025c), Moshi (Défossez et al., 2024), Lychee-FD (Liu et al., 2026), and PersonaPlex-7B-v1 (Roy et al., 2026). The models are selected to cover different streaming speech architectures and native interaction interfaces. All baseline models use the same session manifests, planned event order, source audio clips, and scheduling rules. Because speech-triggered challenges depend on each model's realized output, the subset of challenge clips actually delivered can differ across models. A sample is executed as one persistent session: the session is reset before the sample begins, but not between context events, full-duplex challenges, recovery stages, or memory questions.

The baseline speech models are not provided with gold memory states, reference answers, alignment boundaries, event labels, addressee labels, or future events. This restriction concerns baseline-model inputs; the interfaces available to ACT-Mem's auxiliary modules are specified separately in Appendix E. All models are evaluated with the same adaptive execution controller, the same

Table 11: Native model interfaces and model-specific inference settings.

Model	Native interface	Input / output	Chunk or inference window	Model-specific condition	GPU
MiniCPM-o 4.5	Native streaming worker	16/24 kHz	80 ms input; 1000 ms native window	Greedy decoding, fixed seed, fixed reference audio	1
Freeze-Omni	Socket.IO-based streaming service	16/24 kHz	20 ms native input block	$p = 0$, $k = 1$, temperature $= 1.0$; neutral system prompt	1
Moshi	WebSocket streaming service	24/24 kHz	80 ms input block	Native server decoding; no evaluator-side task-specific sampling override	1
Lychee-FD	vLLM-backed streaming service	16/24 kHz	80 ms input; 400 ms native inference window	Fixed prompt voice; native backend decoding	2
PersonaPlex-7B-v1	Native WebSocket streaming service	24/24 kHz	80 ms input block	Fixed voice prompt and fixed random seed	1

challenge-triggering rules, the same session-reset policy, and the same answer-attribution procedure. Model-specific native chunk sizes, sampling rates, voice prompts, and transport protocols are retained when they are required by the corresponding implementation. These interface-specific parameters are reported explicitly rather than hidden under a common configuration.

D.2 Native Interfaces and Model Configurations

Table 11 summarizes the model-specific settings used for inference. Input and output sample rates refer to the native model bridge; the evaluator maintains a shared reference clock and performs any required resampling. A native inference window is distinct from the evaluator's event window and does not change the logical event sequence.

MiniCPM-o 4.5 uses greedy decoding with the frozen evaluation seed and a fixed reference audio prompt. Freeze-Omni uses the native service configuration with $p = 0$, $k = 1$, and temperature 1.0. Moshi is evaluated through its native streaming service without introducing an additional task-specific decoding rule. Lychee-FD uses the vLLM-backed native inference path with a fixed prompt voice and a 400 ms native inference window. PersonaPlex uses a fixed voice prompt and the evaluation seed for all sessions.

The native interfaces do not expose identical user-turn controls. Freeze-Omni supports an explicit response-boundary operation, whereas the other bridges record local event boundaries when an equivalent native message is unavailable. These local markers advance the evaluator's state machine but do not reset the model or claim that the model received an explicit semantic end-of-turn signal. This distinction is retained in the runtime artifacts.

D.3 Shared Evaluator-Side Configuration

All models are evaluated with a common evaluator-side configuration. User audio is streamed in real time as bounded event clips. Each delivered clip is played once, is not accelerated, and is not truncated because of the assistant's current output. Challenges whose trigger conditions are not met are omitted as specified below. The controller uses an 80 ms reference input chunk, a 20 ms polling interval, and a maximum input real-time factor of 1.25. Native model sample rates and chunk sizes are permitted, but all timing decisions are mapped to the shared reference clock.

Assistant speaking state is determined from decoded assistant PCM using the shared waveform-level VAD. Packet arrival, codec activity, and text-token activity are not treated as evidence of speech. Interruption audio is injected only after at least 500 ms of continuous decoded assistant speech, and backchannel audio is injected only after at least 800 ms of such speech. If the trigger condition is not satisfied within the 10 s trigger window, the challenge is recorded as unrealized and its audio is not injected.

The common protocol parameters are listed in Table 12. These parameters are frozen before evaluation and are not tuned on held-out sessions.

Table 12: Shared evaluator-side inference and timing configuration.

Parameter	Value
Reference input chunk	80 ms
Evaluator polling interval	20 ms
Event-window padding	120 ms
Assistant-start timeout	10 s
Assistant-end timeout	12 s
Interruption trigger speech	500 ms
Backchannel trigger speech	800 ms
Minimum response speech	160 ms
Stable assistant silence	2 s
Side-talk observation window	1.2 s
QA start timeout	30 s
QA answer timeout	30 s
QA pre-input quiet period	1.2 s
QA pre-input drainage timeout	60 s
Late-response quarantine	10 s
Maximum session duration	900 s

D.4 Prompt and Information Isolation

All baseline models receive a neutral voice-assistant instruction in the language of the corresponding sample. The default English instruction is "You are a voice assistant," with the corresponding neutral Chinese instruction for Chinese samples. The instruction is transported through the native interface when required by the model, but no task-specific memory or interaction instruction is added.

The following information is not exposed to any baseline model:

- gold ledger states or future memory updates;
- event types, challenge identifiers, or addressee labels;
- MFA word boundaries or evaluator-side VAD decisions;
- reference answers and expected memory values; and
- assistant reference audio generated from the benchmark annotations.

