Speaking While Listening: A Survey and Empirical Audit of Full-Duplex Spoken Dialogue Systems

Anonymous ACL submission

Abstract

More than a dozen spoken dialogue systems have recently claimed to be "full-duplex," yet the term has been used to describe substantially different capabilities. Existing surveys collapse them onto a single axis (cascaded/end-to-end, or engineered/learned) and miss the distinctions that matter most for builders. We argue that much of this ambiguity is taxonomical: current terminology does not specify where duplex decisions are made, which interaction types are supported, or how a system behaves moment by moment. This paper introduces three complementary frameworks: **(i)** an **L0–L3 Architectural Hierarchy** that locates where duplex decisions are made; **(ii)** a $T \times I \times R$ **Interaction Ontology** that specifies the temporal relation, user intent, and required system response for each interaction; and **(iii)** a **Decision State Machine** (IDLE/LISTEN/SPEAK/WAIT/DUAL) that describes how systems move between states. Across published systems and benchmarks, our audit reveals a realization gap: although many architectures can in principle operate in full-duplex states, their observed behavior remains constrained by the interaction patterns represented in training and evaluation. We identify the open-source-versus-industrial data gap, as well as the still-unrealized goal of L3 representation-level modeling, as key frontiers for future research on full-duplex dialogue. The related material is available at this https url.

1 Introduction

Full-duplex spoken dialogue—characterized by concurrent speaking and listening, overlaps, backchannels, and barge-ins—is not a strictly new capability. Before the large language model (LLM) era, industrial systems such as Google Duplex (Leviathan and Matias, 2018), Ant Group (Jin et al., 2021), and Alibaba DAMO (Lin et al., 2022) had already deployed modular full-duplex pipelines at production scale by 2022. What changed with OpenAI's GPT-4o demonstration in May 2024[1] was the architectural paradigm: showcasing an end-to-end speech-to-speech model that could be smoothly interrupted mid-utterance. This demonstration elevated full-duplex interaction from a specialized industrial feature to a baseline product expectation.

In the subsequent two years, more than a dozen new systems have emerged claiming this capability, including Moshi (Défossez et al., 2024), MinMo (Chen et al., 2025b), OmniFlatten (Zhang et al., 2025b), SyncLLM (Veluri et al., 2024), LSLM (Ma et al., 2025), and FireRedChat (Chen et al., 2025a). However, these systems cluster under the "full-duplex" banner while functioning in radically different ways. For instance, a modular controller that yields the floor upon detecting an interruption, a parallel-stream token decoder, and a hidden-state predictor feeding a downstream module are all branded identically. Grouping these distinct mechanisms under a single label obscures critical differences in their actual interaction coverage, latency profiles, and failure modes.

We argue that the current ambiguity in the field stems less from architectural differences than from the lack of a unified taxonomy. Existing surveys (Ji et al., 2024; Chen and Yu, 2025) typically collapse this design space onto a single axis (e.g., cascaded versus end-to-end, or engineered versus learned). While useful, these single-axis classifications are insufficient: they fail to articulate *where* a duplex decision is made in the model stack, *what* specific interaction types a system can handle, or *how* the system's behavior evolves moment by moment.

This survey makes two coupled contributions, conceptual and empirical.

This survey contributes three complementary frameworks for describing full-duplex spoken dialogue systems. First, an **L0–L3 Architectural**

[1] https://openai.com/index/hello-gpt-4o/

Hierarchy (§3) locates where the duplex decision is made in the model stack, ranging from external modules to hidden-state predictors, token-level synchronization, and shared latent representations. Second, a **T** × **I** × **R Interaction Ontology** (§4) describes full-duplex moments in terms of temporal relation, user intent, and required system response, singling out six acid-test cells that distinguish a genuine full-duplex system from a half-duplex one. Third, a **Decision State Machine** (§5) extends the lineage of Wang et al. (2024a) and Liao et al. (2025) to describe what the system is doing at each instant using five states (IDLE, LISTEN, SPEAK, WAIT, and DUAL), coupling every transition to a specific ontology cell.

We do not leave these frameworks at the level of abstract terminology; instead, we use them to audit the published evidence along three dimensions. The **architecture audit** (§3, Table 1) maps representative systems onto the L0–L3 hierarchy and identifies their reachable FSM states. The **training-data audit** (§6) traces system behavior to the interaction cells covered by public corpora, presenting a data-bottleneck thesis supported by three pieces of evidence. The **evaluation audit** (§7) introduces the *realization gap* between architectural capacity and demonstrated behavior as a diagnostic frame, using it to clarify what current benchmarks actually test and to disambiguate the simultaneously active FDB and FD-Bench series. Finally, we close with three concrete L3 architectural hypotheses anchored in recent continuous-latent and JEPA work (§8).

Survey scope. We structure our empirical audits around three categories of systems: (i) primary targets, comprising post-GPT-4o systems that explicitly claim full-duplex behavior; (ii) historical baselines, represented by pre-2024 industrial L0 pipelines; and (iii) taxonomy references, such as adjacent streaming speech-LMs (e.g., GLM-4-Voice, Qwen-Omni series), which are included to contextualize the design space but are explicitly excluded from per-cell audits. Conversely, models restricted to TTS-only, ASR-only, voice-cloning, or non-dialogue audio generation are strictly out of scope.

