WHY FULL-DUPLEX SPEECH MODELS COLLAPSE IN LONG CONVERSATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Full-duplex speech models listen and speak at the same time, which makes them promising for natural, open-ended conversation. But can they sustain a coherent conversation for tens of minutes? Because they generate a token every frame, even while listening, they exhaust their context windows within minutes. Yet existing benchmarks evaluate only single turns or a few conversational rounds, too short to expose the long-horizon problem. We introduce the Full-Duplex Long-Horizon Benchmark (FD-LHB), which tests five recent full-duplex models on long replayed conversations and in live conversation with an LLM partner. Every evaluated model eventually either degrades or exhausts its context. Models that keep the full history often degrade before their context runs out, while sliding-window models still fall into *repetition*, *silence*, and *gibberish*. For sliding-window models, we identify two contributing mechanisms. Gibberish and silence are linked to Initialization-Anchor Eviction, where the sliding window discards the first KV entry. Repetition is also reinforced by the model's own recent outputs. Guided by this diagnosis, we apply two simple fixes: Initialization-Anchor Pinning (IAP) keeps that first entry, and Duplex Repetition Penalty (DRP) discourages echoing while guarding against prolonged silence. Together, they support the diagnosis and, without retraining, substantially reduce conversational collapse in replays of up to an hour while running in real time.

1 INTRODUCTION

Human conversation can continue for tens of minutes or longer, as speakers move between long turns, silence, interruptions, backchannels (e.g., "uh-huh" or "yeah"), and overlapping speech (Levinson & Torreira, 2015). Full-duplex speech models (Défossez et al., 2024; Roy et al., 2026; Kim et al., 2026; Fang et al., 2026) aim to support this form of interaction by listening and speaking concurrently. Maintaining this behavior over long conversations remains difficult because, unlike text LLMs or half-duplex systems, full-duplex models continuously generate tokens frame by frame, even while listening. This rapidly fills the context with silence and recurring backchannels, while each generated output also becomes part of the next input and can influence later generation.

0 min — coherent "Hi, my name's Alex. I'm in the Seattle area. Where are you?"

6 min — repetition "...**you go back to your normal life**, **you go back to your normal life**, and then **you go back to your normal life**..."

9–10 min — silence (no lexical output)

13 min — gibberish "core offensive of the core the good blue of the event seven..."

Figure 1: **Example of long-conversation collapse.** It shows progression of coherent speech, repetition, silence, and gibberish, with timestamps aligned to the timeline in Figure 2a.

Despite these challenges, long-horizon behavior in full-duplex models remains poorly understood. Existing full-duplex benchmarks (Lin et al., 2025; 2026b;a; Peng et al., 2025; Arora et al., 2025; He et al., 2026) evaluate capabilities such as turn-taking, interruption handling, pauses, backchanneling, and overlapping speech, within a single turn or a few conversational rounds, not over many minutes. Existing models manage conversation history in one of two ways. Growing-context models (Kim et al., 2026; Fang et al., 2026) keep the full interaction, so their context grows until it reaches a context limit. Sliding-window models (Défossez et al., 2024; Roy et al., 2026) keep only the most recent frames, discarding the oldest as new ones arrive, so they can run indefinitely with fixed-size cache. Whether either design is sufficient for long-form full-duplex interaction, and how the models fail when it is not, has not been systematically investigated.

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

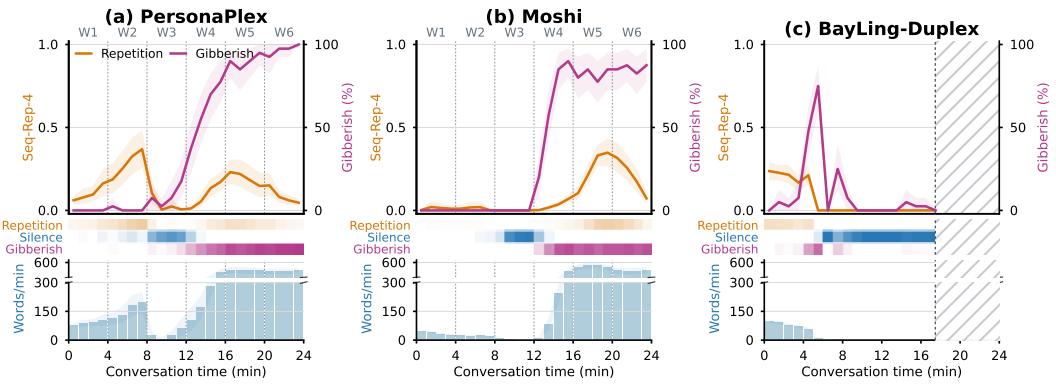


Figure 2: **How three models degrade over a long conversation.** Top: repetition and gibberish rate; middle: occurrence of each failure mode; bottom: output words per minute. Sliding-window models (a) PersonaPlex and (b) Moshi exhibit an initial period of coherent speech, followed by silence and eventual gibberish, with repetition emerging at different times. The growing-context model (c) BayLing-Duplex develops gibberish and silence before reaching its context limit.

To investigate this problem, we introduce the **Full-Duplex Long-Horizon Benchmark (FD-LHB)**, a benchmark designed specifically to evaluate full-duplex models over extended interaction. FD-LHB combines controlled replay of CANDOR conversations (Reece et al., 2023) with adaptive interaction against an LLM-based spoken partner. Using FD-LHB, we find that every evaluated model eventually either degrades, through repetition, silence, or gibberish (Fig. 1), or exhausts its context: sliding-window models collapse even though their memory never runs out (Fig. 2a,b), and growing-context models can collapse before reaching their context limit (Fig. 2c).

We then ask why these collapses occur, focusing on the two sliding-window models, and identify two contributing mechanisms (Sec. 4). Silence and gibberish are linked to **Initialization-Anchor Eviction**: the sliding window eventually discards the first KV entry computed without prior cached context. We hypothesize that this dependence reflects a mismatch between empty-cache training and continual inference. We also identify **Self-Reinforcing Repetition**: a word the model has recently said becomes more likely to be said again.

Finally, these mechanisms motivate two simple fixes that need no retraining. *Initialization-Anchor Pinning (IAP)* keeps the first KV entry in the cache while the rest of the window slides as usual, and *Duplex Repetition Penalty (DRP)* discourages recently used words without making the model less likely to speak. Together, they substantially reduce repetition, silence, and gibberish in both sliding-window models. In addition, the combined method still runs in real time and is faster than a StreamingLLM-style baseline (Xiao et al., 2024b) while matching its cache stability. We do not claim that they solve long-horizon full-duplex modeling; rather, our results show that simple fixes targeting the two identified causes prevent much of the collapse.

Our contributions are as follows:

- **FD-LHB: benchmark for long-horizon full-duplex interaction.** FD-LHB measures when and how conversational failures emerge through two tracks: 40 controlled replays evaluated up to 60 minutes (FD-LHB-CANDOR) and 40 adaptive sessions up to 20 minutes (FD-LHB-LLM).
- **Systematic characterization and diagnosis of collapse.** Across five recent full-duplex models using sliding-window or growing-context, we characterize the onset and progression of three failure modes: repetition, prolonged silence, and gibberish. Experiments on two sliding-window models identify Initialization-Anchor Eviction and Self-Reinforcing Repetition as mechanisms underlying the collapse.
- **Validation through training-free interventions.** Simple, targeted fixes reduce gibberish from 100% to 0%. Repetition falls from 65% to 0% in Moshi and 90% to 2.5% in PersonaPlex; prolonged silence falls from 100% to 7.5% and 85% to 47.5%, respectively. These interventions require no retraining and can be incorporated into compatible baselines at inference time.

2 Related Work

2.1 Full-Duplex Speech Generation and Streaming Interaction

Models such as Moshi (Défossez et al., 2024), PersonaPlex (Roy et al., 2026), Raon-SpeechChat (Kim et al., 2026), BayLing-Duplex (Fang et al., 2026), and Lychee-FD (Liu et al., 2026) support low-latency full-duplex interaction. Existing benchmarks (Lin et al., 2025; 2026b;a; Peng et al., 2025; Arora et al., 2025; He et al., 2026; Wang et al., 2026a; Ray et al., 2026) evaluate turn-taking, overlap, interruption handling, multi-round quality, and task completion, but do not examine their stability over extended conversations. We address this gap by evaluating repetition, silence, and gibberish during continuous generation lasting tens of minutes.

2.2 Long-Horizon Stability and KV-Cache Management

Autoregressive generation can suffer from degeneration in text (Holtzman et al., 2019), video (Ge et al., 2022), and speech (Song et al., 2025; Chen et al., 2024). Scheduled Sampling (Bengio et al., 2015) and Professor Forcing (Lamb et al., 2016) address the mismatch between training and generation through modified training procedures. We instead investigate instability during continual full-duplex inference, where generated outputs can reinforce later repetition (Xu et al., 2022; Zhu et al., 2023). DRP targets this feedback by modifying token selection without retraining.

Beyond repetition, continual inference raises the question of how conversation history should be retained. Under a fixed memory budget, StreamingLLM (Xiao et al., 2024b) preserves initial attention sinks together with recent tokens, as do LM-Infinite (Han et al., 2024) and InfLLM (Xiao et al., 2024a); H2O (Zhang et al., 2023) retains tokens with high accumulated attention; and SnapKV (Li et al., 2024) selects prompt positions using attention patterns from an observation window. Building on these studies, we examine the role of the first KV entry in the long-horizon stability of full-duplex speech models. StreamingLLM is the closest to IAP because it also preserves entries from the beginning of the sequence. In Moshi and PersonaPlex, the first KV entry alone prevents gibberish in all tested runs, whereas the tested later entries do not do so consistently (Sec. 4.1.2). We hypothesize that this dependence reflects initialization-anchor eviction: training chunks always begin with this entry, but long inference evicts it (Sec. 4.1). IAP keeps only this entry and updates only its position index, which is cheaper than re-indexing the whole cache as StreamingLLM does (Sec. 6.4).

3 Full-Duplex Long-Horizon Benchmark

3.1 Evaluation Protocol

Existing benchmarks do not test whether these models remain stable over sustained interactions lasting tens of minutes. To address this gap, we introduce the Full-Duplex Long-Horizon Benchmark (FD-LHB).

FD-LHB has two complementary tracks: controlled replay (**FD-LHB-CANDOR**) and adaptive interaction (**FD-LHB-LLM**). Replay (FD-LHB-CANDOR) gives every model the same recorded human speech, enabling controlled comparison, but it is non-interactive: the input doesn't respond to the model's output. Adaptive (FD-LHB-LLM), by contrast, is interactive, with an LLM partner whose synthesized speech reacts to the model's behavior like a real conversation, but this means the input differs across runs, making controlled comparison harder. Because each track compensates for the other's weakness, the two are complementary.

FD-LHB-CANDOR. FD-LHB-CANDOR is built from CANDOR (Reece et al., 2023), a corpus of long, natural human conversations. We remove the first three turns from each conversation to exclude greetings and initial silence, rank the conversations by their remaining duration, and select the longest 40. The resulting streams have minimum, mean, and maximum durations of 55.5, 67.4, and 92.6 minutes, respectively. We evaluate each replay for up to 60 minutes, or until the available conversation ends. For each conversation, we select the speaker channel containing more speech and use it as the fixed audio input to the evaluated model. Thus, although the original CANDOR recordings are interactive human conversations, FD-LHB-CANDOR replays the same fixed input to every model, enabling controlled comparison.

FD-LHB-LLM. To complement the non-interactive fixed replay of FD-LHB-CANDOR, FD-LHB-LLM evaluates the model in a closed-loop conversation with an LLM-based spoken partner whose responses adapt to the evaluated model's output. At each turn, Gemini-2.5-Flash (Comanici et al., 2025) generates the partner's next text response from the recent dialogue history, and a local `facebook/mms-tts-eng` VITS model (Pratap et al., 2023) converts it to 16-kHz speech for the evaluated model. The evaluated model's response is then returned to Gemini, which generates the next partner turn. Each session begins with one of ten everyday-life topics, such as hobbies, travel, family, or work, and the partner responds adaptively to the evaluated model rather than following a fixed script. We run 40 sessions per dialogue model, each for up to 20 minutes or until the model reaches its context capacity.

3.2 Failure Modes and Onset Metrics

As shown in Fig. 1 and Fig. 2, we observe three main forms of long-horizon collapse: **repetition**, **silence**, and **gibberish**. To measure when each form first appears, we define its *onset time*. Repetition is evaluated over 32-word windows, while silence and gibberish are evaluated in one-minute bins. For each form, the onset is the start of the first window or bin that meets its criterion.

Repetition. We measure repetition with word-level Seq-Rep-4 (Xu et al., 2022), which captures repeated four-word patterns in the model's recent output. We use this metric because it detects recurring phrase-level patterns. We compute it over 32-word windows and define a window as repetitive when Seq-Rep-4 ≥ 0.30; examples around this threshold are shown in Table 3. The repetition onset is the start of the first of two consecutive repetitive windows, so a single repeated phrase is not counted.