A model may therefore use only its native audio history and the neutral instruction supplied at session initialization. The proposed memory module, when evaluated in the controlled comparison, uses the same frozen `MiniCPM`-o 4.5 backbone and the same audio protocol; its additional memory conditioning is applied only through the method-specific inference path described in the main paper and the implementation appendix.

D.5 Audio Output, Transcription, and Answer Attribution

All assistant audio is captured from the native model bridge and retained as a decoded waveform. The waveform is used for VAD-based state detection, challenge triggering, stop-latency measurement, and response-onset attribution. Memory answers are evaluated from independent answer windows rather than from a concatenated session transcript.

For cross-model textual comparison, each answer waveform is transcribed with the same frozen external Whisper-small configuration. The language is taken from the sample manifest, decoding temperature is set to zero, and the same transcription backend is used for all models. Native model text, when available, is retained as a secondary artifact but is not used as the primary cross-model transcript. A transcription failure is recorded explicitly and makes the corresponding primary textual score unavailable; it is not silently converted to a zero-valued answer.

The answer window opens after decoded assistant speech onset and closes on decoded silence, an explicit turn boundary, or the configured answer timeout. A late response after timeout is kept in a quarantine interval and is not assigned retroactively to the expired question. No answer text is reused between questions.

D.6 Calibration, Test Isolation, and Reproducibility

Before the held-out evaluation, a fixed calibration split is used to freeze the shared VAD, challenge-trigger thresholds, timeout values, text normalization rules, and scoring parameters. The held-out sessions are not used for prompt tuning, decoding-parameter tuning, VAD adjustment, timeout selection, or checkpoint selection.

For every model, the evaluation record stores the public model identifier, checkpoint identity, repository revision when available, model-profile configuration, prompt transport, decoding settings, random seed, native sample-rate configuration, assigned GPU resources, and runtime environment identity. Checkpoint and configuration hashes are retained in the artifact manifest so that the reported results can be traced to an unambiguous model state without exposing machine-specific filesystem paths in the paper.

GPU isolation is enforced during inference. A run is considered invalid for paper reporting if an unrelated process contaminates an assigned GPU. The same session and sample identifiers are preserved across all runtime artifacts, allowing failed samples and resumed samples to be audited without double counting.

D.7 Failure Handling and Effective Denominators

Runtime failures, server disconnects, unsupported native controls, trigger timeouts, answer timeouts, and no-response questions are retained in the execution record. A failed sample is not silently removed from the attempted sample count. Instead, the evaluation reports protocol validity, behavioral observability, completion or failure status, and the effective denominator of every metric.

A challenge that is not realized because its trigger condition is not reached is distinguished from a challenge that is realized but handled incorrectly. Likewise, a valid session with no assistant speech is distinguished from a protocol-invalid session. Conditional behavioral scores are computed only when their required audio and timing evidence exists, while challenge realization, trigger timeout, and failure counts are reported separately. This accounting prevents differences in interface robustness, speech observability, and challenge exposure from being mistaken for differences in full-duplex behavior or memory fidelity.

E ACT-Mem Implementation Details

E.1 Implementation Scope and Frozen Components

ACT-Mem is implemented as an external memory module coupled to a frozen `MiniCPM-o` 4.5 full-duplex backbone (Cui et al., 2026). The backbone, including its acoustic encoder, streaming decoder, voice-generation components, and native full-duplex session manager, remains frozen throughout training and evaluation. A frozen speech transcription and timing component provides event-aligned textual observations to the memory controller. The trainable part consists of an event controller and a typed retriever, which jointly contain 8.92M parameters.

The online computation is organized into three stages:

$$\text{event observation} \longrightarrow \text{transactional memory update} \longrightarrow \text{typed QA retrieval}. \tag{30}$$

Context and full-duplex challenge events update the external memory state. During memory QA, the retriever constructs a compact evidence card from the active state, and the card is provided to the frozen backbone through the native text-prefill interface. The full-duplex audio stream and the memory controller therefore share the same persistent session.

E.2 Event Controller

For each delivered event e_t, a frozen ASR component transcribes the evaluator-delimited audio segment. The scheduler supplies the event index and speaker/addressee metadata. These metadata are distinct from the controller's learned predictions: a non-user or non-addressed event can veto admission through an explicit gate. The adapter uses the default audibility flag rather than supplying a separate online audibility estimate.

A lightweight text encoder and recurrent state module process the transcript and chronological context. Four prediction heads produce the operation, addressee, semantic-completeness, and confidence estimates:

$$\left(\widehat{o}_t, \widehat{a}_t, \widehat{c}_t, p_t^{\text{conf}}\right) = f_\theta\left(x_t, h_{t-1}\right), \tag{31}$$

where

$$\widehat{o}_t \in \{\textsc{Write}, \textsc{Update}, \textsc{Ignore}, \textsc{Retain}\}, \qquad \widehat{a}_t \in \{\text{assistant}, \text{other}\}, \qquad \widehat{c}_t \in \{0, 1\}.$$

The controller proposes atomic entity–value slots from each event. Each slot stores an entity key, value, event identifier, chronological event index, operation type, confidence, source role, and semantic-completeness flag. The controller output is converted into an admission score

$$g_t = p_t^{\text{addr}} p_t^{\text{complete}} \left[p_t(\textsc{Write}) + p_t(\textsc{Update})\right] c_t, \tag{32}$$

where p_t^{addr}, p_t^{complete}, and c_t denote the predicted addressee probability, completeness probability, and controller confidence, respectively.