2 A Brief History of Full-Duplex Spoken Dialogue Systems

The history of FD-SDS is best read as three eras separated by two pivots. The pre-LLM era (Sacks et al., 1974; Skantze, 2021) saw industrial L0 pipelines — Google Duplex (Leviathan and Matias, 2018), Ant Group (Jin et al., 2021), and Alibaba DAMO (Lin et al., 2022) — deploy VAD+EoT+DM stacks at production scale, with proprietary corpora that founded the open-source-versus-industrial data gap (§6). The technical pivot came when speech became a token (Zhang et al., 2023; Nguyen et al., 2023; Borsos et al., 2023; Wang et al., 2023; Zhang et al., 2024a; Défossez et al., 2022); the commercial pivot followed when GPT-4o (May 2024) made full-duplex a product expectation, with the immediate replication wave still half-duplex (Xie and Wu, 2024a; Fang et al., 2025; Zeng et al., 2024). The third era opened with LSLM (Ma et al., 2025) and Moshi (Défossez et al., 2024) in late 2024; the L2 design space then fragmented across OmniFlatten (Zhang et al., 2025b), SyncLLM (Veluri et al., 2024), and Mini-Omni2 (Xie and Wu, 2024b), while MinMo (Chen et al., 2025b) established the L1 hidden-state pattern for industrial deployments. Late 2025 saw a modular comeback (FireRedChat (Chen et al., 2025a), FlexDuo (Liao et al., 2025), X-Talk (Liu et al., 2025)); 2026 brought further L1 FD systems (Fun-Audio-Chat (Team et al., 2025), Covo-Audio (Wang et al., 2026b), Step-Audio R1.1 (Wu et al., 2025)), plug-and-play L0 modules (SoulX-Duplug (Yan et al., 2026)), and L2 systems exploring non-standard backbones (DuplexMamba (Lu et al., 2025)) and codec-free designs (SALMONN-omni (Yu et al., 2026)). Adjacent streaming non-FD systems (GLM-4-Voice (Zeng et al., 2024), Qwen2.5-/3.5-Omni (Xu et al., 2025; Team, 2026)) share architectural shape but do not claim full-duplex behavior (§1, scope). Figure 1 situates these on one timeline; Appendix B expands the pre-FD context and per-system narrative.

Stepping back, four architectural patterns recur across this proliferation and structure the rest of the paper. We refer to them as L0–L3 throughout and develop the cross-system audit in §3:

- **L0 Module-level**: external scheduler outside the LLM (FireRedChat, FlexDuo, X-Talk, SoulX-Duplug).
- **L1 Hidden-state-level**: external module reads LLM hidden state (FD-claiming: MinMo, Freeze-Omni (Wang et al., 2024b), Fun-Audio-Chat, Covo-Audio; the Thinker–Talker of Qwen2.5/3.5-Omni shares the same shape without claiming FD).
- **L2 Token-level**: duplex encoded in the to-

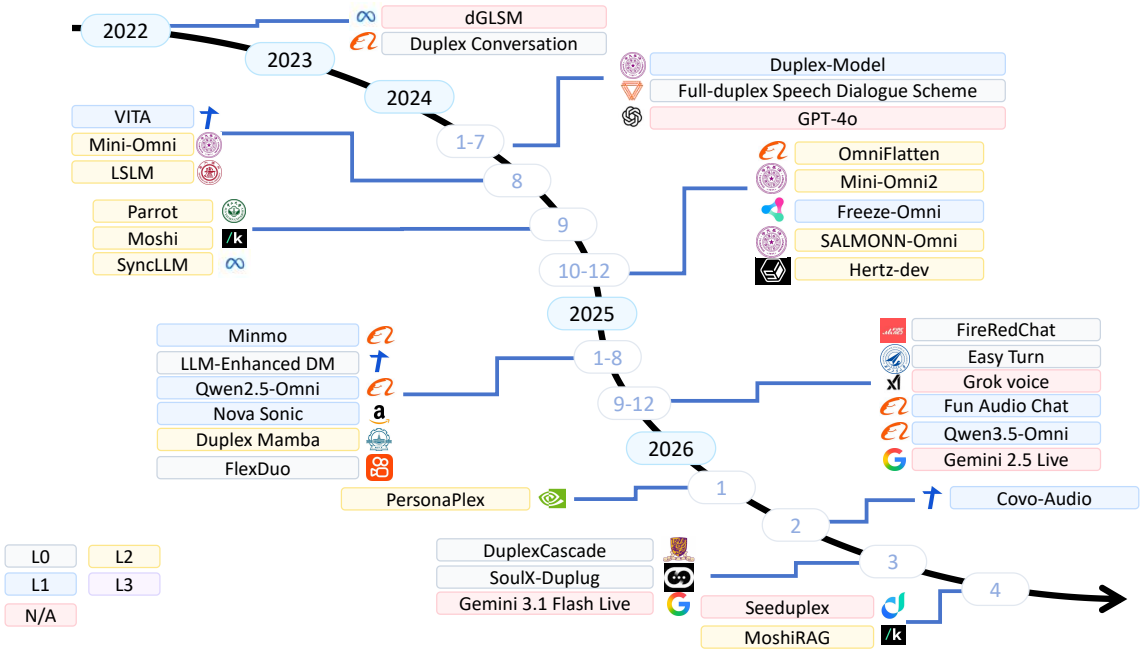

Figure 1: Timeline of published full-duplex spoken dialogue systems, 2021–2026, grouped by the L0–L3 architectural layer defined below. N/A indicates that it is not applicable to full-duplex conversation classification or closed-source

ken sequence (Moshi, LSLM, OmniFlatten, SyncLLM, Mini-Omni2, DuplexMamba on Mamba/SSM, SALMONN-omni codec-free; GLM-4-Voice shares the L2 architectural shape without claiming FD).

- **L3 Representation-level**: shared latent across streams; no published system yet.

Where prior surveys fall short. WavChat (Ji et al., 2024) organizes systems along a cascaded/end-to-end trichotomy treating full-duplex as a binary attribute; Chen and Yu (Chen and Yu, 2025) propose an Engineered-vs-Learned axis. Both collapse the design space onto a single axis. Three blind spots motivate this survey: i) neither distinguishes the stack layer at which the duplex decision is made (the L0–L3 convergence pattern of §3); ii) neither articulates *which* interactions a full-duplex system should handle (§4); iii) neither describes *how* the system decides what to do at any instant (§5).