Silence. We measure silence using the number of spoken words in each bin. A bin is considered silent when it contains fewer than three words. To exclude short conversational pauses, we define prolonged-silence onset as the first bin of two consecutive silent bins.

Gibberish. We characterize gibberish using two measures: a Gemini-2.5-Flash judge and predictive entropy, computed from the model's distribution over its next text token. Gemini classifies each one-minute speaking bin as either gibberish or non-gibberish, while predictive entropy measures the model's uncertainty. We evaluate bins containing at least three words and define the gibberish onset as the first bin classified as gibberish by Gemini. A blinded listening study with eight raters shows that the judge never calls normal speech gibberish but sometimes misses real gibberish (Appendix A.3), so the reported gibberish rates may be slightly lower than the true rates. We do not use external-LM perplexity; Appendix A.5 explains why.

4 Diagnosing Long-Conversation Collapse

Based on FD-LHB (Sec. 3), we next investigate the mechanisms behind the three forms of collapse observed in Fig. 2: repetition, silence, and gibberish.

4.1 Initialization Anchor Eviction

The similar timing of silence and gibberish in Moshi (Défossez et al., 2024) and PersonaPlex (Roy et al., 2026) shown in Fig. 2 motivates us to examine their shared cache-management strategy. Both models use a sliding window that discards the oldest cached entries as new ones arrive. As shown in Fig. 3, training splits each sample into chunks that start from an empty cache, so every chunk begins with an *initialization anchor*: the first KV entry, computed without any earlier cached state. We

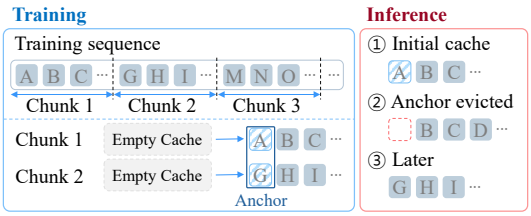


Figure 3: **Initialization-Anchor Eviction.** When training starts each chunk from an empty cache, every chunk begins with a fresh initialization anchor. In sliding-window inference, the cache is never reset, so the initial anchor is eventually evicted.

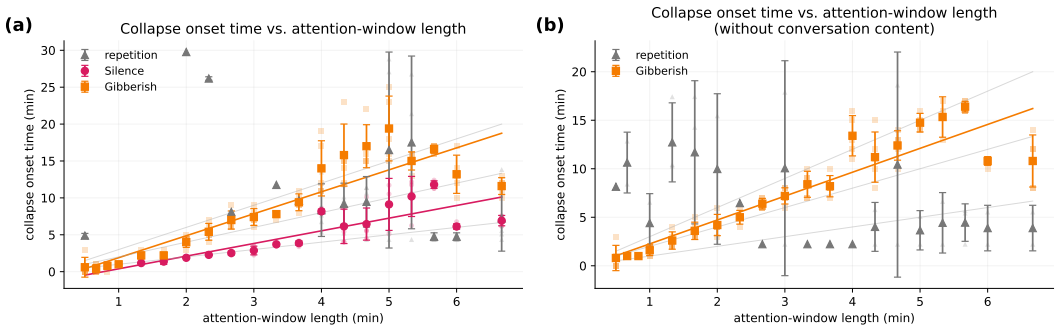


Figure 4: **Silence and gibberish track cache turnover.** Collapse onset time versus attention-window length W. (a) Conversational input. (b) Silence-only input. Silence and gibberish begin later as W grows, roughly in proportion to W.

confirmed this setup in the official Moshi fine-tuning code; whether pretraining and PersonaPlex training follow it is not public (Appendix B.1). At inference, however, the cache is never reset. Once it is full, the oldest entries are evicted, starting with the initialization anchor. Training therefore always contains an anchor, whereas long inference eventually does not, creating a training–inference mismatch. We hypothesize that this initialization-anchor eviction causes the collapse and test this hypothesis in Secs. 4.1.1 and 4.1.2.

4.1.1 Collapse Timing Follows Cache Turnover

Based on the initialization-anchor eviction hypothesis (Sec. 4.1), we vary the attention-window length W and measure when each type of collapse begins (Fig. 4). Results from Moshi and PersonaPlex are combined.

Fig. 4a shows the results with conversational audio input. Silence onset grows at $(1.63\pm0.49)W$ and gibberish onset at $(2.70\pm0.20)W$. Repetition, in contrast, varies widely across runs and shows no clear relation to W. To separate cache turnover from conversational content, Fig. 4b repeats the sweep with silence-only input. Gibberish still appears and grows at $(2.54\pm0.19)W$, whereas repetition does not start later as W grows. These results show that silence and gibberish onsets scale roughly with W, with gibberish following this pattern even in the absence of conversational content, linking these failures to cache turnover, while repetition shows no consistent relationship. We investigate repetition further in Sec. 4.2.

4.1.2 The first KV entry is sufficient

Based on the initialization-anchor eviction hypothesis (Sec. 4.1), we investigate *which* cache entry and *how many* entries must be retained to prevent gibberish. In both experiments below, we measure the percentage of runs that do not produce gibberish over time (Fig. 5). Results from Moshi and PersonaPlex are combined.

In Fig. 5a, we keep exactly one early entry in the cache and vary only which one it is: the 1st, 2nd, 3rd, 4th, 5th, 9th, or 17th entry, or the entry at $L/4$, $L/2$, or $3L/4$, where L is the window length. Only the entry itself changes; its place in the cache and its position index stay the same. Surprisingly, only the first entry prevents gibberish in all runs; every other choice fails in at least one run. In Fig. 5b, we keep the first P consecutive entries, with $P \in \{0, 1, 2, 4, 8, 16, 32, 64, 128\}$. With $P{=}0$, every run reaches gibberish. With any $P{\geq}1$, no run does, and the curves overlap exactly. Together, these results show that the first entry alone prevents gibberish in all tested runs, and keeping more initial entries gives no further improvement.

4.2 Recent Outputs Reinforce Repetition

We next examine whether repetition in full-duplex speech is driven by self-reinforcement. In text generation, repeated content in the context makes the same content more likely, so a loop can reinforce itself (Xu et al., 2022; Zhu et al., 2023). Based on this, we hypothesize that recent outputs

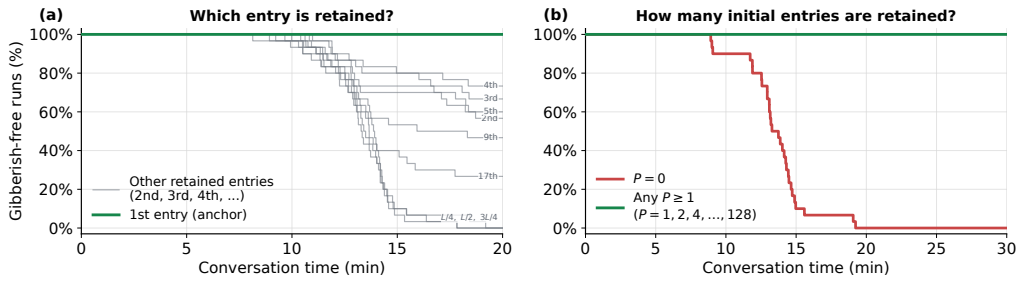


Figure 5: **Retaining the first entry is sufficient.** Percentage of runs without gibberish over time. (a) Which single early entry is kept. (b) Number of initial entries kept, P.

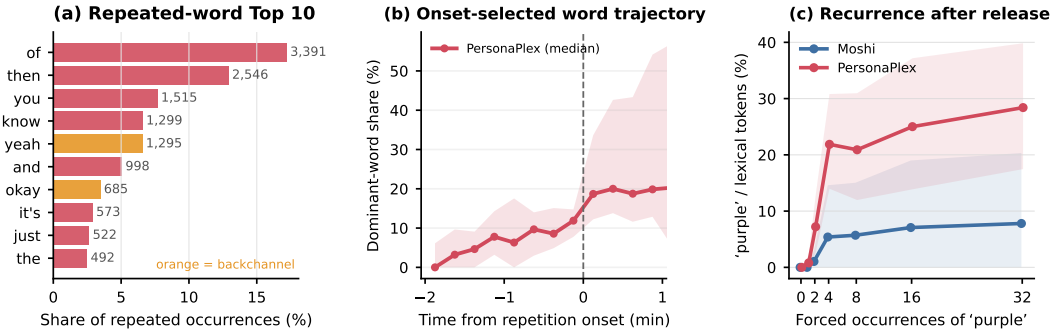


Figure 6: **Recent outputs reinforce repetition.** (a) Top 10 repeated words during repetition periods; orange marks backchannels. (b) Share of each PersonaPlex run's dominant word around repetition onset; seven diagnostic runs without a sustained onset are excluded. (c) Share of *purple* in the 20 s after inserting it d times.

similarly reinforce repetition in full-duplex speech, and test this hypothesis in Fig. 6. First, the most repeated words during repetition periods are short, frequent words such as *of*, *then*, and *you*, and backchannels such as *yeah* and *okay* (Fig. 6a). Second, for each PersonaPlex run we select the word that dominates during repetition periods and track how frequently it occurs. It starts rising before onset and keeps rising afterward (Fig. 6b). This suggests that a repetition loop does not appear suddenly, but grows from a word the model has already been saying. Third, we insert *purple*, a word absent from the previous context, and count how often the model repeats it over the next 20 s. Recurrence increases with more insertions in both models, most strongly in PersonaPlex (Fig. 6c). Together, these results show that recent exposure makes a word recur.

5 Testing the Diagnosis with Training-Free Interventions

The diagnosis yields two testable predictions: retaining the initialization anchor should reduce failures associated with cache eviction, and limiting reinforcement from recent words should reduce repetition. We test these predictions with Initialization-Anchor Pinning (IAP) and Duplex Repetition Penalty (DRP), respectively. Both are training-free methods that operate at inference time and can be applied to existing models.

5.1 Initialization-Anchor Pinning (IAP)

Sec. 4.1.2 showed that keeping the first KV entry is enough to prevent gibberish, while keeping more initial entries adds nothing. IAP therefore keeps this initialization anchor instead of letting the sliding window evict it. Until the cache is full, nothing changes. Once it is full, the model normally drops the oldest entry each time a new one arrives. IAP does the same, except that it never drops the anchor: when a new entry arrives, it removes the oldest entry after the anchor instead. This is done in every layer. As in StreamingLLM (Xiao et al., 2024b), the anchor's position index stays next to the rest of the window so that position indices remain continuous. StreamingLLM, however,

re-indexes the entire cache as the window slides. IAP instead simply increases the anchor's index by one each time the window slides. Only this single entry changes, which keeps IAP cheap (Sec. 6.4).

5.2 Duplex Repetition Penalty (DRP)

DRP reduces repeated words without letting the model fall silent instead. It changes only the assistant's text-token logits; the audio streams are unchanged.

DRP penalizes any word that appeared in the model's last $W{=}512$ text outputs, using the standard repetition penalty of CTRL (Keskar et al., 2019): a positive logit is divided by ρ and a negative one is multiplied by it. A word is penalized based only on whether it appeared recently, not on how often. `PAD` and `EPAD`, which the model uses for listening and turn-taking (Roy et al., 2026; Défossez et al., 2024), are excluded from this penalty. Penalizing words lowers their total probability and makes `PAD` and `EPAD` relatively more likely. To undo this, we raise all word logits by the same amount, so that the total probability of words stays the same as before the penalty. This removes the relative boost that the penalty gives to PAD and EPAD.

With this penalty alone, repetition decreases, but `PAD` can still dominate for long periods, leaving the model silent. To address this, when the model has been silent for a long time and the user is quiet, we gradually lower the `PAD` logit. We do not force any token; the model still samples its next token as usual (Appendix C.2).

6 Results and Analysis

6.1 Experimental Setup

We evaluate five full-duplex models using two context-management strategies: sliding-window and growing-context. Moshi (Défossez et al., 2024) and PersonaPlex (Roy et al., 2026) are sliding-window models with a 3,000-frame window at 12.5 Hz, so their window first fills after four minutes. Raon-SpeechChat (Kim et al., 2026), BayLing-Duplex (Fang et al., 2026), and Lychee-FD (Liu et al., 2026) are growing-context models that run until their context is full, after about 5, 17.5, and 10.9 minutes, respectively. Reaching context capacity ends the run; this limits sustained operation but is not counted as a collapse event. In Fig. 7, such runs are right-censored: they count as failure-free until they end, and the curves are estimated with the Kaplan–Meier method (Kaplan & Meier, 1958). We apply IAP+DRP to the two sliding-window models, since anchor eviction occurs only when a sliding window discards old entries. Generation settings for each model are given in Table 9.

6.2 Main Results

Fig. 7 shows how long each system stays free of each collapse on FD-LHB-CANDOR (top) and FD-LHB-LLM (bottom); "Ours" denotes IAP+DRP. Observation lengths differ across models, so we compare methods within each model rather than across models. All percentages below are computed over 40 runs per system.