A proposal is processed through the following transaction states:

$$\textsc{Prepare} \rightarrow \begin{cases} \textsc{Commit}, & g_t \geq \tau_{\text{write}}, \\ \textsc{Abort}, & \widehat{a}_t = \text{other or the event is inadmissible}, \\ \textsc{Uncertain}, & \text{otherwise}. \end{cases}$$

The calibration split selects $\tau_{\text{write}} = 0.35$. Committed proposals are passed to the versioned memory store, while aborted and uncertain proposals remain unavailable to ordinary retrieval.

E.3 Versioned Transactional Ledger

The memory store maintains active values, superseded versions, uncertain proposals, and provenance metadata. Each committed item is represented as

$$m_j = (k_j, v_j, \tau_j, \iota_j, \rho_j, \gamma_j, \sigma_j),$$

where k_j is the entity key, v_j is the value, τ_j is the source-event index, ι_j is the source event identifier, ρ_j is the operation label, γ_j is the confidence, and $\sigma_j \in \{\textit{active}, \textit{superseded}, \textit{uncertain}\}$ is the state status.

A Write operation creates a new active item. An Update operation creates a new active version and supersedes the previous active value associated with the same entity key:

$$k_j = k_i, \qquad \tau_j > \tau_i, \qquad \rho_j = \textsc{Update}. \tag{33}$$

The predecessor is moved to the superseded store and linked through `supersedes` and `superseded_by` fields. A Write or Update proposal whose admission score is below the threshold is retained as uncertain. Ignore records an aborted proposal, whereas Retain preserves the currently active state.

The ledger has a maximum active capacity of 32 slots. When the capacity is exceeded, lower-confidence and older active items are moved to the uncertain state while supersession links remain available for audit. Retrieval uses active items by default; superseded and uncertain items are retained for diagnostic analysis and error attribution.

E.4 Typed Retrieval and Evidence Cards

The adapter supplies a query type to the retriever. In the provided implementation, an annotated type is loaded from the configured QA metadata when available; otherwise, a deterministic rule routes the ASR-transcribed question. These annotation-assisted and text-routed modes must be distinguished when interpreting results. The query-type set is:

$$\mathcal{K} = \{\texttt{latest_value_qa}, \texttt{ordinary_memory_qa}, \texttt{final_summary_qa}, \texttt{distractor_ignore}\}.$$

The retrieval scope and maximum number of returned items depend on this type.

Table 13: Question routing and retrieval scope.

Question type	Retrieval behavior	Maximum items
Latest-value QA	Selects the target entity's current active value after ranking candidate facts.	2 candidates, 1 final value
Ordinary memory QA	Retrieves active facts relevant to the question.	4
Final-summary QA	Aggregates active facts across entities and memory types.	8
Distractor-ignore QA	Routes directly to rejection behavior without memory retrieval.	0

For a question q and memory item m, the retriever combines representation similarity, lexical overlap, recency, confidence, and state-based penalties:

$$S(q,m) = S_{\text{dense}}(q,m) + \alpha_{\text{lex}} S_{\text{lex}}(q,m) + \alpha_{\text{conf}}\gamma_m + \alpha_{\text{rec}} S_{\text{rec}}(q,m) - \alpha_{\text{bad}} B(m), \tag{34}$$

where $B(m)$ penalizes superseded, uncertain, or distractor items. The retriever ranks candidate items by $S(q,m)$ and applies a score-and-margin gate. An evidence card is generated only when the retrieved evidence passes the gate.

The evidence card contains the question type, entity keys, active values, source event identifiers, and a compact answering instruction:

$$C_q = \text{Serialize}\left(q,\ \{m_j :\ m_j \in \mathcal{R}(q)\}\right). \tag{35}$$

The card is provided once at the corresponding QA boundary through the backbone's native text-prefill interface. Context events and full-duplex challenge events continue through the native audio path, while memory conditioning is applied only to the associated same-session question.

The evidence-card intervention begins at the corresponding QA boundary and does not directly alter preceding interaction events. Cross-variant differences in pre-QA interaction metrics are therefore descriptive whole-run differences, rather than isolated causal effects of retrieval or ledger maintenance.

E.5 Training Objectives

Training uses session-disjoint event and QA records. The controller is trained to predict operation, addressee, completeness, and confidence:

$$\mathcal{L}_{\text{ctrl}} = \mathcal{L}_{\text{op}}^{\text{wCE}} + 0.3\mathcal{L}_{\text{addr}}^{\text{CE}} + 0.3\mathcal{L}_{\text{complete}}^{\text{CE}} + 0.2\mathcal{L}_{\text{conf}}^{\text{BCE}}. \tag{36}$$

The weighted operation loss uses weights $(1.0, 1.5, 1.4, 0.8)$ for WRITE, UPDATE, IGNORE, and RETAIN, respectively.

The retriever is trained with binary relevance supervision over active, superseded, forbidden, hard-negative, and distractor memory items:

$$\mathcal{L}_{\text{retr}} = \mathcal{L}_{\text{BCE}}(\mathbf{s},\mathbf{y}) + \frac{0.5}{|\mathcal{P}||\mathcal{N}|}\sum_{p\in\mathcal{P}}\sum_{n\in\mathcal{N}}\left[0.25 - s_p + s_n\right]_+, \tag{37}$$

where $\mathbf{s}$ denotes candidate scores, $\mathbf{y}$ denotes binary relevance labels, and $\mathcal{P}$ and $\mathcal{N}$ denote positive and negative candidate sets.