3 Architectural Hierarchy and Cross-System Audit

The L0–L3 hierarchy was introduced informally at the end of §2 so that the historical sweep could refer to it without forward-referencing; this section makes it precise, visualizes it in Figure 2, and uses it to audit representative systems in a single table. We treat the hierarchy as the first of the three frameworks because it directly organizes the systems already encountered: the T × I × R interaction ontology (§4) and the decision state machine (§5) introduced in the next two sections then refine this view rather than precede it.

3.1 Two Observations on the Hierarchy

Looking across the L0–L3 layers, two non-trivial patterns are visible. i) **L1 is a broad architectural attractor.** The L1 shape — LLM hidden state read by an external module that produces the speech stream — recurs across designs targeting different capabilities: Chen et al. (2025b), Wang et al. (2024b), Team et al. (2025), and Wang et al. (2026b) use this pattern for full-duplex dialogue; Xu et al. (2025)'s Thinker–Talker uses the same shape for streaming text+speech generation without claiming FD. That independent teams converge on the same architectural pattern across different product goals suggests L1 is a structural attractor for LLM-based speech systems, not merely an artifact of one team's design choices. ii) **L0 remains contested rather than legacy.** Liao et al. (2025), Chen et al. (2025a), and the X-Talk position paper (Liu et al., 2025) are all 2025 arguments that the modular blueprint is competitive on latency, interpretability, and engineering cost, not merely

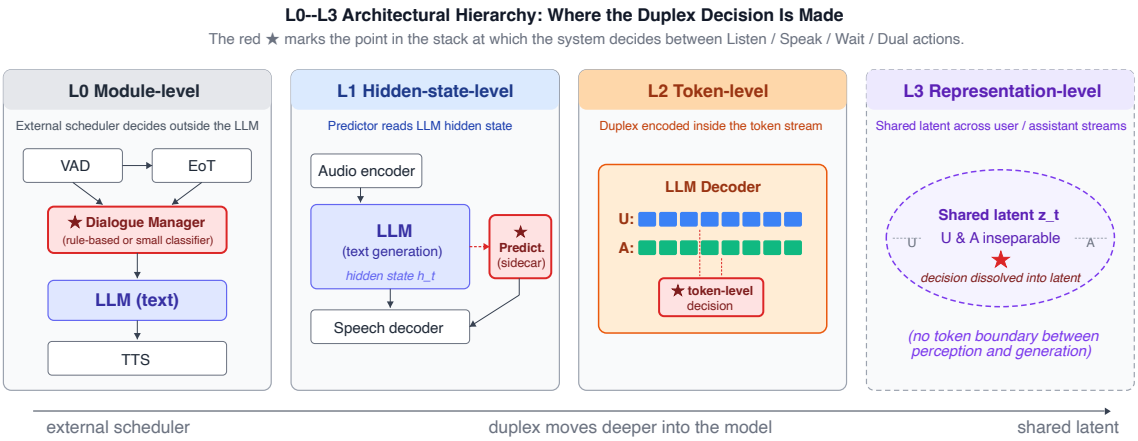

Figure 2: The L0–L3 architectural hierarchy. The red ★ marks the point in the stack at which the system decides between LISTEN, SPEAK, WAIT, and DUAL actions. The decision migrates from an external scheduler (L0) to a sidecar predictor reading the LLM hidden state (L1) to the token stream itself (L2) to a hypothetical shared latent (L3, not yet realized).

a transitional stage before L2. The L0–L3 axis is therefore a description of design choices, not a ladder of progress.

3.2 Cross-System Audit

Table 1 locates representative systems on the L-layer axis and summarizes which state-machine regions each layer can reach by construction. Per-system narrative is deferred to Appendix C.7.

Three patterns are visible in the table. i) The L2 row is the most populous, which is consistent with the academic narrative of the past two years, but L2 itself is heterogeneous: parallel-stream Moshi, flattened OmniFlatten, and chunk-alternated SyncLLM all reach DUAL but with very different time granularities (continuous for parallel, frame-level for flattened, chunk-bounded for SyncLLM). ii) The L1 row contains the only *convergent* pair in the table, and is also the row that production teams now reach for first; this is the industrial centre of gravity. iii) The L0 row is not empty and is not legacy: it is where new pluggable systems are still appearing in 2025–2026.

A subtler pattern is that architectural reachability is necessary but not sufficient. Two L2 systems can both reach DUAL on paper yet behave very differently on the $T3 \cdot I2 \cdot R1$ backchannel cell, because reaching a state is a function of design and traversing it correctly is a function of training data. We return to this gap — architecture sets capacity, training realizes it — in §8 and in the evaluation appendix (Appendix E).

4 The T × I × R Interaction Ontology

The L0–L3 hierarchy of §3 organizes systems by *where* the duplex decision is made; it does not yet say *which* interactions the system should handle. We decompose a full-duplex interaction moment into a triple $(\mathrm{T}, \mathrm{I}, \mathrm{R})$: the *temporal relation* between user and system audio, the *communicative intent* of the user's contribution, and the *required system response*. Table 2 lists all values on each axis. The three axes are deliberately orthogonal — timing, meaning, and policy are distinct design questions, and conflating them is exactly what current system descriptions do.

The temporal axis (T) ranges from clean sequential handoffs (T1) through latched, overlapping, and fully concurrent speech (T2–T4) to bilateral silence (T5) (Sacks et al., 1974; Skantze, 2021). The intent axis (I) extends standard speech-act vocabulary with the categories that matter for duplex modelling: backchannels (I2), floor-claims (I4) and floor-yields (I5), disfluencies (I6), and third-party audio (I7). The response axis (R) names the model's policy choice. The Cartesian product gives 210 nominal cells, most of which are physically impossible (e.g., R1 "continue" is meaningless when the system is not speaking) or pragmatically redundant, leaving a few dozen interaction types that a full-duplex system might face.