Every baseline fails, in one of two ways. On FD-LHB-CANDOR (Fig. 7, top), growing-context models run out of context or collapse first: Raon-SpeechChat reaches capacity after about five minutes, BayLing-Duplex shows repetition, silence, and gibberish in 55%, 100%, and 90% of runs within 17.5 minutes, and Lychee-FD shows repetition and gibberish in 37.5% and 30% within 10.9 minutes. Sliding-window models never exhaust their context but still collapse: over 60 minutes, Moshi and PersonaPlex show the three failures in 65/100/100% and 90/85/100% of runs. FD-LHB-LLM shows the same pattern (Fig. 7, bottom). Thus, none of the evaluated models sustains stable conversation over tens of minutes, regardless of how it manages its context.

IAP+DRP reduces collapse. We apply IAP+DRP to the two sliding-window models. On FD-LHB-CANDOR, the repetition, silence, and gibberish rates drop to 0%, 7.5%, and 0% for Moshi, and 2.5%, 47.5%, and 0% for PersonaPlex. The PersonaPlex+Ours silence in Fig. 7b is mostly temporary: 94.7% of the affected runs resume speaking within a minute after the two-minute silence. On FD-LHB-LLM, where the partner responds to the model, the baselines also fail frequently, most often through gibberish (97.5–100% of sessions) and prolonged silence (70–87.5%). With IAP+DRP,

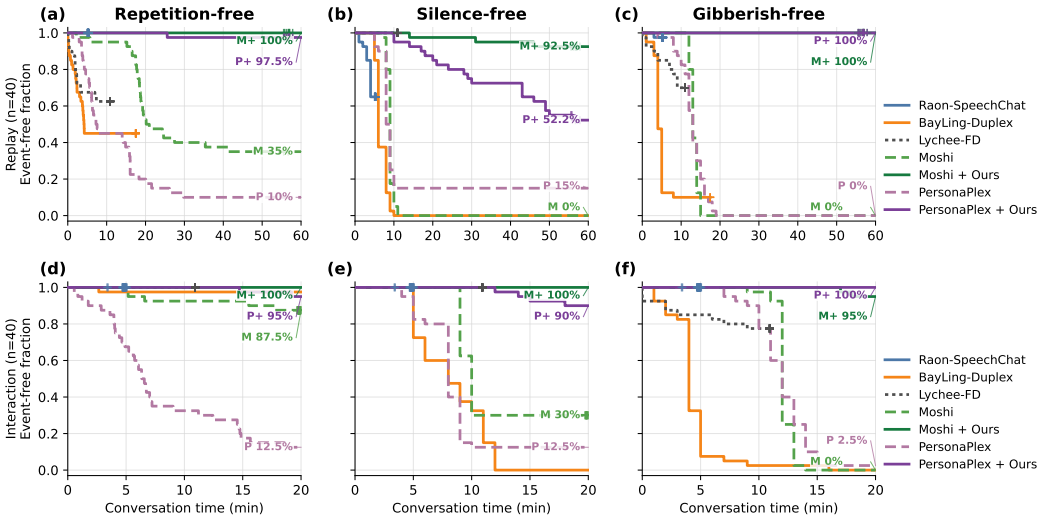


Figure 7: **Repetition, silence, and gibberish across replay and adaptive interaction.** Top: FD-LHB-CANDOR; bottom: FD-LHB-LLM. Curves show the percentage of runs remaining free of repetition (a, d), prolonged silence (b, e), or gibberish (c, f) over time; higher is better. For growing-context models, a plus marks where the model reaches its context capacity. IAP+DRP largely suppresses repetition and gibberish in both models and reduces prolonged silence.

each failure occurs in no more than 10% of sessions for both models. Table 10 lists all counts and onset times. Thus, without retraining, IAP+DRP substantially reduces all three forms of collapse in both sliding-window models, in both controlled replay and adaptive interaction.

IAP+DRP also preserves response quality. Avoiding collapse does not guarantee the quality of what the model says. We therefore also score the content of each one-minute speaking bin on FD-LHB-LLM, using Gemini-2.5-Flash as a judge (Fig. 8). It rates two aspects on a 1–5 scale: *Coherence*, whether the model's speech is fluent and makes sense, and *Context*, whether it responds appropriately to what the partner said. Silent bins receive a score of 1. Both baselines fall to the lowest score within about ten minutes, whereas IAP+DRP keeps both scores steady for the full session. Appendix D.2 reports the same scores for FD-LHB-CANDOR, and Appendix D.4 for the ablation. Thus, IAP+DRP avoids collapse while keeping the model's speech coherent and responsive.

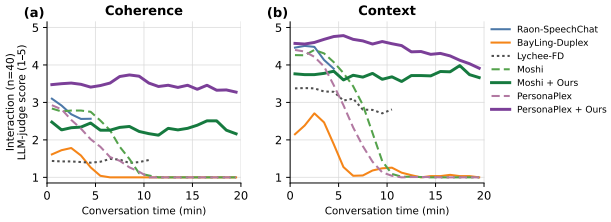


Figure 8: **LLM-judge scores over time.** Mean Gemini-2.5-Flash ratings on FD-LHB-LLM for (a) coherence and (b) context to the partner's preceding speech.

6.3 Ablation

To investigate the effect of each component, we apply IAP and DRP separately and together to Moshi on the FD-LHB-CANDOR replays (Fig. 9). For repetition in Fig. 9a, DRP keeps every run repetition-free, and IAP also raises the repetition-free share from 35% to 90%, suggesting that initialization-anchor eviction contributes to repetition in part. For silence in Fig. 9b, IAP raises the silence-free share from 0% to 67.5%, and adding DRP raises it further to 92.5%. For gibberish in Fig. 9c, IAP alone keeps every run free of Gemini-defined gibberish, whereas DRP alone does not help, supporting the link between gibberish and initialization-anchor eviction. Thus, the two com-

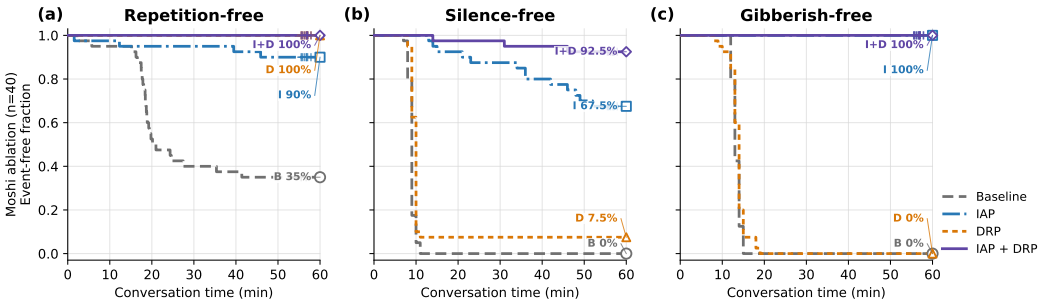


Figure 9: **Ablation of cache retention and repetition control.** Curves show the proportion remaining free of (a) repetition, (b) prolonged silence, and (c) gibberish. Endpoint labels identify Baseline (B), IAP (I), DRP (D), and IAP+DRP (I+D).

ponents are complementary: IAP addresses gibberish and most silence, DRP addresses repetition, and only their combination suppresses all three failures.

6.4 Real-Time Inference Cost

Full-duplex models must respond in real time, so any extra computation matters. We therefore measure the cost of IAP and DRP (Table 1). The table reports two numbers for PersonaPlex: how many of 40 runs stay free of gibberish, and the real-time factor (RTF). RTF is the time, in seconds, needed to process one second of audio. An RTF below one means the model runs faster than real time. We measure RTF five times on an RTX A6000.

Table 1: **Real-time inference cost** Gibberish-free runs (out of 40) and real-time factor (RTF).

Method	Gibberish-free ↑	RTF ↓
Baseline	0/40	0.5957±0.0020
StreamingLLM-style	40/40	1.0323±0.0029
Ours	40/40	0.7525±0.0023

We compare three methods. Baseline uses the standard FIFO cache. StreamingLLM-style (Xiao et al., 2024b) keeps the first four entries and re-indexes all remaining entries at every step. It avoids gibberish in every run, but its RTF is 1.03, so it cannot keep up in real time. Ours also avoids gibberish in every run, but reaches an RTF of 0.75, because IAP re-indexes only the first entry. In short, both methods prevent gibberish, but IAP is faster because it updates one entry per step instead of all of them. Appendix D.8 reports the cost of each component.

7 Conclusion

We introduced FD-LHB to evaluate whether full-duplex speech models can sustain long conversations, using controlled replay and adaptive interaction. Across five models, sustained operation was limited by behavioral degradation or context exhaustion. Controlled experiments link silence and gibberish to Initialization-Anchor Eviction, and repetition to self-reinforcement from recent outputs. Guided by these findings, we applied two simple interventions: IAP retains the initialization anchor, while DRP discourages repetition and guards against prolonged silence. In replays of up to an hour, their combination reduces repetition, silence, and gibberish, and it also improves stability in adaptive conversations with an LLM partner. These gains validate our diagnosis. Both methods require neither retraining nor additional memory, and they run in real time. Overall, we identify how and why full-duplex models collapse in long conversations and show that simple, cause-specific fixes can prevent much of this collapse.

AI use statement

We used Gemini-2.5-Flash to generate partner utterances in FD-LHB-LLM and to provide the automated judgments described in the evaluation protocol. We used OpenAI Codex to assist with code development, debugging, and running analysis scripts. We used Anthropic Claude and OpenAI Codex to help draft and revise parts of the manuscript and to help find related work. The authors

reviewed all AI-assisted text, verified the code by running it and checking the resulting logs and outputs, and take responsibility for the final content, claims, and artifacts.

Ethics Statement

Our listening studies (Appendices A.3 and D.7) were minimal-risk perceptual evaluations. Participants were adult, English-proficient crowdworkers recruited through Prolific, who provided informed consent before participating and could withdraw at any time. They listened to short clips of model-generated speech and answered forced-choice questions; no personally identifying information was collected beyond Prolific's anonymized IDs. Participants were compensated at £9.00/hour, in line with Prolific's recommended rate. The benchmark uses the CANDOR corpus, whose participants consented to recording and research use at collection time (Reece et al., 2023); we access it under its research-access terms and release only a manifest of conversation IDs and crop boundaries, not the audio.

References

Siddhant Arora, Zhiyun Lu, Chung-Cheng Chiu, Ruoming Pang, and Shinji Watanabe. Talking turns: Benchmarking audio foundation models on turn-taking dynamics. *arXiv preprint arXiv:2503.01174*, 2025.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. VALL-E 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *arXiv preprint arXiv:2406.05370*, 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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.

Wenjie Fan, Bin Ma, and Dong Li. Bayesian repetition penalty: A principled adjacent-conditional framework for reversing attention collapse in autoregressive language models. *arXiv preprint arXiv:2607.22694*, 2026.

Qingkai Fang, Yan Zhou, Shoutao Guo, Shaolei Zhang, and Yang Feng. LLaMA-Omni 2: LLM-based real-time spoken chatbot with autoregressive streaming speech synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18617–18629, 2025.

Qingkai Fang, Shoutao Guo, and Yang Feng. BayLing-Duplex: Native full-duplex speech dialogue with a single autoregressive LLM. *arXiv preprint arXiv:2606.14528*, 2026.

Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic VQGAN and time-sensitive transformer. In *European conference on computer vision*, pp. 102–118. Springer, 2022.

Antonio A. Ginart, Naveen Kodali, Jason Lee, Caiming Xiong, Silvio Savarese, and John R. Emmons. LZ Penalty: An information-theoretic repetition penalty for autoregressive language models. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=vNzPB4YCHj.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models, 2024. URL https://arxiv.org/abs/2308.16137.

Zhang He, Wenqian Cui, Haoning Xu, Xiao-Hui Li, Lei Zhu, Haoli Bai, Ma Shaohua, and Irwin King. MTR-DuplexBench: Towards a comprehensive evaluation of multi-round conversations for full-duplex speech language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 5334–5351, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.263. URL https://aclanthology.org/2026.findings-acl.263/.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

Beomsoo Kim, Changho Choi, Dohyun Kim, Dongki Lee, Ethan Ewer, Eunchong Kim, Gyeongman Kim, Haechan Kim, Hyeonghwan Kim, Inkyu Park, et al. Raon-speech technical report. *arXiv preprint arXiv:2605.23912*, 2026.

KimiTeam, Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, Zhengtao Wang, Chu Wei, Yifei Xin, Xinran Xu, Jianwei Yu, Yutao Zhang, Xinyu Zhou, Y. Charles, Jun Chen, Yanru Chen, Yulun Du, Weiran He, Zhenxing Hu, Guokun Lai, Qingcheng Li, Yangyang Liu, Weidong Sun, Jianzhou Wang, Yuzhi Wang, Yuefeng Wu, Yuxin Wu, Dongchao Yang, Hao Yang, Ying Yang, Zhilin Yang, Aoxiong Yin, Ruibin Yuan, Yutong Zhang, and Zaida Zhou. Kimi-audio technical report, 2025. URL https://arxiv.org/abs/2504.18425.