The total objective is

$$\mathcal{L}_{\text{ACT-Mem}} = \mathcal{L}_{\text{ctrl}} + \mathcal{L}_{\text{retr}}. \tag{38}$$

E.6 Optimization and Data Splits

The training data are divided by session into train, validation, calibration, and held-out partitions containing 660, 175, 40, and 219 sessions, respectively. The controller and retriever are jointly optimized in the first stage using AdamW with learning rate 3×10^{-4}, batch size 32, ten training epochs, weight decay 0.01, gradient-norm clipping at 1.0, and random seed 20260907.

In the second stage, the controller is fixed and the retriever is adapted with learning rate 1×10^{-4}, batch size 32, three adaptation epochs, and random seed 20260908. Checkpoints are selected using only the training and validation partitions. The 40-session calibration split is used for threshold

selection without gradient updates. Held-out sessions are reserved for the final comparison and are not used for parameter, checkpoint, or threshold selection.

Training is performed on eight NVIDIA A100 GPUs. The `MiniCPM`-o 4.5 backbone and the speech transcription and timing components remain frozen during both optimization stages.

E.7 Calibration and Same-Session Inference

Calibration evaluates candidate admission thresholds on the calibration sessions and selects the threshold that maximizes memory-eligibility F1. The selected threshold is then fixed for held-out inference. During a test session, the controller processes events in chronological order and updates the ledger after each event. The retriever is invoked only when a memory question reaches the stable-silence boundary.

The inference sequence for a memory question is

$$\begin{aligned}\text{stable assistant silence} &\rightarrow \text{question audio} \rightarrow \text{typed retrieval} \\ &\rightarrow \text{evidence-card construction} \rightarrow \text{native text prefill} \rightarrow \text{spoken answer}.\end{aligned} \tag{39}$$

The question audio and the evidence card are provided within the same persistent backbone session. Each question receives an independently constructed card, and the resulting answer is attributed to its own QA window.

E.8 Runtime Artifacts and Audit Records

For every evaluated session, ACT-Mem records the controller trace, ledger snapshot, retrieved memory identifiers, superseded identifiers, query type, retrieval scores, gate decision, evidence card, answer transcript, response/no-response status, and runtime latency. Each event trace contains the event identifier, event index, predicted operation, addressee, completeness, confidence, admission score, created slots, updated slots, superseded slots, and uncertain or aborted slots.

These records provide an auditable correspondence between an input event, its predicted memory operation, the resulting versioned state, the evidence shown at QA time, and the final spoken answer. They also preserve the effective session and question denominators used for the held-out evaluation.

F Training, Calibration, and Computational Cost

F.1 Training Setup

ACT-Mem is trained with session-disjoint data partitions containing 660 training sessions, 175 validation sessions, 40 calibration sessions, and 219 held-out sessions. The `MiniCPM`-o 4.5 full-duplex backbone and the speech transcription and timing components remain frozen. Only the event controller and typed retriever are optimized, resulting in 8.92M trainable parameters: 4.59M in the controller and 4.33M in the retriever.

Training consists of two stages. The first stage jointly optimizes the event controller and the typed retriever. The second stage initializes from the joint checkpoint, freezes the event controller, and further adapts the typed retriever using session-level memory evidence. Optimization uses AdamW with weight decay 0.01, batch size 32, and gradient-norm clipping at 1.0. The joint stage uses learning rate 3×10^{-4}, ten epochs, and random seed 20260907; the adaptation stage uses learning rate 1×10^{-4}, three epochs, and random seed 20260908. Training is performed on eight NVIDIA A100 GPUs.

All supervision is organized by session, and the held-out sessions are excluded from training, checkpoint selection, and calibration. The final checkpoint is loaded jointly with the frozen speech backbone during same-session evaluation.

F.2 Calibration Protocol

Calibration is performed on the dedicated 40-session calibration split without gradient updates. We search over candidate admission thresholds and select the value that maximizes event-memory eli-

Table 14: Training configuration.

Configuration	Value
Session split	660/175/40/219 train/validation/calibration/held-out
Trainable parameters	8.92M
Controller parameters	4.59M
Retriever parameters	4.33M
Optimization stages	Joint training followed by retriever adaptation
Optimizer	AdamW
Batch size	32
Weight decay	0.01
Gradient clipping	1.0
Joint-stage learning rate and epochs	3×10^{-4}, 10 epochs
Adaptation-stage learning rate and epochs	1×10^{-4}, 3 epochs
Training hardware	8 NVIDIA A100 GPUs

gibility F1. The resulting write threshold is

$$\tau_{\text{write}} = 0.35.$$

After calibration, the threshold is fixed for held-out inference. The calibration record stores the candidate thresholds, the selected value, the calibration split identifier, and the associated checkpoint identity.

F.3 Computational Cost

The computational cost reported for ACT-Mem is incremental relative to the frozen `MiniCPM-o 4.5` backbone. The additional trainable module contains 8.92M parameters, divided between the event controller and the typed retriever as shown in Table 14. During inference, the controller performs one chronological state update for each observed event, and the retriever constructs at most one evidence card for each memory question. The resulting card is inserted through the native question-conditioning path while the full-duplex backbone remains unchanged.

Runtime artifacts record the loaded checkpoint, effective calibration threshold, controller traces, retrieved memory identifiers, evidence cards, answer windows, and per-session failures. These records provide the basis for reporting the incremental memory-module cost separately from the fixed cost of full-duplex backbone inference.