Six acid-test cells. We single out six cells that exercise every axis and reveal current failure modes: i) **Standard** $(T1, I1, R6)$, the cas-

Layer	I	L	S	W	D	Representative systems
L0 Module	✓	✓	✓	△	·	FireRedChat, FlexDuo, X-Talk, SoulX-Duplug
L1 Hidden	✓	✓	✓	✓	△	MinMo, Freeze-Omni, Fun-Audio-Chat, Covo-Audio
L2 Token	✓	✓	✓	✓	✓	Moshi, LSLM, OmniFlatten, SyncLLM, Mini-Omni2, DuplexMamba (Mamba/SSM), SALMONN-omni (codec-free)
L3 Repr.	*— not yet realized —*					*(open architectural frontier)*

Table 1: Cross-system audit by L-layer with FSM-state reachability matrix. ✓ natively supported by construction; △ limited or degenerate; · effectively unreachable. I/L/S/W/D = IDLE/LISTEN/SPEAK/WAIT/DUAL. Row colours match the L0–L3 panels of Figure 2. Only systems that explicitly claim full-duplex behavior are listed; Qwen2.5/3.5-Omni's Thinker–Talker (L1-shaped) and GLM-4-Voice (L2-shaped) target streaming generation rather than FD (see §3.1).

T (Temporal)	I (User Intent)	R (System Response)
T1 Sequential	I1 Information	R1 Continue
T2 Latched	I2 Backchannel	R2 Stop
T3 Overlap	I3 Repair	R3 Wait
T4 Concurrent	I4 Floor-claim	R4 Backchannel
T5 Silence	I5 Floor-yield	R5 Ignore
	I6 Disfluency	R6 Initiate
	I7 Third-party	

Table 2: The three orthogonal axes of the T × I × R interaction ontology. Every full-duplex interaction is a triple (T_i, I_j, R_k).

caded baseline; ii) **Latched zero-gap** $(T2, I1, R6)$, which stresses EoT latency; iii) **Cooperative barge-in** $(T3, I4, R2)$, the cell GPT-4o made famous; iv) **Backchannel during system speech** $(T3, I2, R1)$, where most systems mis-classify the acknowledgement as barge-in and stop — the single most-cited failure mode in user feedback; v) **Third-party speech** $(T3, I7, R5)$, requiring speaker-conditioned VAD or robust speaker discrimination; vi) **Hesitation with long silence** $(T5, I6, R3)$, requiring semantic end-of-turn prediction. Figure 4 (Appendix C) concretizes each on User/Assistant audio timelines. To help readers form an intuitive understanding of the interaction logic of each of the six acid-test cells, we have also built a companion interactive demonstration website that plays back representative audio examples and visualizes the corresponding (T, I, R) traces in real time.[2]

The ontology does triple duty across the rest of the paper: as **data slices** (cell-coverage metric for training corpora, §6), as **evaluation criteria** (the Full-Duplex-Bench series (Lin et al., 2026b) aligns its overlap scenarios with specific cells, §7), and as **ablation targets** for the cross-system audit of §3. A practical consequence is that the ontology makes negative results legible: "system X fails on $T3 \cdot I2 \cdot R1$" is a useful claim; "system X is not always full-duplex" is not. Prior efforts either cover human–human dialogue without the system-response axis (Sacks et al., 1974), list capabilities without an axis structure (Ji et al., 2024; Chen and Yu, 2025), or project onto the modular L0 setting only (Li et al., 2026).

5 The Full-Duplex Decision State Machine

The hierarchy of §3 names *where* the duplex decision is made and the ontology of §4 names *which* interactions the system faces. The decision state machine names what the system is *doing* at each instant. Earlier FSM formulations are tightly tied to a particular system: a two-state SPEAK/LISTEN machine (Wang et al., 2024a) and FlexDuo's three-state extension (Liao et al., 2025).[3] We extend that lineage along two axes: we add a WAIT state (semantic hesitation handling) and a DUAL state (overlap-aware integration), and we couple every state and transition to a specific (T, I, R) ontology cell so that the FSM works as a cross-system survey instrument.

5.1 Five States, Eleven Transitions

We model a full-duplex system as occupying, at every instant, one of five states:

- IDLE — neither user nor system is producing speech.
- LISTEN — only the user is speaking; the system is silently encoding.

[2] https://duplexlm.github.io/DuplexLM/demo.html

[3] FlexDuo's IDLE is a buffer entered *while still speaking* to filter noise; in our framework this corresponds to a DUAL self-loop under an $I7$ trigger with $R5$, not to our classical IDLE.

- SPEAK — only the system is speaking; user audio is silent or below a noise floor.
- WAIT — the system has paused inside its own turn (after a hesitation or unfinished user utterance) and is monitoring before resuming.
- DUAL — both speakers are producing audio with active mutual integration (a sustained backchannel during system speech, a third-party voice the system is filtering out, or true concurrent speech).

Eleven transitions connect these five states; the full transition graph is given in Figure 3. Each transition is labelled with the (T, I) trigger that causes it and the resulting R action. The transitions split naturally into three families. i) *Onset transitions* (IDLE→LISTEN, IDLE→SPEAK) are driven by VAD or by a proactive-initiation timer (R6). ii) *Turn-handoff transitions* (LISTEN→SPEAK, SPEAK→IDLE, WAIT→SPEAK) are gated by end-of-turn prediction in the T1 / T2 regime and by semantic-EOT confirmation when the user's turn includes I6 disfluency. iii) *Overlap transitions* (SPEAK→DUAL, DUAL→SPEAK, DUAL→LISTEN) are the duplex-specific transitions that distinguish a true full-duplex system from a half-duplex one; the destination depends on the intent I of the overlapping audio (I2 backchannel stays in DUAL or returns to SPEAK; I4 floor-claim transitions to LISTEN; I7 third-party noise is ignored and the system returns to SPEAK).