Kyutai. Moshi-finetune. https://github.com/kyutai-labs/moshi-finetune, 2025.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, and Emmanuel Dupoux. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354, 2021. doi: 10.1162/tacl_a_00430. URL https://aclanthology.org/2021.tacl-1.79/.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *Advances in neural information processing systems*, 29, 2016.

Stephen C Levinson and Francisco Torreira. Timing in turn-taking and its implications for processing models of language. *Frontiers in psychology*, 6:731, 2015.

Tianpeng Li, Jun Liu, Tao Zhang, Yuanbo Fang, Da Pan, Mingrui Wang, Zheng Liang, Zehuan Li, Mingan Lin, Guosheng Dong, et al. Baichuan-Audio: A unified framework for end-to-end speech interaction. *arXiv preprint arXiv:2502.17239*, 2025.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

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. *arXiv preprint arXiv:2503.04721*, 2025.

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

Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, Shinji Watanabe, and Hung-yi Lee. Full-Duplex-Bench v1.5: Evaluating overlap handling for full-duplex speech models. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 19447–19451. IEEE, 2026b.

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, 2026. doi: 10.18653/v1/2026.acl-long.419.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Se Jin Park, Julian Salazar, Aren Jansen, Keisuke Kinoshita, Yong Man Ro, and RJ Skerry-Ryan. Long-form speech generation with spoken language models. *arXiv preprint arXiv:2412.18603*, 2024.

Yizhou Peng, Yi-Wen Chao, Dianwen Ng, Yukun Ma, Chongjia Ni, Bin Ma, and Eng Siong Chng. FD-Bench: A full-duplex benchmarking pipeline designed for full duplex spoken dialogue systems. *arXiv preprint arXiv:2507.19040*, 2025.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI technical report*, 2019.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL https://arxiv.org/abs/2212.04356.

Soham Ray, Keshav Dhandhania, Victor Barres, and Karthik R Narasimhan. τ–voice: Benchmarking full-duplex voice agents on real-world domains. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=2Oj6fg0m1j.

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

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. *arXiv preprint arXiv:2602.06053*, 2026.

Yakun Song, Zhuo Chen, Xiaofei Wang, Ziyang Ma, and Xie Chen. ELLA-V: Stable neural codec language modeling with alignment-guided sequence reordering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 25174–25182, 2025.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

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

Chengyou Wang, Hongfei Xue, Guojian Li, Zhixian Zhao, Shuiyuan Wang, Shuai Wang, Xin Xu, Hui Bu, and Lei Xie. Full-duplex interaction in spoken dialogue systems: A comprehensive study from the ICASSP 2026 HumDial challenge. *arXiv preprint arXiv:2604.21406*, 2026a.

Wenfu Wang, Chenxing Li, Liqiang Zhang, Yiyang Zhao, Yuxiang Zou, Hanzhao Li, Mingyu Cui, Hao Zhang, Kun Wei, Le Xu, et al. Covo-Audio technical report. *arXiv preprint arXiv:2602.09823*, 2026b.

Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, et al. Step-Audio 2 technical report. *arXiv preprint arXiv:2507.16632*, 2025.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. Infllm: Training-free long-context extrapolation for llms with an efficient context memory, 2024a. URL https://arxiv.org/abs/2402.04617.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024b.

Zhifei Xie and Changqiao Wu. Mini-Omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024a.

Zhifei Xie and Changqiao Wu. Mini-Omni2: Towards open-source GPT-4o with vision, speech and duplex capabilities. *arXiv preprint arXiv:2410.11190*, 2024b.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *Advances in Neural Information Processing Systems*, 35:3082–3095, 2022.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni technical report, 2025. URL https://arxiv.org/abs/2503.20215.

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

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

Wenhong Zhu, Hongkun Hao, and Rui Wang. Penalty decoding: Well suppress the self-reinforcement effect in open-ended text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1218–1228, 2023.

APPENDIX ROADMAP

Table 2 points to the supporting evidence for each part of the main text. Appendix A validates the evaluation, Appendix B analyzes the two collapse mechanisms, Appendix C specifies IAP and DRP, Appendix D reports extended results, ablations, and computational cost, and Appendix E collects exploratory observations and limitations.

The mechanism analyses in Appendix B.1 measure stability by predictive entropy; all benchmark results use the Gemini gibberish label.

Table 2: Supporting evidence for each part of the main text.

Main-text claim	Appendix	Evidence
Gemini gibberish labels are reliable (Sec. 3.2)	Appendix A.3	Blinded human listening study
Text criteria reflect audible failures (Sec. 3.2)	Appendix A.4	Loudness-based silence and ASR-based repetition
Only the first KV entry matters (Sec. 4.1.2)	Appendix B.1	Keep a single early KV entry or a prefix of entries; position-index and cache-history controls
Recent outputs reinforce repetition (Sec. 4.2)	Appendix B.2	Insert a new word and measure whether the model repeats it
IAP and DRP (Sec. 5)	Appendix C	Update rules, execution order, and final hyperparameters
IAP+DRP reduces collapse (Sec. 6.2)	Appendices D.1 to D.3	Onset times, significance tests, quality scores, and silence durations
Each component contributes (Sec. 6.3)	Appendices D.4 and D.6	Component ablations and alternative repetition controls
Effect of DRP on speech naturalness	Appendix D.7	Blinded A/B listening study early, middle, and late in the conversation
IAP+DRP runs in real time (Sec. 6.4)	Appendix D.8	Per-frame latency and real-time factor
Beyond full-duplex dialogue (exploratory)	Appendix E.1	Additional speech models

A Evaluation Setup and Validation

This section describes the benchmark inputs and model settings (Appendix A.1), the metrics and judges (Appendix A.2), a human validation of the Gemini gibberish judge (Appendix A.3), audio-side checks of the collapse metrics (Appendix A.4), why we use predictive entropy instead of perplexity (Appendix A.5), the sensitivity of repetition and gibberish criteria (Appendix A.6), the Gemini prompts (Appendix A.7), and how to reproduce the results (Appendix A.8). All later analyses use these definitions.

A.1 Benchmark Inputs and Model Settings

Replay input. As described in Sec. 3.1, we remove the first three turns of each CANDOR conversation, keep the 40 longest, and use the speaker channel with more speech as input. Every model and condition receives exactly the same 16-kHz mono audio for a given conversation.

Interactive partner. The partner uses Gemini 2.5 Flash (Comanici et al., 2025) for text and a `facebook/mms-tts-eng` VITS model (Pratap et al., 2023) for 16-kHz speech. It sees the last 16 turns and replies in one or two casual sentences (temperature 0.9), asking natural follow-up questions and reacting naturally when the model's speech is unclear or repetitive. Each session opens with one of ten everyday topics: hobbies, pets, food, travel, entertainment, work, music, family, exercise, or reading. Each model has 40 sessions, each lasting up to 20 minutes or until the model reaches its context limit. The complete partner prompt is given in Appendix A.7.

Model prompts and sampling. PersonaPlex is given the persona "You enjoy having a good conversation. Have a warm, casual chat about everyday life, family, and pets. Your name is Alex." Moshi receives no text persona. For each model, Baseline and Ours use the same prompt, input, and sampling settings, so the two differ only in IAP and DRP. Full settings are listed in Table 9.

Compute. All experiments run on NVIDIA RTX A6000 GPUs with bfloat16 weights.

A.2 METRICS, JUDGES, AND AGGREGATION

We measure three failure types (repetition, silence, and gibberish) and two quality scores (Coherence and Context). Each failure is summarized by its onset time, defined below.

Repetition. We compute Seq-Rep-4 over rolling 32-word windows:

$$\text{Seq-Rep-4} = 1 - \frac{\#\text{ unique word 4-grams}}{\#\text{ word 4-grams}}.$$

Windows advance by 16 words and cover at most 30 seconds of audio. A window is repetitive if Seq-Rep-4 ≥ 0.30. Repetition onset is the start of the first of two consecutive repetitive windows, so a brief repeat that makes only one window repetitive is not counted. Table 3 shows actual outputs at different scores, and Table 8 reports results under other thresholds, window sizes, and a single-window rule.

Table 3: **Example outputs at different Seq-Rep-4 levels.** Each row shows the 32-word window from PersonaPlex replays whose score is closest to the target level. Below 0.30, a phrase recurs at most twice; at 0.30, the same phrase already recurs three times within the window. We therefore use 0.30 as the repetition threshold (highlighted). Dup. is the number of repeated 4-grams among the 29 in the window.

Target	Observed	Dup.	Decoded model output spanning 32 lexical words
0.1	0.103	3/29	"computer at work, and they keep telling me, oh it' got to be a laptop. It' got to be a laptop. And, you know, I don' know that it actually does because"
0.2	0.207	6/29	"that'd be good. But I know here, like, if you leave it with a bank, yeah, they charge you. Yeah, I think here, like, if you leave it with a bank, they"
0.3	0.310	9/29	"know. And **you can just kind of get away**. **You can just kind of get away from** the Midwest. **You can just kind of get away from** that, you know, because it"
0.4	0.414	12/29	"education that people get, you know, the better, you know, the better people the country will be. And, you know, the better, you know, the better, you know, the better, you know"
0.5	0.517	15/29	"it's just you know it's just you know it's just you know it's just you know it's just you it's just you know it's you know it justs you know it's just"
0.6	0.586	17/29	"different group of people. And then the next day, you know it's, you know, it's, you know it's, you know, it's, you know, it's, you know, it's, you know, it's, you know"
0.7	0.690	20/29	"it justs you know it justs you know it's just you know it's just you know it's just you know it's just you know it's just you know"
0.8	0.793	23/29	"moving. Okay, so anyway, so I moving. Okay, so anyway, so I moving. Okay, so anyway, so I moving. Okay, so anyway, so I moving. Okay, so anyway, so I moving. Okay"
0.9	0.897	26/29	"Yeah she's spoiled. Yeah, she's spoiled. Yeah, she's spoiled. Yeah, she's spoiled. Yeah, she's spoiled. Yeah, she's spoiled. Yeah, she's spoiled. Yeah, she's spoiled. Yeah, she's spoiled. Yeah, she's"
1.0	0.966	28/29	"Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah. Yeah Yeah. Yeah Yeah. Yeah Yeah Yeah Yeah Yeah Yeah Yeah Yeah. Yeah Yeah Yeah"

Silence. We split each run into one-minute bins and call a bin *silence* if it contains fewer than three words. Prolonged-silence onset is the first of two consecutive silence bins, so brief pauses are not counted.

Gibberish. We measure gibberish in two ways. For benchmark results, Gemini-2.5-Flash (temperature 0) labels each one-minute bin with at least three words as gibberish or not. This label uses only the output text, so it applies to every model. For the mechanism analyses (Appendix B), we also use the model's *predictive entropy* $H_t = -\sum_i p_t(i) \log p_t(i)$, where p_t is its distribution over the next text token at frame t. High entropy means the model is uncertain about what to say next. Because it requires the model's logits, which not every evaluated model exposes, we do not use it for benchmark results. A bin is labeled gibberish if its text is largely uninterpretable English, such as word salad, broken token fragments, or severely malformed syntax. Repetition, disfluency, generic or off-topic replies, and factual errors do not count as gibberish on their own. Gibberish onset is the first bin labeled gibberish. The exact classification prompt is given in Appendix A.7.

Coherence and Context. Two further Gemini calls score each speaking bin from 1 to 5. Coherence rates whether the speech is fluent and makes sense; Context rates whether it responds appropriately to the partner, so this call also sees the partner's recent speech. Silence bins receive a score of 1, so silence lowers the average instead of dropping out of it. The two scoring prompts are given in Appendix A.7.

Aggregation. Tables 10 and 12 report how many runs reach each failure and, among those runs, the mean±SD onset time. Time-to-event curves(Figs. 7 and 9) show the fraction of runs that have not yet failed, estimated with the Kaplan–Meier method (Kaplan & Meier, 1958). Some runs end early because the model reaches its context limit or the input audio ends. Such a run is *right-censored*: we cannot know whether it would have failed later, so it counts as failure-free while it is running and is left out of the calculation after it ends. As a result, a curve's final value can differ slightly from the raw fraction of failure-free runs in the tables. NR (not reached) means the event did not occur before the run ended.

A.3 Human Validation of the Gemini Gibberish Judge

We checked the Gemini gibberish label against human listeners. Ten proficient-English listeners were recruited through Prolific, gave informed consent, and were paid for their time; two failed an attention check and were excluded, leaving eight. Each rated 40 one-minute clips from Moshi and PersonaPlex, half labeled gibberish by Gemini and half not. Listeners did not see the model, condition, transcript, or Gemini label. A clip is gibberish for humans if at least five of eight listeners say so.

Gemini agrees with the human majority on 36 of 40 clips (Cohen's $\kappa = 0.80$; 1 means perfect agreement), similar to agreement between listeners (mean pairwise $\kappa = 0.76$). In all four disagreements, humans heard gibberish that Gemini missed; Gemini never called normal speech gibberish. So the real amount of gibberish may be slightly higher than we report (Table 4).