G Statistical Analysis

G.1 Unit of Analysis and Aggregation

The session is the primary statistical unit. The six memory questions within a session, as well as the interaction events belonging to that session, are first aggregated into session-level quantities. Overall results are then computed by arithmetic macro-averaging over sessions:

$$\overline{m} = \frac{1}{|\mathcal{I}_m|} \sum_{i \in \mathcal{I}_m} m_i, \tag{40}$$

where m_i is the value of metric m for session i, and $\mathcal{I}_m$ is the set of sessions with a valid value for that metric.

Questions from the same session are not treated as independent statistical replicates. Failure handling is metric-specific. Unanswered memory questions receive no credit under fixed question denominators, and unrealized challenges remain in the planned-challenge denominator for RCR. Interaction metrics use their eligible-event denominators. A session with no valid denominator for a metric is excluded from that metric-specific mean, with the effective denominator reported; undefined values are not replaced by zero.

G.2 Paired Session Bootstrap

The controlled comparison between ACT-Mem and the frozen `MiniCPM-o 4.5` backbone uses the same 219 held-out sessions for both systems. We therefore use a paired percentile bootstrap at the

session level. For each bootstrap replicate b, 219 session identifiers are sampled with replacement. The outputs of both systems for each sampled identifier are kept paired, and the session-level metric is recomputed before aggregation.

For metric m, the paired effect is

$$\Delta_m = \overline{m}_{\text{ACT-Mem}} - \overline{m}_{\text{MiniCPM}}, \tag{41}$$

where the two means are computed on the same effective set of sessions for that metric. We generate 20,000 bootstrap replicates using random seed 20260807. The 95% percentile confidence interval is

$$\text{CI}_{95}(m) = \left[Q_{0.025}\left(\{\Delta_m^{(b)}\}_{b=1}^{20000} \right), Q_{0.975}\left(\{\Delta_m^{(b)}\}_{b=1}^{20000} \right) \right], \tag{42}$$

where Q_α denotes the empirical α-quantile of the bootstrap effect distribution.

The primary inferential intervals are reported for latest-value accuracy and final-state fact coverage. When intervals are available for additional memory or interaction metrics, they are computed using the same paired session-level procedure. Absolute changes in percentage points are reported as the primary effect size; relative percentage changes are descriptive summaries.

Pairing is by session identifier and does not imply identical realized challenge delivery across systems. The intervals quantify session-sampling uncertainty conditional on the recorded runs and the specified scorer; they do not account for repeated-run generation variability, ASR errors, or scoring-rule validity.

G.3 Baseline and Subgroup Reporting

The results for the five full-duplex baseline models are primarily descriptive. Each model is evaluated on the same session manifests and is aggregated with the same session-level procedure. No scalar ranking is derived by combining interaction and memory metrics. In particular, challenge realization is interpreted as exposure, while conditional full-duplex scores are interpreted together with challenge realization, trigger-timeout rate, and behavioral observability.

Language-, scenario-, and question-type analyses are reported as exploratory stratified summaries. The same definitions, denominators, and macro-aggregation rule are used within each stratum. Each subgroup result is accompanied by its session count and effective metric denominator. These stratified analyses are not treated as independent confirmatory tests, and no additional significance claims are made without a prespecified multiple-comparison procedure.

G.4 Denominators and Reproducibility

For every reported metric, we retain the attempted-session count and the effective denominator used for aggregation. Runtime metrics additionally retain protocol-valid and behaviorally observable session counts, while memory metrics retain the total number of questions and the number of questions of each relevant type. Conditional quantities, including conditional full-duplex behavior and answer accuracy given a response, are reported together with their coverage variables.

All bootstrap computations use the fixed session identifiers, the fixed held-out split, and the fixed random seed specified above. The released statistical records contain session-level metric values, effective denominators, paired differences, bootstrap settings, and confidence intervals, allowing the reported aggregate results to be reproduced without treating individual questions as independent observations.

H Complete Benchmark Results

H.1 Evaluation Scope

This section reports the complete aggregate results for `MiniCPM-o 4.5`, `Freeze-Omni`, `Moshi`, `Lychee-FD`, and `PersonaPlex-7B-v1` on all 1,094 sessions. Each model is evaluated on the same immutable session manifests, comprising 6,564 same-session memory questions and checks, including 2,188 latest-value questions. All ratios are calculated within each session and then macro-averaged over sessions. The tables below include all aggregate runtime and memory metrics defined in Appendix C.

Table 15: Runtime validity, challenge exposure, and response coverage on all 1,094 sessions. Values are percentages except assistant speech duration.

Model	Protocol valid↑	Behaviorally observable↑	Challenge realization↑	Conditional cFD↑	Adaptive FD↑	Trigger timeout↓	Response coverage↑	Speech duration (s)
MiniCPM-o 4.5	100.00	100.00	73.51	40.23	35.28	6.04	28.38	78.35
Freeze-Omni	100.00	100.00	86.17	28.40	30.41	0.53	76.03	203.16
Moshi	100.00	100.00	44.63	37.36	13.08	27.67	12.61	29.23
Lychee-FD	99.82	100.00	75.64	47.32	48.19	1.09	35.30	99.19
PersonaPlex-7B-v1	100.00	100.00	93.54	22.74	35.41	1.73	32.43	74.20

Notes. Conditional cFD is conditioned on behaviorally scorable challenges and must be interpreted together with challenge realization and trigger timeout. Adaptive FD is aggregated from session-level scores and therefore need not equal the product of the two separately aggregated columns.