Composing with the ontology. Each ontology cell corresponds to a small set of transitions: $(T3, I4, R2)$ cooperative barge-in is SPEAK→DUAL→LISTEN; $(T3, I2, R1)$ backchannel is a brief SPEAK→DUAL→SPEAK excursion; $(T3, I7, R5)$ third-party is the same shape with a different intent classification; $(T5, I6, R3)$ hesitation is LISTEN→WAIT→LISTEN. The state machine therefore turns "what does system X do in cell Y?" into a structured lookup, and surfaces a per-system constraint the ontology alone cannot: which transitions a given L-layer can architecturally attempt. Per-system FSM traces (Moshi, MinMo, FireRedChat, OmniFlatten) are deferred to Appendix F.

6 The Data Bottleneck

The audit of §3 shows that architectural reachability is necessary but not sufficient: two L2 systems can both reach DUAL on paper yet behave very differently on the $T3 \cdot I2$ backchannel cell. We argue that across the L0–L2 systems audited, the binding constraint on system-level full-duplex behavior is not the architectural choice but the cell-coverage distribution of training data. Three pieces of evidence motivate this thesis.

First, within the L2 row of Table 1, the five token-level FD systems (Moshi, LSLM, OmniFlatten, SyncLLM, Mini-Omni2) are architecturally eligible to reach the same cells, but their actual cell coverage (Table 3, Appendix C.7) varies substantially without aligning to sub-design choices: OmniFlatten and SyncLLM use different sub-designs but both claim $T3 \cdot I2$ and $T4$ coverage; Mini-Omni2 shares OmniFlatten's flattened family but claims neither. The aligning factor is what data each system was post-trained on. *Second*, Moshi's paper attributes a large fraction of its turn-taking competence to fine-tuning on Fisher (Cieri et al., 2004; Défossez et al., 2024); removing Fisher demonstrably degrades behavior despite no architectural change. *Third*, MinMo's L1 predictor is a one-layer Transformer that reaches usable barge-in / backchannel behavior largely because of the 4000-hour dialogue mix (3000 h real, 1000 h simulated) that the team annotated with heuristic turn-taking labels (Chen et al., 2025b); comparable predictors without the same data scale do not.

Two-track structural asymmetry. Public training data (∼5 k h total) is dominated by Fisher, Switchboard (Godfrey et al., 1992), and Call-Home telephone audio; recent additions are CANDOR (Reece et al., 2023) (Zoom calls), MagicData-RAMC (Yang et al., 2022) (Mandarin mobile), Easy Turn (Li et al., 2026) (turn-class labels), and HumDial Track 2 (Wang et al., 2026a) (the first public corpus with explicit $T3 \cdot I7$ third-party sub-tasks). Industrial proprietary data is widely believed to be one to two orders of magnitude larger and biased toward the deploying product's domain (e.g., call-center conversations are rich in $T3/T4$ overlaps); it is closed and non-reproducible. The asymmetry directly explains the open-source-versus-industrial gap in published FD behavior; Appendix D gives the full corpus inventory and the cell-by-cell coverage matrix.

What the data audit predicts. The cells systematically under-served in public training data — $T4$ sustained concurrent speech and (until 2026) $T3 \cdot I7$ third-party voice — are the same cells on which

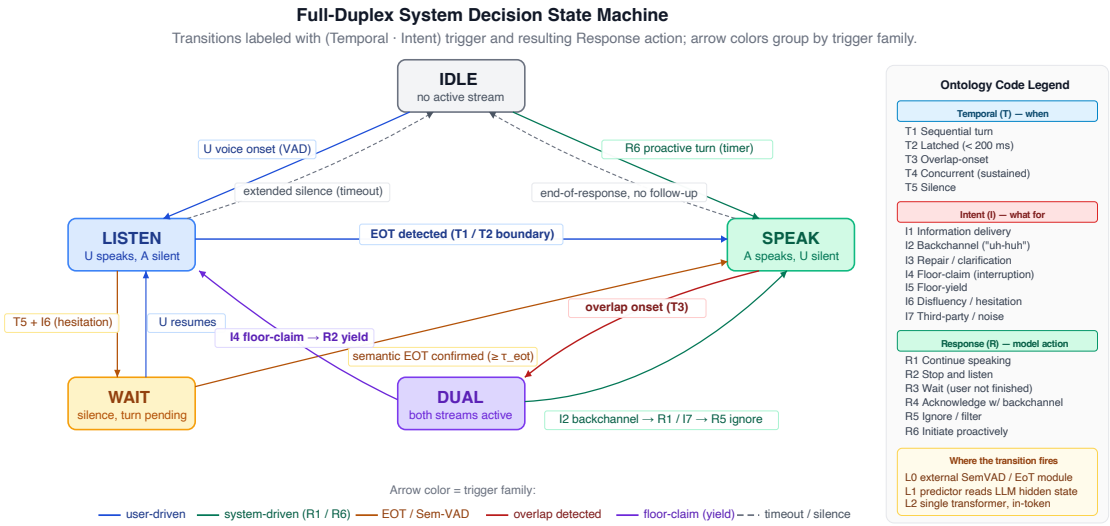

Figure 3: The full-duplex decision state machine: five states and eleven transitions. Each transition is labelled by its (T, I) trigger and the resulting R action; arrow colors group transitions by trigger family.

every audited system fails to demonstrate robust performance in current evaluations. The forthcoming HumDial Track 2 release closes the $T3 \cdot I7$ gap by design but leaves $T4$ open; we treat $T4$ as the primary residual data frontier and identify five concrete data directions (synthesis pipelines, third-party injection, cross-dataset alignment, real $T4$ collection, RL-with-conversational-rewards) in Appendix D.6.

The thesis is bounded: it applies at L0–L2 (the L3 frontier is an architecture problem, not a data problem; see §8); it claims that data sets the *ceiling* of behavior while architecture sets the gain; and it is conditional on the current evaluation regime, which the next section discusses.