Table 4: **Human validation of the Gemini gibberish label.** Eight listeners rated the same 40 clips; the human majority is the reference.

Measure	Result
Agreement with human majority	36/40 (90%)
Cohen's κ (Gemini vs. humans)	0.80 (95% CI 0.61–0.99)
Mean pairwise κ (human vs. human)	0.76 (28 pairs)
Precision (Gemini gibberish is correct)	20/20 (1.00)
Recall (human gibberish found by Gemini)	20/24 (0.83)

Two caveats apply. First, humans listened to audio, while Gemini read the transcript. Second, we picked the clips to be exactly half gibberish, but real benchmark data does not have this even split, so the 90% agreement may not hold exactly on the full benchmark. Overall, the study shows that the Gemini label is a reasonable proxy for human judgment.

A.4 Checking the Collapse Metrics on Audio

Our collapse metrics use the model's text output, because every model produces text and this makes the metrics easy to reproduce. Here we check, on saved audio, that silence and repetition can also be heard in the speech (Table 5). Gibberish is already checked by human listeners in Appendix A.3.

Table 5: **Checking the collapse metrics on audio.** For each failure, we check whether the text-based metric agrees with the audio.

Failure	How we check it on audio	What we find
Silence	Measure how loud each minute is	Loudness and word count agree on which minutes are silent (AUROC 0.983)
Repetition	Transcribe the audio with Whisper and recompute repetition	Runs repetitive in text are also repetitive in audio; IAP+DRP also reduces repetition in the audio

Silence. We split each one-minute bin into 20-ms frames and count a frame as active if it is louder than -45 dBFS, a level close to silence. A bin counts as silent if its share of active frames is below a cutoff. We use half of the runs to choose this cutoff and the other half to test it. We then check whether these silent bins match the bins that our word-count definition calls silent. The two

definitions agree closely. They label almost the same minutes as silent (AUROC 0.983). Their silence onsets differ by only 0.67 minutes on average. Both definitions also find the same runs with prolonged silence.

Repetition. The model's text output may not exactly match what it says aloud, so we also measure repetition from the audio. We transcribe all 160 replay runs of Moshi and PersonaPlex (Baseline and IAP+DRP) with two ASR models, Whisper `small.en` and `large-v3` (Radford et al., 2022), and apply the same repetition criterion to each transcript (Table 6). When the text shows repetition, the audio almost always does too (precision 1.00 with `small.en` and 0.98 with `large-v3`). The audio also shows repetition in about 30% more runs than the text does (recall 0.72 and 0.68), so the text-based metric tends to undercount repetition. Even when repetition is measured from the audio, IAP+DRP still greatly reduces it compared with the baselines.

Table 6: **Repetition detected from text and from audio.** Number of replay runs (out of 40) that reach repetition, using the model's text output or ASR transcripts of its audio.

Model	Method	Text output	Whisper `small.en`	Whisper `large-v3`
Moshi	Baseline	26	40	40
Moshi	IAP+DRP	0	2	8
PersonaPlex	Baseline	36	40	40
PersonaPlex	IAP+DRP	1	5	3

A.5 Why Predictive Entropy Instead of Perplexity

A common way to score generated speech is the perplexity (PPL) of its transcript under an external language model (Lakhotia et al., 2021), and full-duplex models, including Moshi, have been evaluated this way (Défossez et al., 2024; Veluri et al., 2024). However, PPL measures how predictable text is, not whether it makes sense. A repetitive loop is highly predictable and gets low PPL (Holtzman et al., 2019; Park et al., 2024), while unusual but readable speech can get high PPL.

Table 7 shows examples where readable speech gets *higher* PPL than gibberish under three external LMs: GPT-2-large (Radford et al., 2019), Llama-3.1-8B (Grattafiori et al., 2024), and Mistral-7B-v0.3 (Jiang et al., 2023). Fig. 10 shows this over 2,750 speaking bins. Each point is one bin, placed by its PPL (x-axis) and its predictive entropy (y-axis). Along the x-axis, gibberish (red) and non-gibberish (green) bins cover overlapping PPL ranges, so no single PPL cutoff separates them. Some gibberish bins even get lower PPL than typical readable speech, as the two marked examples from Table 7 show. Along the y-axis, the two groups barely overlap: non-gibberish bins stay near zero entropy, while most gibberish bins lie far above. The same pattern holds for all three external LMs. We therefore use the Gemini label for benchmark results and predictive entropy for mechanism analyses. This does not mean PPL is useless in general, only that it does not detect gibberish here.

A.6 Sensitivity to Metric Thresholds

Repetition. Our repetition criterion involves three choices: the Seq-Rep-4 threshold (0.30), the window size (32 words), and the persistence rule (two consecutive windows). To check that our conclusions do not hinge on these choices, we recompute repetition on the 20 longest replays while varying each one: thresholds of 0.20, 0.30, and 0.40; windows of 16, 32, and 64 words; and a single-window rule that flags a run as soon as one window crosses the threshold. Table 8 reports how many of the 20 runs are flagged in each setting. Looser settings flag more runs, as expected. In every setting, however, the baselines are flagged in at least 9 of 20 runs, whereas IAP+DRP is flagged in no run under the two-window rule and in at most 3 runs under the single-window rule. The gap between the baselines and IAP+DRP therefore does not depend on these choices.

Gibberish. Unlike repetition and silence, gibberish onset requires only a single bin labeled gibberish. To check whether this choice matters, we also apply a stricter rule that requires three consecutive speaking bins labeled gibberish. For the two sliding-window models, the comparison between the baseline and IAP+DRP is unchanged. In replay, both baselines still reach gibberish in 40 of 40 runs and IAP+DRP in none. In interaction, the baselines drop only slightly (Moshi from 40 to 38 runs,

Table 7: **Readable speech can get higher PPL than gibberish.** Eight example bins from Moshi and PersonaPlex. Under all three external LMs, the readable examples get higher PPL than the gibberish ones, because gibberish here often consists of short, repeated fragments that are easy to predict. Predictive entropy H and the Gemini label G (1: gibberish) separate them correctly. Excerpts are shortened; Fig. 10 shows all bins.

Type	Model	GPT-2 PPL	Llama PPL	Mistral PPL	H	G	Assistant-text excerpt
Readable	Moshi	68.9	45.0	42.4	0.07	0	"Hello, what's up? I was just talking to my mom earlier, and I got this really good idea. How about yours?"
	Moshi	36.8	30.5	19.1	0.11	0	"I caught up on some reading. That sounds like a good book. I should check it out."
	PersonaPlex	86.0	47.6	35.5	0.16	0	"Hello. My name's Angela. I'm in Ohio. Are you near Orlando? What is your opinion?"
	PersonaPlex	53.5	36.0	22.3	0.12	0	"My week's been okay. It's been busy. How about yours? I'm finding out as a teacher, it's true."
Gibberish	Moshi	4.5	3.3	3.6	3.81	1	"then a a a a a a a then a a a … worldend a a a then a grand …"
	Moshi	5.2	4.2	4.7	6.54	1	"a a– at the Under a a a again … a behind a … a a a a a …"
	PersonaPlex	3.5	3.1	3.3	3.42	1	"of many of of of of a … of four of of all … c c of …"
	PersonaPlex	5.0	3.7	3.9	4.39	1	"of these large of these probably of of … Johnson … of of of next …"

Table 8: **Repetition sensitivity on the 20 longest replays.** Each cell gives the number of runs (out of 20) flagged as repetitive at thresholds 0.20/0.30/0.40. Windows advance by half their length and cover at most 30 seconds. Our main setting is two consecutive 32-word windows at 0.30 (the middle number in the 32-word column of the upper block). The subset was fixed before examining outcomes.

Rule	Model	Method	16 words	32 words	64 words
two consecutive	Moshi	Baseline	19/14/9	17/14/11	18/12/10
two consecutive	Moshi	IAP+DRP	0/0/0	0/0/0	0/0/0
two consecutive	PersonaPlex	Baseline	20/19/12	20/18/14	19/17/13
two consecutive	PersonaPlex	IAP+DRP	0/0/0	0/0/0	0/0/0
single window	Moshi	Baseline	20/19/15	20/18/13	18/14/10
single window	Moshi	IAP+DRP	0/0/0	0/0/0	0/0/0
single window	PersonaPlex	Baseline	20/20/19	20/20/17	20/19/17
single window	PersonaPlex	IAP+DRP	3/2/0	1/0/0	0/0/0

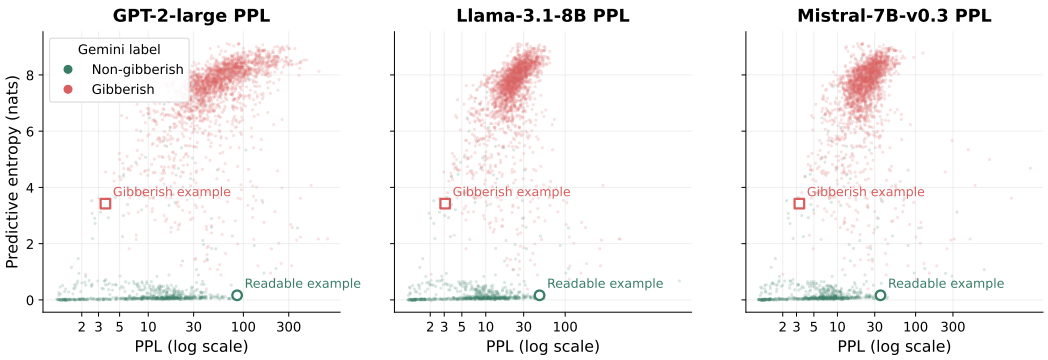


Figure 10: **PPL does not separate gibberish, while predictive entropy does.** Each point is a speaking bin, colored by the Gemini label (red: gibberish, green: non-gibberish). Along the x-axis (PPL, log scale), the two colors overlap; along the y-axis (predictive entropy), they separate clearly. Marked points are examples from Table 7.

PersonaPlex from 39 to 35), while IAP+DRP goes from 2 to 1 run for Moshi and stays at 0 for PersonaPlex. The stricter rule matters for the growing-context models. BayLing-Duplex often falls silent right after its first gibberish bin, so its count drops from 36 to 3 runs in replay and from 40 to 11 in interaction; its collapse then shows up as silence instead. Lychee-FD never produces three gibberish bins in a row, so its count drops to 0 in both tracks. A three-bin rule would therefore hide most gibberish in these models, and we use the single-bin rule.

A.7 Gemini Prompts

The following are the prompt templates used for the Gemini-based measures and the interactive partner. Braced fields denote text inserted at run time. The three evaluation prompts use Gemini 2.5 Flash at temperature 0. The partner uses the same model at temperature 0.9, receives the last 16 dialogue turns, and generates at most 90 tokens per turn. Judge inputs are JSON-encoded; the assistant text is truncated to 1,500 characters and, for Context, the user text to 1,200 characters.

Binary gibberish classification.

```
Classify whether the assistant text is linguistic gibberish.

GIBBERISH = 1:
The output is largely uninterpretable as English because of word salad,
broken token fragments, severely malformed syntax, or nonsensical word
composition.

GIBBERISH = 0:
The output remains understandable English, even if it is repetitive,
disfluent, generic, factually incorrect, or unrelated to the user.

Judge only the assistant text. Do not evaluate relevance. Do not classify
understandable repetition alone as gibberish.

ASSISTANT TEXT: {assistant_text_json}
Return ONLY JSON: {"gibberish": 0 or 1}
```

Coherence score.

```
Rate ONLY the intrinsic fluency of this one-minute assistant speech segment
on an integer scale from 1 to 5. Read only the assistant text.

5 = fluent, grammatical, and sensible English, even if unrelated to the user.
3 = clunky or somewhat repetitive, but still readable and interpretable.
1 = gibberish, word salad, broken fragments, or speech stuck in a looping
pattern.

ASSISTANT: {assistant_text_json}
Return ONLY JSON: {"coherence": n}
```

Context score.

```
Rate ONLY how appropriately this one-minute assistant speech segment responds
to the user's speech from the matched minute, on an integer scale from 1 to 5.

5 = on-topic and relevant; short backchannels can be highly appropriate.
3 = vague or generic, but plausibly responsive.
1 = irrelevant, canned, or ignores the user. Uninterpretable gibberish cannot
be relevant and must score 1.

USER: {user_text_json}
ASSISTANT: {assistant_text_json}
Return ONLY JSON: {"context": n}
```

FD-LHB-LLM partner. The system instruction is:

```
You are a warm, friendly person on a casual phone call with a friend named
Alex. Talk naturally about everyday life -- your day, hobbies, food, weekend
plans, whatever comes up. React genuinely to what Alex says, ask follow-up
questions, share your own little stories. Keep each turn SHORT (1-2
sentences), like real speech. Do NOT quiz them or recite facts -- just a nice
flowing chat. If Alex says something odd, unclear, or repetitive, react like a
real person would (e.g. 'wait, what?', 'you okay?', gently change topic). This
particular call naturally revolves around {topic}.
```

For the first turn, before any dialogue history is available, the user message is:

```
(start the call with a warm, natural greeting, then soon bring up {topic})
```

A.8 Reproducibility

FD-LHB involves no training: it runs existing models on fixed inputs and scores their outputs. Upon publication, we will release the code to run each model with IAP and DRP, the FD-LHB-LLM partner, and the metric computation, including the Gemini judge prompts. We will also release a manifest listing the CANDOR conversations and crop boundaries used in each experiment. CANDOR audio is available only through its research-access process, so we release the manifest rather than the audio. To reproduce the results, one rebuilds the inputs from the manifest, runs each model with the settings in Table 9, and scores the outputs as described in Appendix A.2.