Table 16: Turn-boundary and interruption-handling results. Rates are percentages and stopping latencies are measured in seconds.

Model	Premature onset↓	Carry-in↓	Pre-end overlap↓	Pause intrusion↓	Stop P50↓	Stop P90↓	Stop@ 0.8 s↑
MiniCPM-o 4.5	18.70	25.86	44.38	59.47	1.032	1.446	55.69
Freeze-Omni	49.76	42.44	91.82	73.64	1.239	1.802	45.75
Moshi	66.59	5.49	72.25	45.11	0.777	0.864	60.79
Lychee-FD	24.68	32.96	57.14	53.89	0.312	0.440	90.12
PersonaPlex-7B-v1	46.64	26.23	72.73	90.62	1.689	2.104	16.40

H.2 Runtime Validity and Challenge Exposure

Table 15 reports execution validity, behavioral observability, adaptive-challenge exposure, conditional full-duplex behavior, response coverage, and assistant speech duration. Challenge realization measures the fraction of planned challenges that became behaviorally realized. Conditional cFD evaluates behavior on scorable challenges, whereas Adaptive FD is the macro-average of the corresponding session-level adaptive score. Assistant speech duration is descriptive and is not treated as a performance measure.

H.3 Turn-Boundary and Interruption Metrics

Table 16 reports ordinary-turn boundary behavior, pause intrusion, and interruption stopping. Premature onset measures assistant speech beginning before the valid user-end boundary. Carry-in indicates that assistant speech from the preceding response remains active when the next user input begins. Pre-end overlap records any assistant speech overlapping the user before the user-end boundary.

H.4 Same-Session Memory Metrics

Tables 17 and 18 report all aggregate memory metrics. Latest-value accuracy requires the updated value to be present and the superseded value to be absent. Answer accuracy given response evaluates the same criterion only among latest-value questions for which an answer was produced.

H.5 Effective Denominators

Table 19 reports the principal aggregate denominators. All models are evaluated on 1,094 attempted sessions. Lychee-FD contains 1,092 protocol-valid sessions; its remaining behaviorally observable sessions are retained in the attempted-session accounting. Metric-specific denominators for challenge, timing, pause, and interruption statistics follow the definitions in Appendix C.

Table 17: Response availability and factual-update results. All values are percentages.

Model	Response rate↑	No-response rate↓	Latest-value accuracy↑	Superseded-value error↓	Accuracy given response↑
MiniCPM-o 4.5	98.90	1.10	4.30	0.59	4.32
Freeze-Omni	99.94	0.06	11.61	3.38	11.61
Moshi	54.31	45.69	2.19	0.32	3.60
Lychee-FD	92.41	7.59	25.50	1.65	30.80
PersonaPlex-7B-v1	96.40	3.60	5.85	2.29	5.93

Table 18: Distractor handling, final-state coverage, and answer-overlap results. All values are percentages.

Model	Distractor adoption↓	Distractor rejection↑	Summary fact coverage↑	Legacy token F_1↑
MiniCPM-o 4.5	14.17	85.19	2.01	29.45
Freeze-Omni	65.90	34.10	0.00	15.42
Moshi	44.15	8.32	0.04	7.64
Lychee-FD	48.17	51.55	1.06	21.36
PersonaPlex-7B-v1	79.71	17.46	0.24	7.70

Notes. Distractor adoption and rejection are reported together because silence may reduce adoption without demonstrating correct rejection. Legacy token F_1 is retained for compatibility and is not used as the sole measure of memory fidelity.

I Qualitative Examples and Failure Analysis

I.1 Example Selection and Presentation

We present representative traces that expose failure modes not fully visible from aggregate scores. The examples were selected from completed, protocol-valid evaluation sessions and were not used for training, calibration, or checkpoint selection.

Each trace combines the scheduled challenge, the expected state transition, runtime interaction measurements, and the model's spoken answer. Spoken outputs are reported from the frozen-ASR transcripts, with only whitespace and terminal punctuation normalized for readability. The examples are diagnostic rather than estimates of failure frequency.

I.2 Interaction and Memory Can Fail Independently

Table 20 summarizes examples covering the four full-duplex scenarios. The traces demonstrate that correct interaction timing does not guarantee coherent memory and, conversely, that recovering an isolated fact does not imply correct full-duplex behavior.

Interruption is not equivalent to revision. The interruption example shows the clearest interaction–memory dissociation. The model stopped within the required interval for both realized challenges, yielding perfect stopping success in the selected session. However, it recovered only one of the two corrected values and admitted information explicitly marked as irrelevant. The interruption mechanism therefore reacted to the acoustic event without constructing the complete post-correction state.

Pause errors can precede otherwise plausible answers. In the pause-handling example, both models spoke during a region that should have been treated as an unfinished user turn. Their later memory behaviors diverged: one was selective but unable to state the revised values, whereas the other recovered a revised value but admitted a personal item. This case illustrates why a plausible answer to one question cannot substitute for joint measurement of interaction boundaries and persistent-state fidelity.

Explicit rejection is stronger evidence than silence. A model receives distractor-rejection credit only when its spoken answer explicitly rejects the irrelevant information. No response remains an

Table 19: Principal denominators for the complete benchmark evaluation.