7 Evaluation and the Realization Gap

Conventional speech metrics fail in the full-duplex regime by construction: WER measures the ASR module, MOS the TTS module, BLEU the response generator, and end-of-turn latency the scheduler — each assumes that dialogue consists of discrete user-then-system events, and none answers the question that defines a full-duplex system, "what should the system do per-frame while the user is still speaking?". WER does not test graceful stopping when interrupted; MOS does not test floor-claim avoidance; end-of-turn latency presupposes a turn boundary that does not exist on the $T4$ concurrent cell. The benchmarks reviewed below are the response to this gap.

Two FDB series, not one. We disambiguate two active benchmark lines with similar names. **Full-Duplex-Bench (FDB)** (Lin et al., 2025b, 2026b, 2025a, 2026a) is a versioned suite (v1–v3) evolving from passive overlap tests to active multi-turn probes, notably revealing that all audited systems fail on $T4$ tracking (mirroring our §6 data-bottleneck findings). Conversely, **FD-Bench** (Peng et al., 2025) is an independent, configurable scenario generator. We defer details on other recent resources (Arora et al., 2025; Zhang et al., 2025a; Li et al., 2026) and the ICASSP 2026 HumDial challenge to Appendix E, alongside our per-benchmark cell-coverage matrix.

The realization gap. The hierarchy of §3 describes a system's *architectural capacity* — what cells the design admits in principle. The benchmarks above measure *demonstrated behavior* — what cells the trained system actually serves. The discrepancy is what we call the *realization gap*, and it is the operative reading of the data-architecture-evaluation chain. The gap is empirically non-trivial: Mini-Omni2 and OmniFlatten both belong to the L2 flattened family with identical architectural reach, but the former demonstrates neither $T3 \cdot I2$ nor $T4$ on FDB while the latter claims both. The cause is downstream of architecture and upstream of evaluation — it is the training-data cell coverage of §6. We therefore recommend that every full-duplex system report a three-component profile: (i) its L-layer, (ii) the cell-coverage profile of its training data, and (iii) its demonstrated per-cell

behavior on the FDB v1.5 / v2 protocol. A single aggregate score collapses these three into a number that obscures the diagnostic question "is the system failing because of architecture, data, or evaluation construct?".

Structural gaps in the current regime. Three regime-level gaps remain. First, almost all benchmark audio is static and pre-recorded: even FDB v2's examiner follows a scripted task family rather than a genuinely adaptive adversary. Second, ∼95% of public test audio is English under telephone or Zoom acoustics; industrial mobile-microphone and Mandarin conditions are not standardized. Third, no public benchmark currently includes an explicit causal-influence probe distinguishing substantive from apparent full-duplex — chunk-alternating designs can score well on stop-latency without truly integrating user audio mid-turn. We treat the third gap as the most actionable, and Appendix E sketches what such a probe would look like.

8 Frontiers and Conclusion

Sections §6 and §7 identify $T4$ sustained concurrent speech as the primary residual gap in the L0–L2 frame, unserved by current models due to its absence from public corpora and benchmark metrics. While this gap is theoretically closable by data alone, a deeper architectural frontier remains: the L3 row of Table 1 is empty. We treat this as the field's main forward-looking hypothesis.

8.1 The L3 Hypothesis

L3 would dissolve the token-level boundary between perception and generation and make the DUAL state native to a shared latent representation rather than negotiated through discrete codebooks. Three pieces of upstream technical evidence make this concrete enough to be a hypothesis rather than a placeholder, and we list three candidate architectures the survey treats as live options.

(i) Continuous-latent autoregressive streaming. MAR (Li et al., 2024) shows an autoregressive Transformer can predict continuous-valued tokens via a small diffusion/flow-matching head (Lipman et al., 2022), removing the K-tokens-per-frame bottleneck of every L2 sub-design (Appendix A.3). Transposed to dialogue, user and assistant audio would share a continuous-latent stream and DUAL would be "two streams active" in the same latent — not a separate mechanism. Kimi-Audio's streaming flow-matching detokenizer (Ding et al., 2025) and the codec-free SALMONN-omni (Yu et al., 2026) are partial synthesis-side precursors.

(ii) JEPA-style dual-stream prediction. The JEPA family (Assran et al., 2023; Bardes et al., 2024) learns representations by predicting masked latent regions from visible context rather than reconstructing raw signals. A dual-stream audio JEPA — assistant latent predicted from joint user-plus-assistant context — would natively integrate $T3 \cdot I*$ overlap rather than decide stop/yield on top of a single-speaker stream, and would inherit the self-supervised property that the model learns full-duplex dynamics from unlabeled two-channel audio without explicit turn-taking annotations.

(iii) World-model-conditioned dialogue. Generative-world-model work (Bruce et al., 2024) learns a latent encoding the controllable state of the world rather than the surface signal. Applied to dialogue, the latent would encode the dialogue state (who holds the floor, what was just said, what is expected next) and the audio surface would be conditionally generated; IDLE/LISTEN/SPEAK/WAIT/DUAL of §5 would correspond to latent regions, not external state variables.

Obstacles. Three obstacles keep all three unrealized: (a) the discrete-tokenizer ecosystem is now a deep substrate that continuous-latent designs must displace; (b) streaming flow-matching and continuous-AR are not yet real-time at the 200–500 ms latency current L2 systems achieve; (c) no agreed formulation of what an L3 dialogue latent should encode exists, so the three hypotheses remain candidate framings rather than unified theory.

Conclusion

This survey contributes three complementary frameworks (L0–L3 hierarchy, T×I×R ontology, five-state machine) grounded by three empirical audits (architecture, data, evaluation). Together, they replace the binary "is this full-duplex?" question with structured profiles that situate systems, align benchmarks, and expose specific gaps like the realization gap. We hope this shared vocabulary brings clarity to the recent proliferation of systems (§2), providing a concrete starting point for next-generation corpora, evaluations, and L3 architectures.