B Diagnostic Experiments: Details and Additional Analyses

This section gives the setup of the diagnostic experiments in the main text and additional analyses of the two mechanisms: initialization-anchor eviction (Appendix B.1) and self-reinforcing repetition (Appendix B.2). The anchor analyses measure gibberish with predictive entropy (Appendix A.2) and a threshold of $H = 2$ nats, rather than with the Gemini label.

B.1 Initialization-Anchor Eviction

Evidence for the training premise. The main text assumes that every training sequence starts from an empty cache. We confirmed this in the official Moshi fine-tuning code (Kyutai, 2025), where each training sample is a segment of at most `duration_sec` seconds processed from an empty cache. Its README also warns that lowering `duration_sec` may cause the model to become silent more quickly at inference, which is consistent with a mismatch between training and inference lengths. This code covers fine-tuning rather than pretraining, and PersonaPlex is trained separately, so we test the role of anchor eviction directly with the interventions below.

Only the first entry matters. In the main text, we test which cache entries must be kept to prevent gibberish (Fig. 5). Here we describe these tests in more detail and show the entropy of every run. Both tests use 15 CANDOR conversations for each of Moshi and PersonaPlex (30 runs per condition, fixed decoding seed). The *identity* test keeps one early entry and varies only which one it is, with its cache slot and position index fixed; the *count* test keeps the first P entries, $P \in \{0, 1, 2, 4, \ldots, 128\}$. Fig. 11 shows each run's entropy averaged over the late part of the conversation (minutes 13–20 in

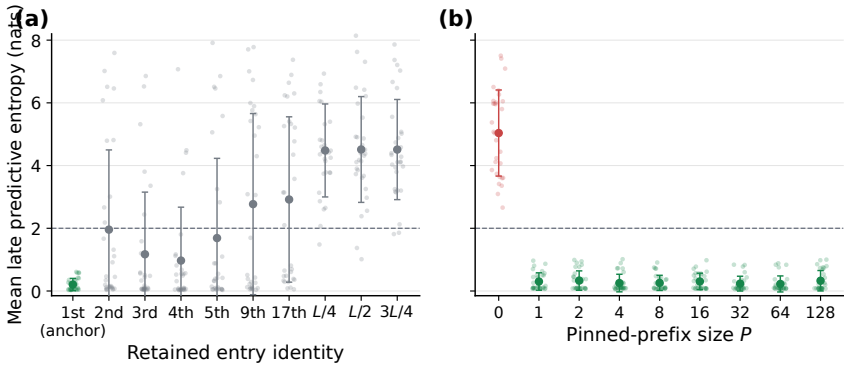


Figure 11: **Late-conversation entropy when keeping different cache entries.** (a) Keeping one early entry. (b) Keeping the first P entries. Each dot is one run; the dashed line is $H = 2$.

(a) and 10–30 in (b)), together with the mean and SD of each condition. In the identity test (a), only the first entry keeps every run below $H = 2$; every other choice leaves some runs far above it, so the result does not depend on the exact threshold. In the count test (b), entropy is high without any retained entry ($P = 0$) and low for every $P \geq 1$. Varying the decoding seed (42–46) on one PersonaPlex replay gives the same result.

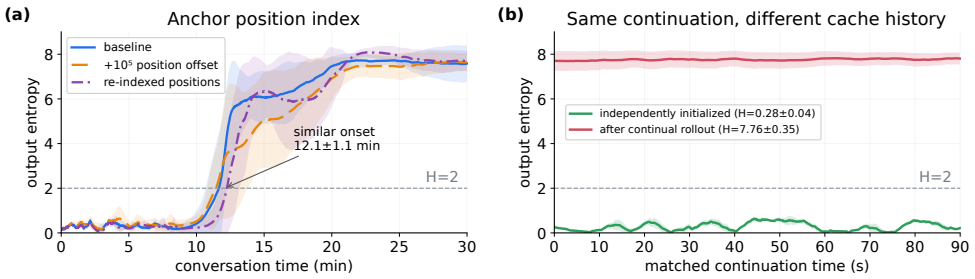


Figure 12: **Position indices do not matter; the cache history does.** (a) Output entropy over a conversation with the original, shifted, or re-indexed position indices. (b) Output entropy for the same 90-second audio, played after a fresh start or after a long conversation. Mean $\pm$ SD over five seeds, pooled across Moshi and PersonaPlex.

Large position indices are not the cause. In long conversations, position indices grow far beyond those seen in training. Because RoPE depends only on relative distance, this should not matter, and Fig. 12a confirms it. Shifting all positions by 10^5, or re-indexing the visible cache at every step so that its positions always run from 0 to $L - 1$ and never exceed the training range, leaves the onset of high entropy unchanged (12.1 $\pm$ 1.1 minutes).

The cache history is what matters. If position indices are not the cause, what differs between early and late in a conversation is the history stored in the cache. To isolate it, we play the same 90-second audio to the model twice: once right after a fresh start, with an empty cache, and once after the model has already run a long conversation. Everything else is identical, including the audio, the position indices, and the decoding settings. After a fresh start, entropy stays low

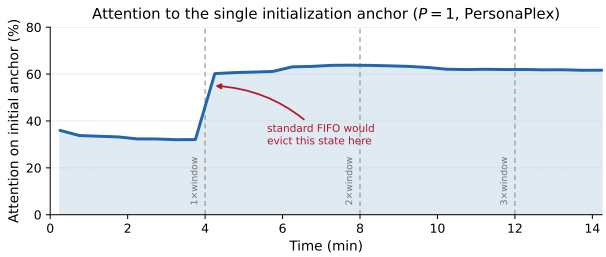


Figure 13: **The pinned anchor acts as an attention sink.** Share of attention on the pinned anchor in one PersonaPlex replay with IAP ($P = 1$), in 30-second averages. Dashed lines mark each full turnover of the sliding window.

($H = 0.28 \pm 0.04$); after the long conversation, it is very high ($H = 7.76 \pm 0.35$; Fig. 12b). The history in the cache therefore changes the model's behavior.

The pinned anchor acts as an attention sink. IAP resembles StreamingLLM (Xiao et al., 2024b), which also keeps the first few tokens of a sequence in the cache. StreamingLLM showed that language models place a large share of their attention on these first tokens, which it calls *attention sinks*, and that the model fails once they are removed. To check whether the pinned anchor plays the same role, we measure how much attention it receives over time. We run IAP ($P = 1$) on one 14-minute PersonaPlex replay and record the share of attention that goes to the pinned anchor, averaged over all layers and heads (Fig. 13). The anchor already receives about 32% of the attention during the first window. After the first window turnover at 4 minutes, its share rises above 60% and stays there. The pinned anchor therefore acts as an attention sink, and the model relies on it even more once the rest of the early context has left the window.

B.2 Self-Reinforcing Repetition

Word-insertion experiment. This section gives the setup of the word-insertion experiment in Sec. 4.2 (Fig. 6c). For each model, we take 20 CANDOR contexts that later reach repetition and branch them at least 30 seconds before repetition onset. Each branch receives $d \in \{0, 1, 2, 4, 8, 16, 32\}$ forced occurrences of *purple*, a word absent from the preceding two minutes, and then runs freely for 20 seconds with three sampling seeds. We then measure how often *purple* appears among the model's words in these 20 seconds. Forced occurrences are not counted. Without insertion, neither model says *purple*. In PersonaPlex, its share rises sharply up to $d = 4$ and keeps growing, reaching 28.4% at $d = 32$. In Moshi, it stays below 10% at every dose.

C Implementation Details of IAP and DRP

This section provides the implementation details and final settings of IAP and DRP. Evaluation results and ablations are reported in Appendix D.

C.1 Initialization-Anchor Pinning

IAP pins the first KV entry ($P = 1$) in the cache. The other $L - 1$ slots follow the normal FIFO update, so the cache size stays at L and no training is needed.

The pinned entry needs one extra step. RoPE (Su et al., 2023) attention depends on the distance between query and key positions. If the pinned entry stayed at its original position, its distance to new queries would keep growing and soon exceed the distances seen in training. IAP therefore increases the pinned entry's position by one each time the window slides, so that it always sits right before the oldest remaining entry. All other entries keep their positions, and each new entry receives the next position. To do this, we store the pinned key without RoPE applied and re-apply RoPE at its new position whenever it moves. Only this one key is updated at each step, instead of re-indexing the whole cache as in StreamingLLM (Xiao et al., 2024b).

C.2 Duplex Repetition Penalty

DRP modifies the text logits in two parts. The first penalizes recently used lexical tokens while keeping the total lexical probability unchanged, so that the penalty does not make the model less likely to speak. The second applies a small `PAD` correction after unusually long silent periods. DRP is applied before temperature scaling and top-k filtering. When predictive entropy is used for analysis, it is computed from the original logits before applying DRP.

Penalizing recent lexical tokens. Let $z_{t,i}$ be the text logit of token i at frame t, let $\mathcal{V}$ be the set of lexical text tokens (all text tokens except `PAD` and `EPAD`), and let $\mathcal{H}_t \subseteq \mathcal{V}$ be the set of tokens appearing among the model's last W lexical tokens. We first apply the repetition penalty of Keskar

et al. (2019):

$$R_\rho(z_{t,i}) = \begin{cases} z_{t,i}/\rho, & i \in \mathcal{H}_t,\ z_{t,i} > 0, \\ \rho z_{t,i}, & i \in \mathcal{H}_t,\ z_{t,i} \le 0, \\ z_{t,i}, & \text{otherwise}, \end{cases} \tag{1}$$

where $\rho > 1$ controls the penalty strength. Each token in $\mathcal{H}_t$ receives the same penalty regardless of how often it appeared.

This penalty also lowers the total probability of lexical tokens, which would make `PAD` and `EPAD` relatively more likely. To prevent this, we then add the same offset c_t to every lexical-token logit:

$$\begin{aligned} \tilde{z}_{t,i} &= R_\rho(z_{t,i}) + c_t, \qquad i \in \mathcal{V}, \\ c_t &= T_t \log \frac{\sum_{j\in\mathcal{V}} \exp\big(z_{t,j}/T_t\big)}{\sum_{j\in\mathcal{V}} \exp\big(R_\rho(z_{t,j})/T_t\big)}, \end{aligned} \tag{2}$$

where T_t is the text temperature. With this offset, the total lexical probability after temperature scaling is the same as before the penalty. Probability removed from recently used tokens therefore goes to other lexical tokens rather than to `PAD` or `EPAD`.

Prolonged-`PAD` safeguard. The repetition penalty can occasionally make it harder for the model to resume speaking after a long silence. We therefore lower the `PAD` logit when the model has stayed silent for a long time and VAD indicates that the user is quiet.

Let r_t be the current number of consecutive `PAD` frames, r_0 the length at which the correction begins, and τ_{ramp} the number of frames over which it grows to its maximum $b_{\max}$. Let $q_t = 1$ when VAD detects that the user is quiet and $q_t = 0$ otherwise. At frame t, we subtract

$$b_t = q_t b_{\max} \operatorname{clip}\left(\frac{r_t - r_0 + 1}{\tau_{\text{ramp}}}, 0, 1\right) \tag{3}$$

from the `PAD` logit, where $\operatorname{clip}(x, 0, 1) = \min(\max(x, 0), 1)$. The correction is zero until the silence reaches r_0 frames, then grows linearly over τ_{ramp} frames up to $b_{\max}$. It is off while the user is speaking and resets when the model outputs `EPAD` or a lexical token.

C.3 Final Settings

Our final configuration uses a history of $W = 512$ lexical text tokens. For the prolonged-`PAD` safeguard, the correction begins when the `PAD` run reaches $r_0 = 375$ frames (30 s), increases to its maximum over $\tau_{\text{ramp}} = 125$ frames (10 s), and is capped at $b_{\max} = 3.0$. The repetition-penalty strength is $\rho = 1.5$ for Moshi and $\rho = 1.3$ for PersonaPlex. IAP retains a single initialization anchor ($P = 1$).

Model-specific generation settings are summarized in Table 9.

Table 9: **Generation settings for each model.** T_a/T_t denote audio/text temperatures and k_a/k_t the corresponding top-k values. The last column reports the context limit and evaluation horizon. Baseline and Ours use identical settings within each track.