Model	Attempted sessions	Protocol-valid sessions	Observable sessions	Memory questions	Latest-value questions
MiniCPM-o 4.5	1,094	1,094	1,094	6,564	2,188
Freeze-Omni	1,094	1,094	1,094	6,564	2,188
Moshi	1,094	1,094	1,094	6,564	2,188
Lychee-FD	1,094	1,092	1,094	6,564	2,188
PersonaPlex-7B-v1	1,094	1,094	1,094	6,564	2,188

Table 20: Representative qualitative examples from protocol-valid sessions. RCR and cFD are reported as proportions for the selected session.

Scenario	Expected behavior	Observed evidence	Diagnosis
Interruption	Stop the current response, replace *Saturday evening* with *Saturday morning*, and replace *twenty-five minutes* with *ten minutes*.	Both interruptions were realized, with median stopping latency 0.368 s and Stop@0.8 of 1.00. The model recovered *Saturday morning*, but did not recover *ten minutes*. When asked about the side information, it answered that the distractor should be saved.	Fast interruption stopping was accompanied by only partial factual revision and incorrect admission. This separates acoustic interruption recovery from transactional memory updating.
Pause handling	Remain silent during within-turn pauses, then commit the completed correction: *Thursday morning* and *fifteen minutes*.	For the same session, both examined models intruded into every evaluated pause. One model rejected the distractor but recovered neither scored latest value. The other recovered *Thursday morning*, but answered "I will save all the details into the final memory" for the distractor question and later included a *travel toothbrush* in its summary.	The first behavior is a missed update; the second is an admission failure followed by summary contamination. Both occur together with incorrect pause-boundary control.
Backchannel	Continue the current response without treating the backchannel as a new turn or a persistent fact.	All three challenges were realized, but session cFD was 0.33. The model nevertheless recalled the stable requirement to avoid hazelnut flavoring and retain whole beans, and correctly stated that the kitchen-clutter detail should not be saved. Its final summary remained incomplete.	Correct retention and distractor rejection can coexist with poor continuation behavior. Backchannel handling and final-state consolidation therefore require separate attribution.
Speech to others	Do not answer the non-addressed speech and do not admit its content into memory.	The session obtained RCR 0.75 and cFD 0.25. The model explicitly rejected the side information, but failed both latest-value questions and produced an incomplete final summary.	Rejecting a queried distractor does not establish that the system maintained the intended active state. Addressee control, supersession, and later answer generation remain distinct stages.

Quoted outputs are frozen-ASR transcriptions. The interpretations use the same fixed-denominator decisions as the quantitative evaluation; no qualitative relabeling is applied.

unanswered case. This distinction prevents systems with low response coverage from appearing selective merely because they remain silent.

I.3 Auditable Transaction Trace

Table 21 presents an ACT-Mem trace from a held-out English pause-handling session. The user first supplied provisional budgeting values, later corrected the review time to *this Saturday morning* and the electricity cap to *$150*, and explicitly marked an unrelated detail as unsuitable for storage.

This trace illustrates the value of retaining transaction provenance. The high-level UPDATE decision was correct, but the correction did not supersede the provisional entries because the extracted entity representation was inconsistent. An answer-only evaluation would record a failed latest-value question without distinguishing a missed correction from failed entity resolution, failed supersession, retrieval error, or response omission.

The same trace also reveals a limitation of clause-level admission. A negated exclusion statement may contain a fluent entity–value pattern, causing its literal wording to be stored even though its pragmatic function is to prohibit storage. More robust semantic scope modeling is therefore required for utterances that combine relevant facts, corrections, and explicit non-persistence instructions.

Table 21: An ACT-Mem transaction trace illustrating both successful control decisions and downstream state-construction failures.

Stage	Gold transition	Predicted transaction	Consequence
Initial request	WRITE provisional review time and budget constraints.	The controller predicted WRITE and committed extracted entity–value slots.	The initial values became active and were available to subsequent retrieval.
Correction	UPDATE the review time to *this Saturday morning* and the electricity cap to *$150*; supersede both provisional values.	The controller predicted UPDATE, but the extracted correction was represented as one compound slot with an entity key different from the provisional slots.	A new item was created, while the original items remained active and no supersession link was formed. The failure therefore occurred after the operation decision, during slot normalization and ledger resolution.
Distractor-bearing event	Commit the relevant condition and IGNORE the explicitly excluded side detail.	The relevant condition was written, but the exclusion clause "just a side detail and should not be saved" was also extracted as an active fact.	The ledger became contaminated even though the user's utterance explicitly expressed non-persistence. This is an incorrect-admission failure rather than a retrieval-only failure.
Latest-value QA	Retrieve the active corrected electricity cap and answer *$150*.	No spoken answer was produced.	The case remains incorrect under the fixed denominator. The trace localizes the outcome to unresolved state construction followed by answer realization.
Distractor QA	Produce an explicit rejection without exposing the side detail as evidence.	No spoken answer was produced.	Neither adoption nor successful rejection is credited; silence is kept separate from selective exclusion.

Controller decisions and slot contents are derived from frozen-ASR event transcripts. The trace reports the stored state before answer scoring and therefore permits failures to be localized without inferring hidden state from the spoken answer alone.

Table 22: Failure taxonomy used for qualitative attribution.