Limitations

Two limitations bound our claims. First, our per-system and cell-coverage audits rely on self-reported behaviors rather than independent end-to-end reproductions, compounded by the fact that several industrial L1 systems are closed-source. Second, our frameworks are calibrated against existing L0–L2 architectures. What a genuine L3 system entails remains open, and our candidate architectures (§8.1) are provisional extrapolations. If L3 ultimately introduces novel interaction paradigms, our ontology and state machine will need extension rather than mere application.

References

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

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. 2024. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and 1 others. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, and 1 others. 2024. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*.

Jean Carletta. 2007. Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus. *Language Resources and Evaluation*, 41(2):181–190.

Junjie Chen, Yao Hu, Junjie Li, Kangyue Li, Kun Liu, Wenpeng Li, Xu Li, Ziyuan Li, Feiyu Shen, Xu Tang, and 1 others. 2025a. Fireredchat: A pluggable, full-duplex voice interaction system with cascaded and semi-cascaded implementations. *arXiv preprint arXiv:2509.06502*.

Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, and 1 others. 2025b. Minmo: A multimodal large language model for seamless voice interaction. *arXiv preprint arXiv:2501.06282*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Yuxuan Chen and Haoyuan Yu. 2025. From turn-taking to synchronous dialogue: A survey of full-duplex spoken language models. *arXiv preprint arXiv:2509.14515*.

Chung-Ming Chien, Manu Orsini, Eugene Kharitonov, Neil Zeghidour, Karen Livescu, and Alexandre Défossez. 2026. Moshirag: Asynchronous knowledge retrieval for full-duplex speech language models. *arXiv preprint arXiv:2604.12928*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The fisher corpus: A resource for the next generations of speech-to-text. In *LREC*, volume 4, pages 69–71.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.

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

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, and 1 others. 2025. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, and 1 others. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2025. Llama-omni: Seamless speech interaction with large language models. In *International Conference on Learning Representations*, volume 2025, pages 57607–57624.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520. IEEE.

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

Ailin Huang, Boyong Wu, Bruce Wang, Chao Yan, Chen Hu, Chengli Feng, Fei Tian, Feiyu Shen, Jingbei Li, Mingrui Chen, and 1 others. 2025. Step-audio: Unified understanding and generation in intelligent speech interaction. *arXiv preprint arXiv:2502.11946*.

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, and 1 others. 2024. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23802–23804.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and 1 others. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, and 1 others. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Ruiqi Li, Ziang Zhang, and 1 others. 2025. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In *International Conference on Learning Representations*, volume 2025, pages 93809–93826.

Chunxiang Jin, Minghui Yang, and Zujie Wen. 2021. Duplex conversation in outbound agent system. In *Interspeech*, pages 4866–4867.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. 2023. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36:27980–27993.

Yaniv Leviathan and Yossi Matias. 2018. Google duplex: An ai system for accomplishing real-world tasks over the phone. *Google AI blog*, 8.

Guojian Li, Chengyou Wang, Hongfei Xue, Shuiyuan Wang, Dehui Gao, Zihan Zhang, Yuke Lin, Wenjie Li, Longshuai Xiao, Zhonghua Fu, and 1 others. 2026. Easy turn: Integrating acoustic and linguistic modalities for robust turn-taking in full-duplex spoken dialogue systems. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16957–16961. IEEE.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. 2024. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445.

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

Guan-Ting Lin, Chen Chen, Zhehuai Chen, and Hung-yi Lee. 2026a. Full-duplex-bench-v3: Benchmarking tool use for full-duplex voice agents under real-world disfluency. *arXiv preprint arXiv:2604.04847*.

Guan-Ting Lin, Shih-Yun Shan Kuan, Jiatong Shi, Kai-Wei Chang, Siddhant Arora, Shinji Watanabe, and Hung-yi Lee. 2025a. Full-duplex-bench-v2: A multi-turn evaluation framework for duplex dialogue systems with an automated examiner. *arXiv preprint arXiv:2510.07838*.

Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, Shinji Watanabe, and Hung-yi Lee. 2026b. 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)*, pages 19447–19451. IEEE.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H Liu, and Hung-yi Lee. 2025b. Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. *arXiv preprint arXiv:2503.04721*.

Ting-En Lin, Yuchuan Wu, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022. Duplex conversation: Towards human-like interaction in spoken dialogue systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3299–3308.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Zhanxun Liu, Yifan Duan, Mengmeng Wang, Pengchao Feng, Haotian Zhang, Xiaoyu Xing, Yijia Shan, Haina Zhu, Yuhang Dai, Chaochao Lu, and 1 others. 2025. X-talk: On the underestimated potential of modular speech-to-speech dialogue system. *arXiv preprint arXiv:2512.18706*.

Xiangyu Lu, Wang Xu, Haoyu Wang, Hongyun Zhou, Haiyan Zhao, Conghui Zhu, Tiejun Zhao, and Muyun Yang. 2025. Duplexmamba: Enhancing real-time speech conversations with duplex and streaming capabilities. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 62–74. Springer.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2025. Language model can listen while speaking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24831–24839.

Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoit Sagot, Abdelrahman Mohamed, and 1 others. 2023. Generative spoken dialogue language modeling. *Transactions of the Association for Computational Linguistics*, 11:250–266.

OpenAI. 2024. Hello GPT-4o. OpenAI announcement, accessed 2026-05-18.

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

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

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

Rajarshi Roy, Jonathan Raiman, Sang-gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. 2026. Personaplex: Voice and role control for full duplex conversational speech models. *arXiv preprint arXiv:2602.06053*.

Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *language*, 50(4):696–735.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Qwen Team. 2026. Qwen3.5-omni technical report. *arXiv preprint arXiv:2604.15804*.

Tongyi Fun Team, Qian Chen, Luyao Cheng, Chong Deng, Xiangang Li, Jiaqing Liu, Chao-Hong Tan, Wen Wang, Junhao Xu, Jieping Ye, and 1 others. 2025. Fun-audio-chat technical report. *arXiv preprint arXiv:2512.20156*.