System and track	T_a/T_t	Sampling restriction	Seed	Context / evaluation horizon
PersonaPlex, both tracks	.8/.65	$k_a/k_t = 250/25$	42	3,000 frames / full replay; 20 min interaction
PersonaPlex, diagnostics	.8/.7	$k_a/k_t = 250/25$	42–46	varied window / 20 or 30 min
Moshi, replay (Baseline and Ours)	.8/.7	$k_a/k_t = 250/25$	42	3,000 frames / full input
Moshi, interaction (Baseline and Ours)	.8/.9	$k_a/k_t = 250/25$	42	3,000 frames / 20 min
Moshi, diagnostics	.8/.7	$k_a/k_t = 250/25$	42–46	varied window / 20 or 30 min
Raon-SpeechChat, replay	–/.6	top-p = .8, top-k = 20, BC = 2	42	8,192-position cache cap
Raon-SpeechChat, interaction	–/.9	top-p = .95, top-k = 66, BC = 0	42	8,192-position cache cap / 20 min
BayLing-Duplex, replay	–/.6	top-p = .8; ratio 10:5:10	42	32,768-position native cap
BayLing-Duplex, interaction	–/.6	top-p = .8; ratio 10:5:10	42–81	native cap / 20 min
Lychee-FD, both tracks	native API	native API	not exposed	400-ms incremental / 10.92-min configured support

Table 10: **Complete FD-LHB results.** Every condition contains 40 runs. T_{rep}, T_{psil}, and T_{gib} denote the first of two consecutive repetitive 32-word windows, the first of two consecutive silent one-minute bins, and the first gibberish-labeled speaking bin, respectively. Onsets are mean $\pm$ sample SD over affected runs. Configured or native context limits are right-censored; NR means no event was observed within the evaluation horizon.

Track	Model	Method	T_{rep}	T_{psil}	T_{gib}
Replay	Raon-SpeechChat	Baseline	NR 0/40 reached	3.2 ± 1.1 14/40 reached	3.0 1/40 reached
	BayLing-Duplex	Baseline	2.2 ± 1.6 22/40 reached	6.8 ± 1.4 40/40 reached	4.2 ± 1.2 36/40 reached
	Lychee-FD	Baseline	2.4 ± 2.2 15/40 reached	NR 0/40 reached	5.2 ± 3.4 12/40 reached
	PersonaPlex	Baseline	10.3 ± 7.1 36/40 reached	8.2 ± 1.3 34/40 reached	12.9 ± 3.1 40/40 reached
	PersonaPlex + Ours	IAP+DRP	25.6 1/40 reached	31.6 ± 15.1 19/40 reached	NR 0/40 reached
	Moshi	Baseline	19.6 ± 7.5 26/40 reached	9.0 ± 0.8 40/40 reached	13.3 ± 0.9 40/40 reached
	Moshi + Ours	IAP+DRP	NR 0/40 reached	30.3 ± 16.0 3/40 reached	NR 0/40 reached
Interaction	Raon-SpeechChat	Baseline	NR 0/40 reached	NR 0/40 reached	NR 0/40 reached
	BayLing-Duplex	Baseline	2.7 1/40 reached	8.2 ± 2.7 40/40 reached	4.3 ± 2.4 40/40 reached
	Lychee-FD	Baseline	NR 0/40 reached	NR 0/40 reached	3.2 ± 3.3 9/40 reached
	PersonaPlex	Baseline	7.1 ± 4.6 35/40 reached	7.6 ± 1.6 35/40 reached	11.8 ± 2.0 39/40 reached
	PersonaPlex + Ours	IAP+DRP	16.9 ± 3.1 2/40 reached	14.8 ± 2.5 4/40 reached	NR 0/40 reached
	Moshi	Baseline	10.0 ± 6.1 5/40 reached	9.5 ± 0.5 28/40 reached	12.2 ± 0.8 40/40 reached
	Moshi + Ours	IAP+DRP	NR 0/40 reached	NR 0/40 reached	17.5 ± 0.7 2/40 reached

D Extended Results

This section extends the results in Sec. 6. Appendix D.1 gives the complete benchmark results and significance tests. Appendix D.2 reports response quality and failure trajectories over long replays, and Appendix D.3 examines how long silences last. Appendix D.4 measures the effect of IAP and DRP separately, Appendix D.6 replaces DRP's repetition penalty with other repetition controls, and Appendix D.7 checks with human listeners whether DRP makes ordinary speech worse. Finally, Appendix D.8 reports the computational cost.

D.1 Complete Main Results

Table 10 gives the full numbers behind Fig. 7. Baseline and IAP+DRP receive the same 40 replays, so we compare their results on each replay. We check whether IAP+DRP avoids each failure in more replays than the baseline, using an exact McNemar test (McNemar, 1947). This gives six tests (two models and three failures). All six are significant, even with the stricter Bonferroni threshold ($p < 0.05/6$).

D.2 Long-Horizon Trajectories and Response Quality

Fig. 8 reports LLM-judge scores on FD-LHB-LLM; here we report the same scores on the FD-LHB-CANDOR replays (Fig. 14). As in the main text, Gemini rates each one-minute speaking bin from 1 to 5 for Coherence and Context, and silent bins receive a score of 1 so that silence lowers the average

instead of dropping out of it. The replay results follow the same pattern as the interaction track: both sliding-window baselines fall to the lowest score within about 20 minutes, whereas IAP+DRP keeps both scores steady for the full 60 minutes. BayLing-Duplex stays near the lowest score throughout because it often responds in a language other than English.

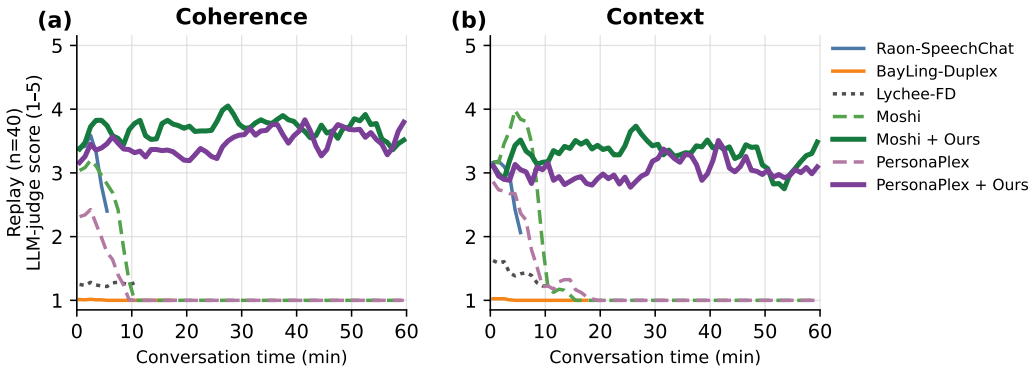


Figure 14: **LLM-judge scores over time on FD-LHB-CANDOR.** Mean (a) Coherence and (b) Context scores over 40 replays per system. Silence bins receive a score of 1. Curves for growing-context models end when the model reaches its context limit.

Fig. 15 repeats the view of Fig. 2 after applying IAP+DRP, extended to the full 60-minute replay (W1–W15). Without IAP+DRP, both sliding-window models reach gibberish about three windows in (Fig. 2a,b). With IAP+DRP, repetition and gibberish stay near zero through all 15 windows, and words per minute stay steady instead of rising sharply, as they do when the baselines turn to gibberish.

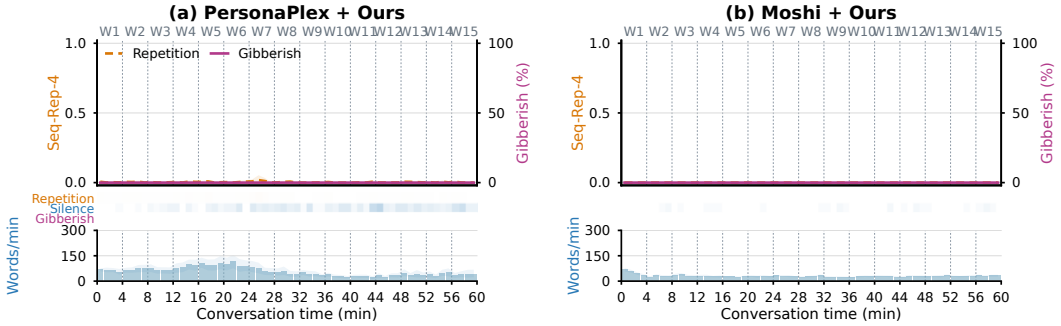


Figure 15: **Failure trajectories with IAP+DRP over 40 CANDOR replays.** Same format as Fig. 2 for (a) PersonaPlex and (b) Moshi with IAP+DRP, extended to 60 minutes. Top: repetition (Seq-Rep-4) and the percentage of speaking bins labeled gibberish; middle: when each failure appears; bottom: words per minute. W1–W15 mark consecutive four-minute sliding windows.

D.3 Silence Duration and Resumption of Speech

The silence metric in the main text counts a run once it contains a two-minute silence, even if the model later speaks again. Here we check how long these silences last and what happens afterward. Over all 40 CANDOR replays per condition, Table 11 reports how many runs contain a silence of at least two minutes and of at least five minutes. It also reports how many runs resume speaking after the first two-minute silence, and how long this takes. This checks only whether the model speaks again, not whether its speech is coherent or relevant.

Two-minute silences occur in every Moshi baseline run and in 34 of 40 PersonaPlex baseline runs; IAP+DRP reduces them to 3 and 19 of 40, respectively. In almost every condition, including the baselines, the model eventually speaks again. The difference lies in how quickly the model resumes

Table 11: **Silence duration and resumption of speech on all 40 CANDOR replays.** A one-minute bin is silent if it contains fewer than three words; the final partial minute is included. Any 2 min and Any 5 min count runs with a silence of at least two or five consecutive minutes. Resumed counts runs that produce at least one word after the first two-minute silence. Time to resume is measured from the end of that silence to the first word; cells report the median, IQR, and maximum in seconds.

Model	Method	Any 2 min	Any 5 min	Resumed	Median time to resume (s)
Moshi	Baseline	40/40	7/40	40/40	95.3 IQR 57.2–134.4; max 202.6
Moshi	IAP	13/40	0/40	13/13	7.5 IQR 2.6–24.8; max 70.2
Moshi	DRP	37/40	7/40	37/37	95.2 IQR 68.4–134.0; max 203.9
Moshi	IAP+DRP	3/40	0/40	3/3	10.2 IQR 6.7–16.1; max 22.0
PersonaPlex	Baseline	34/40	7/40	34/34	53.6 IQR 37.0–118.8; max 172.0
PersonaPlex	IAP+DRP	19/40	1/40	18/19	26.6 IQR 12.4–40.2; max 47.3

and what it produces afterward. Without IAP, resumption is slow (median 95.3 s for Moshi and 53.6 s for PersonaPlex), and five-minute silences occur in 7 of 40 runs. Moreover, although every baseline run resumes speaking, all of them also reach gibberish, typically soon after resuming (silence starts at about 9 minutes and gibberish at about 13 minutes; Table 10). This suggests that the silence is a transition into gibberish rather than a pause. With IAP, resumption is much faster (median 7.5 s with IAP and 10.2 s with IAP+DRP in Moshi, and 26.6 s with IAP+DRP in PersonaPlex), five-minute silences almost disappear, and no run produces gibberish. In PersonaPlex, IAP+DRP still leaves a two-minute silence in 19 of 40 runs, but 18 of them resume speaking (median 26.6 s), so most of these silences are temporary pauses.

D.4 Component Ablations

Table 12 compares Baseline, IAP, DRP, and IAP+DRP on Moshi over 40 matched CANDOR replays. IAP alone removes gibberish in all 40 runs and also reduces silence (from 40 to 13 runs) and repetition (from 26 to 4 runs). DRP alone removes repetition in all runs but does not prevent gibberish or silence (40 and 37 runs). Combining the two gives the best result on all three failures, with silence in only 3 runs. In Fig. 16, only IAP+DRP also keeps Coherence high over the full 60 minutes. The two methods therefore play complementary roles, although these results alone do not show that each failure has a single cause.

Table 12: **Moshi component ablation on 40 matched CANDOR replays.** Definitions and reporting follow Table 10.