Failure	Observable criterion
Trigger timeout	A speech-dependent challenge is planned but cannot be injected because the required assistant-speech condition is not reached.
Boundary violation	The assistant begins before a completed user turn, speaks during a protected pause, or intrudes into a non-response window.
Slow interruption stop	Assistant speech continues beyond the stopping threshold after an interruption is delivered.
Missed write	An admissible new fact produces no committed memory item.
Incorrect admission	Non-addressed, incomplete, backchannel, or explicitly excluded content is committed as persistent memory.
Failed supersession	A valid correction is detected, but the prior active value is not linked and marked as superseded.
Stale retrieval	A superseded value is exposed to the answer generator despite the presence of a valid active replacement.
Distractor leakage	An excluded item enters the evidence card or appears affirmatively in the spoken answer or final summary.
Retrieval miss	The required active item exists in the ledger but is absent from the question-conditioned evidence.
Answer realization error	Appropriate evidence is available, but the spoken response omits, distorts, or contradicts it, or no answer is produced.
Transcription error	The frozen-ASR transcript changes a value or boundary in a way that affects controller input or answer scoring.

I.4 FAILURE TAXONOMY

We assign each qualitative error to the earliest observable stage that can explain it. Table 22 defines the resulting taxonomy. Later symptoms are retained for analysis but are not treated as evidence that all earlier stages were correct.

The examples identify three recurring patterns. First, interaction success is local: rapid stopping or correct backchannel continuation does not guarantee admission or revision of the corresponding facts. Second, memory selectivity is compositional: a system may reject one queried distractor while retaining stale values or producing an incomplete summary. Third, an explicit transaction trace makes compound errors separable. In particular, correct operation classification does not ensure correct slot extraction, entity resolution, supersession, retrieval, or spoken realization. These distinctions explain why the benchmark reports interaction boundaries and memory outcomes separately and why no single scalar score is sufficient to characterize full-duplex persistent memory.

Table 23: Artifact manifest for reproducing the benchmark and experiments.

Artifact group	Representative records	Reproducibility role
Session inputs	`source.json`, session audio, event audio	Defines language, scenario, event order, speaker roles, timing, and the rendered input waveform.
Interaction manifests	`interactive_manifest.json`, challenge and timing manifests	Specifies challenge conditions, observation windows, trigger rules, and the fixed execution order.
Memory supervision	`memory_qa.json`, reference-state records, update links	Provides question types, accepted and forbidden values, expected operations, supporting evidence, and final-state supervision.
Reference behavior	`reference_behavior.json`	Stores the intended interaction response for each challenge and the corresponding state-transition constraints.
Runtime execution	`runtime_events.jsonl`, VAD and timing records	Records challenge realization, response boundaries, interruption stopping, overlap, pauses, and execution failures.
Generated outputs	Assistant waveform, raw predictions, QA windows, frozen-ASR transcripts	Preserves the model output and the temporal windows used for answer attribution and transcription.
Question-level scores	Per-question score records and session summaries	Allows reconstruction of response coverage, latest-value accuracy, stale assertions, distractor outcomes, and final-state coverage.
ACT-Mem traces	Controller–ledger traces, retrieval records, evidence cards	Exposes event operations, committed items, supersession links, active values, and question-conditioned evidence.
Configuration and scripts	Inference, scoring, aggregation, and calibration configurations	Reproduces execution settings and regenerates the reported tables from the recorded outputs.

J Reproducibility and Artifact Manifest

This section specifies the artifact organization required to reproduce the reported experiments. All released records use stable session identifiers and preserve the mapping from each input session to its events, challenges, questions, model outputs, and scores. The artifact structure is summarized in Table 23.

J.1 Artifact Organization

The records are linked by session identifier and question identifier. Consequently, a question-level score can be traced back to its input audio, the corresponding event prefix, the realized challenge schedule, the generated answer, and the evidence used for scoring. Failed, unanswered, and timed-out executions remain in the released records and are not removed before aggregation.

J.2 Data Splits and Provenance

The release-level development and test partitions contain 237 and 857 sessions, respectively. The method-development partitions contain 660 training sessions, 175 validation sessions, 40 calibration sessions, and 219 held-out sessions. Splits are defined at the session level, so that events, questions, and audio derived from one session cannot appear in more than one partition.

The training and validation records are used for parameter fitting and checkpoint selection. The calibration records are used only for selecting the admission threshold. The held-out records are reserved for the final comparison and are not used for training, calibration, or model selection. Each reported held-out result can therefore be reconstructed from the corresponding fixed session manifest and its stored output records.

J.3 Execution and Score Reconstruction

The complete reproduction path is

$$\begin{aligned}\text{session manifest} &\to \text{streaming execution} \to \text{runtime event log} \\ &\to \text{generated audio} \to \text{frozen-ASR transcript} \\ &\to \text{question-level scoring} \to \text{session aggregation} \to \text{reported table}.\end{aligned} \tag{43}$$

The runtime log determines whether an adaptive challenge was realized and provides the timestamps required for interaction metrics. The question records determine the fixed denominators for

memory evaluation. Session ratios are computed before macro-averaging across sessions, and unanswered or invalid responses remain in the corresponding denominators. The final aggregation files are generated from these session-level records rather than manually transcribed into the tables.

For the controlled comparison, the held-out session identifiers are retained in paired form for the backbone and memory-augmented runs. This allows the reported confidence intervals to be regenerated with paired session resampling while preserving the correspondence between the two conditions.

J.4 Integrity and Reproduction Checks

Before scoring, the reproduction script verifies that every session manifest has a matching input waveform, event list, challenge record, and question set. It also checks that event and question references are unique and resolvable, that all event timestamps fall within the declared session duration, and that no session identifier occurs in multiple partitions. After inference, the script verifies that every output can be mapped to a fixed input session and that the score denominators agree with the released question manifests.

These checks ensure that the reported results are reproducible from session-level inputs and recorded outputs, while preserving the traceability required to inspect individual interaction and memory failures.