Method	T_{rep}	T_{psil}	T_{gib}
Baseline	19.6 ± 7.5 26/40 reached	9.0 ± 0.8 40/40 reached	13.3 ± 0.9 40/40 reached
IAP	24.8 ± 21.3 4/40 reached	33.0 ± 14.3 13/40 reached	NR 0/40 reached
DRP	NR 0/40 reached	9.6 ± 0.6 37/40 reached	13.7 ± 2.0 40/40 reached
IAP+DRP	NR 0/40 reached	30.3 ± 16.0 3/40 reached	NR 0/40 reached

D.5 DRP Ablation

DRP has two parts (Appendix C.2). The first penalizes recent words (eq. (1)) and then adds an offset that keeps the total probability of words unchanged (eq. (2)). The second is a `PAD` safeguard against

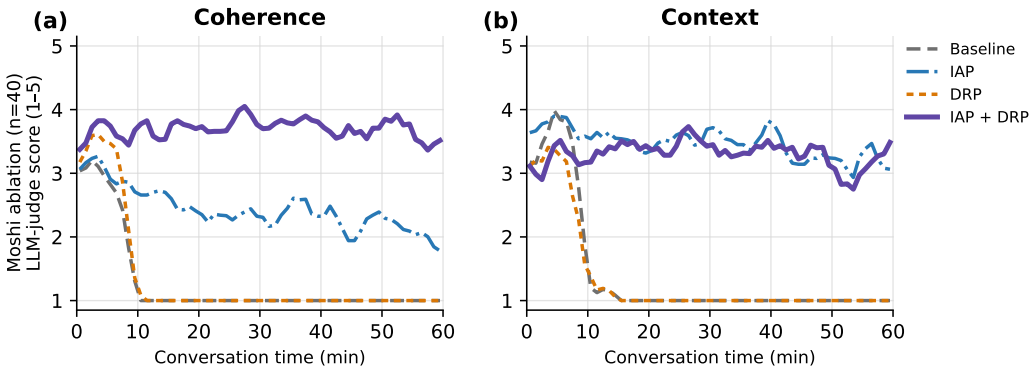


Figure 16: **Response quality in the Moshi component ablation.** Mean Coherence and Context scores over 40 matched CANDOR replays. Silence bins receive a score of 1.

long silence (eq. (3)). To see what each part does, we compare them on 10 matched 20-minute PersonaPlex replays. All runs use IAP ($P = 1$).

Table 13: **DRP ablation on 10 matched PersonaPlex CANDOR replays.** All runs use IAP and last 20 minutes. Rep. and Sil. count runs that reach repetition or prolonged silence. Coh. and Ctx. are mean Gemini Coherence and Context scores; silent bins receive a score of 1.

Method	Rep. ↓	Sil. ↓	Coh. ↑	Ctx. ↑
IAP only	7/10	1/10	1.69	2.67
IAP + standard RP	2/10	5/10	2.20	1.56
IAP + DRP (w/o `PAD` safeguard)	0/10	5/10	2.57	2.15
IAP + DRP	0/10	2/10	2.82	2.57

Standard RP uses only the penalty. It reduces repetition from 7 to 2 runs, but the model becomes silent in 5 runs and its Context score drops. Without the `PAD` safeguard, DRP removes the remaining repetition and raises both scores, but silence still occurs in 5 runs. Full DRP also adds the `PAD` safeguard, which reduces silence to 2 runs while keeping repetition at zero and raises both scores further. The two parts of DRP therefore fix different problems.

D.6 Alternative Repetition Controls

We compare DRP with three other ways to suppress recently used words, on the same 10 PersonaPlex replays. Only the repetition control changes; the model, input, seed, and IAP ($P = 1$) are fixed. Standard RP applies the CTRL penalty (Keskar et al., 2019) without DRP's other parts. LZ (Ginart et al., 2026) uses the published settings ($\alpha = 0.15$, 512-token lookback, 32-token buffer). The Bayesian penalty (Fan et al., 2026) uses a unigram prior with add-one smoothing, estimated from the first four minutes of PersonaPlex baseline output on 20 other CANDOR replays; the 10 evaluation replays are not used.

Table 14: **Alternative repetition controls on 10 matched PersonaPlex CANDOR replays.** All runs use IAP and last 20 minutes. Columns follow Table 13.

Control	Rep. ↓	Sil. ↓	Coh. ↑	Ctx. ↑
IAP only	7/10	1/10	1.69	2.67
Standard RP	2/10	5/10	2.20	1.56
LZ	8/10	2/10	1.86	2.69
Bayesian	0/10	1/10	2.33	1.94
DRP	0/10	2/10	2.82	2.57

LZ does not prevent repetition on PersonaPlex, and Standard RP leaves the model silent in half of the runs. Bayesian prevents both repetition and silence, but its Context score is lower than DRP's. DRP prevents repetition, keeps silence low, and has the highest Coherence and a relatively high Context score. DRP also needs only a fixed window of recent tokens, whereas LZ and Bayesian require extra state or a prior estimated from data. We used the published settings for LZ and Bayesian and did not tune them for PersonaPlex, so these results do not show that DRP is the best possible control. Other penalties that suppress recently used words may also work well once tuned.

D.7 Human Listening Comparison

Because DRP penalizes every recently used word, including words that people naturally repeat, such as function words, backchannels, and topic words, it could make speech less natural. We ran a blinded A/B listening study to check whether DRP makes ordinary speech worse as a conversation goes on. We compare IAP+DRP with IAP alone on Moshi, isolating the effect of DRP. We keep IAP in both conditions because without it, both would produce gibberish after about 13 minutes, leaving nothing meaningful to compare in later phases. The study uses 37 matched pairs sampled early, middle, and late in the conversation (near 2, 10, and 18 minutes). In each trial, listeners heard 8 seconds of shared conversation context followed by two 12-second continuations, and chose A, B, or a tie on five dimensions: naturalness, intelligibility, prosody, conversational style, and appropriateness to the context. Conditions, transcripts, and generation times were hidden. Fifteen participants recruited through Prolific, who gave informed consent and were compensated, completed the study. We report a preference score for IAP+DRP: choosing it counts as 1, a tie as 0.5, and choosing IAP alone as 0, so 0.5 means no preference.

Overall, scores are close to 0.5 on every dimension (0.47–0.53; Table 15), but this hides a trend over time: listeners prefer IAP alone early (0.31–0.43) but prefer IAP+DRP in the middle (0.56–0.60) and late (0.53–0.60) phases. This suggests that DRP suppresses some natural word reuse early in a conversation but becomes helpful once repetition starts to build up.

Table 15: **Human listening preferences for IAP+DRP over IAP alone on Moshi.** Each cell shows how often listeners chose IAP+DRP, with ties counted as half. Higher values mean IAP+DRP was chosen more often, and 0.5 means no preference. Brackets show 95% bootstrap intervals; an interval that includes 0.5 indicates no clear preference. Numbers in parentheses are the numbers of matched pairs; the study contains 285 judgments per dimension from 15 listeners.

Phase	Naturalness	Intelligibility	Prosody	Conv. style	Context
Early (14)	0.43 [0.30, 0.55]	0.38 [0.27, 0.48]	0.41 [0.28, 0.54]	0.43 [0.30, 0.56]	0.31 [0.22, 0.41]
Middle (11)	0.60 [0.51, 0.69]	0.59 [0.50, 0.68]	0.60 [0.50, 0.71]	0.60 [0.50, 0.71]	0.56 [0.46, 0.68]
Late (12)	0.53 [0.43, 0.65]	0.55 [0.41, 0.69]	0.60 [0.41, 0.78]	0.58 [0.43, 0.73]	0.58 [0.44, 0.73]
All (37)	0.52 [0.44, 0.59]	0.50 [0.43, 0.57]	0.53 [0.45, 0.61]	0.53 [0.45, 0.61]	0.47 [0.40, 0.55]

D.8 Computational Cost

Setup. We measure the inference cost of each method on PersonaPlex using a single NVIDIA RTX A6000 GPU with bfloat16 weights. We compare the standard FIFO cache, a StreamingLLM-style cache that keeps the first four entries and re-indexes the whole cache at every step, DRP, IAP, and IAP+DRP. All methods use the same 440-second replay, persona, sampling settings, seed, and 3,000-entry cache. Timing starts only after the cache is full, since the methods differ only once old entries begin to be evicted. We then measure 1,000 consecutive frames (80 seconds of audio) and report the average time per frame. Each method is run five times, each in a new process. Model loading, file I/O, and resampling are excluded.

Metrics. Table 16 reports the time per frame. *LM GPU* is the time the language model takes on the GPU to produce one frame. *Pipeline* adds input-audio encoding and output-audio decoding. *Total* is the time of the full pipeline for one frame. *RTF* is the time, in seconds, needed to process one second of audio; an RTF below 1 means the model runs in real time.

Table 16: **Per-frame inference cost on PersonaPlex.** Mean ± SD over five runs with a full 3,000-entry cache. Overhead is the increase in total time relative to FIFO. An RTF below 1 means faster than real time.

Method	LM GPU (ms/frame)	Pipeline (ms/frame)	Total (ms/frame)	RTF	Overhead
FIFO	39.939±0.111	47.653±0.163	47.653±0.163	0.5957±0.0020	–
StreamingLLM-style	74.818±0.153	82.583±0.229	82.583±0.229	1.0323±0.0029	+73.3%
DRP	40.485±0.100	48.154±0.140	48.944±0.138	0.6118±0.0017	+2.7%
IAP	51.153±0.136	59.083±0.598	59.083±0.598	0.7385±0.0075	+24.0%
IAP+DRP	51.694±0.136	59.414±0.191	60.198±0.185	0.7525±0.0023	+26.3%

Results. DRP adds almost no cost (+2.7%), since it only edits the text logits. IAP adds 24.0%, but IAP+DRP still runs well within real time (RTF 0.75). StreamingLLM-style is the slowest (+73.3%): because it re-indexes every cached entry at each step, its RTF exceeds 1, so it cannot keep up with real time.

E Exploratory Observations and Limitations

This section reports exploratory observations outside the main benchmark and discusses the limitations of this work. Appendix E.1 checks whether repetition also appears in additional speech models; because it uses different protocols, its results are not pooled with FD-LHB. Appendix E.2 then discusses the limitations.

E.1 Exploratory Observations on Additional Speech Models

We also ran several additional speech models on long inputs to see whether repetition appears beyond the models in our main benchmark (Table 17). Each model uses its own prompt, input, context unit, and stopping rule, and we observe one sequence per model. These results are therefore qualitative: they cannot be compared with FD-LHB or used to rank the models. Seven of the nine models eventually repeat, while GLM-4-Voice and LLaMA-Omni2 stay stable in the tested sequence.

Table 17: **Exploratory observations on additional speech models.** One sequence per model; lengths and protocols differ across models.

Model	Tested length	Observed behavior
mini-omni (Xie & Wu, 2024a)	2,048 tokens	Repeats replies before reaching the context limit.
mini-omni2 (Xie & Wu, 2024b)	2,048 tokens	Coherent until near the limit, then repeats.
Covo-Audio (Wang et al., 2026b)	a few thousand tokens	Replies converge to a small repeated set.
Kimi-Audio (KimiTeam et al., 2025)	about 1.3K of 8K tokens	Switches to describing the audio and repeats that description.
Qwen2.5-Omni (Xu et al., 2025)	about 11 turns / 2.5K tokens	Outputs template text, then repeats one response.
Step-Audio 2-mini (Wu et al., 2025)	tested dialogue	Coherent early, then repeats replies.
GLM-4-Voice (Zeng et al., 2024)	8,192 tokens	Stable and on-topic.
LLaMA-Omni2 (Fang et al., 2025)	60 turns / 3,232 tokens	Stable.
Baichuan-Audio (Li et al., 2025)	separate long-input test	Repeats one answer; the run later ends when GPU memory runs out.

E.2 Limitations

Scope of the diagnosis. Our diagnosis covers only the two sliding-window models, Moshi and PersonaPlex. Growing-context models such as BayLing-Duplex also collapse before reaching their context limit, but we do not identify the cause, and IAP does not apply to them because they never evict cache entries. Even for the sliding-window models, our evidence for Initialization-Anchor Eviction comes mainly from controlled experiments. The fixed-slot test (Sec. 4.1.2) shows that the first KV entry is important, while the matched-cache test (Appendix B.1) shows that the previous

cache state affects the failure. The fixed-slot test measures predictive entropy, so it does not directly measure dialogue quality. It also does not show how often the same failure occurs across different conversations or inputs. Broader tests across models, checkpoints, inputs, random seeds, architectures, and languages are needed.

Evaluation horizon and cross-model comparison. FD-LHB observes each model for a finite time, so NR means only that a failure did not occur before the run ended, not that the model would remain stable indefinitely. The evaluated models also differ in architecture, context management, and generation settings, including model-specific text temperatures in the adaptive track. Therefore, comparisons between methods within the same model are more directly interpretable than comparisons across different models.

Metrics and trade-offs. Our evaluation metrics are not perfect measures of overall conversation quality. First, our criteria (Sec. 3.2) detect repetition, silence, and gibberish separately. A conversation that avoids these failures may still be incoherent or irrelevant. Second, reducing one type of failure can introduce a trade-off elsewhere. For example, DRP may penalize natural word reuse, and listeners preferred IAP alone over IAP+DRP early in the conversation (Appendix D.7). Moreover, the proposed methods do not eliminate all failures: in PersonaPlex, IAP+DRP still produces a two-minute silence in 19 of 40 replays. Finally, the Gemini-based gibberish detector is also imperfect. Our human validation (Appendix A.3) found no false positives but did find some missed cases, suggesting that the reported gibberish rates may underestimate the true frequency. These limitations motivate broader and longer-term human evaluation of both conversation quality and possible trade-offs.