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

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Wei Xia, and Yuanjun Xiong. 2024a. A full-duplex speech dialogue scheme based on large language model. *Advances in Neural Information Processing Systems*, 37:13372–13403.

Shuiyuan Wang, Zhixian Zhao, Hongfei Yue, Chengyou Wang, Shuai Wang, Hui Bu, Xin Xu, and Lei Xie. 2026a. Humdial-eibench: A human-recorded multi-turn emotional intelligence benchmark for audio language models. *arXiv preprint arXiv:2604.11594*.

Wenfu Wang, Chenxing Li, Liqiang Zhang, Yiyang Zhao, Yuxiang Zou, Hanzhao Li, Mingyu Cui, Hao Zhang, Kun Wei, Le Xu, and 1 others. 2026b. Covo-audio technical report. *arXiv preprint arXiv:2602.09823*.

Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. 2024b. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*.

Donghang Wu, Haoyang Zhang, Jun Chen, Hexin Liu, Eng Siong Chng, Fei Tian, Xuerui Yang, Xiangyu Zhang, Daxin Jiang, Gang Yu, and 1 others. 2025. Mind-paced speaking: A dual-brain approach to real-time reasoning in spoken language models. *arXiv preprint arXiv:2510.09592*.

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

Zhifei Xie and Changqiao Wu. 2024b. Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities. *arXiv preprint arXiv:2410.11190*.

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. 2025. Qwen2.5-omni technical report. *Preprint*, arXiv:2503.20215.

Ruiqi Yan, Wenxi Chen, Zhanxun Liu, Ziyang Ma, Haopeng Lin, Hanlin Wen, Hanke Xie, Jun Wu, Yuzhe Liang, Yuxiang Zhao, and 1 others. 2026. Soulx-duplug: Plug-and-play streaming state prediction module for realtime full-duplex speech conversation. *arXiv preprint arXiv:2603.14877*.

Zehui Yang, Yifan Chen, Lei Luo, Runyan Yang, Lingxuan Ye, Gaofeng Cheng, Ji Xu, Yaohui Jin, Qingqing Zhang, Pengyuan Zhang, and 1 others. 2022. Open source magicdata-ramc: A rich annotated mandarin conversational (ramc) speech dataset. *arXiv preprint arXiv:2203.16844*.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. 2026. Salmonn-omni: A standalone speech llm without codec injection for full-duplex conversation. *Advances in Neural Information Processing Systems*, 38:24617–24643.

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

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

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, and 1 others. 2024. Anygpt: Unified multimodal llm with discrete sequence modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9637–9662.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773.

He Zhang, Wenqian Cui, Haoning Xu, Xiaohui Li, Lei Zhu, Haoli Bai, Shaohua Ma, and Irwin King. 2025a. Mtr-duplexbench: Towards a comprehensive evaluation of multi-round conversations for full-duplex speech language models. *arXiv preprint arXiv:2511.10262*.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chao-Hong Tan, Zhihao Du, and 1 others. 2025b. Omniflatten: An end-to-end gpt model for seamless voice conversation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14570–14580.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2024a. Speechtokenizer: Unified speech tokenizer for speech language models. In *International Conference on Learning Representations*, volume 2024, pages 31798–31818.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Xu Han, Zihang Xu, Yuanwei Xu, Weilin Zhao, Maosong Sun, and Zhiyuan Liu. 2024b. Beyond the turn-based game: Enabling real-time conversations with duplex models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11543–11557.

A Foundations: Audio Tokenizer and Encoder Lineage

The L2 token-level full-duplex systems audited in Appendix C.7 all share an upstream dependency: they assume audio has already been transformed into a representation a Transformer language model can consume. The form of that representation — which tokenizer, with what frame rate, how many codebooks, and whether the first codebook is shared with text — determines what the L2 architecture can express. This appendix provides the minimum tokenizer and encoder background needed to read the per-system breakdowns in Appendix C.7 and the sub-design discussions in §3.2. It is not a tokenizer survey; it identifies the three pieces of upstream infrastructure on which the rest of the paper rests.

A.1 Three Routes from Audio to LLM

Three structurally distinct routes have been used to deliver speech into a Transformer LLM, and each maps onto a recognizable line of L2 systems:

- **Discrete audio tokens.** The waveform is quantized into a sequence of integer tokens that the LLM can predict the same way it predicts text tokens. This route is the most-traveled because it reuses the LLM training and inference stack without modification; it is what underlies Mini-Omni (Xie and Wu, 2024a), Moshi (Défossez et al., 2024), OmniFlatten (Zhang et al., 2025b), GLM-4-Voice (Zeng et al., 2024), and SyncLLM (Veluri et al., 2024).
- **Continuous audio embeddings.** A streaming speech encoder (e.g., a Whisper-encoder or a custom SSL encoder) outputs continuous vectors that are aligned to the LLM's

hidden-state space via a projector. LLaMA-Omni (Fang et al., 2025) is the canonical example; the route preserves more acoustic information than tokenization but requires special handling on the input side of the LLM.

- **Hybrid text-leads / audio-trails.** A text token stream drives generation, with audio tokens emitted in parallel or shortly thereafter conditioned on the text. Moshi's Inner Monologue (Défossez et al., 2024) is the load-bearing instance: the text track provides the semantic backbone and the audio tracks attach acoustic realization to it.

Routes (i) and (iii) together account for essentially all L2 systems in Appendix C.7; route (ii) is the natural home for the codec-free designs (SALMONN-omni (Yu et al., 2026) brought this route into a full-duplex setting in 2026).

A.2 Three Generations of Audio Tokenizers

Discrete tokenization for speech has gone through three generations, each defined by what it optimizes for and what it loses. The L2 explosion of 2024 is downstream of the third generation reaching engineering maturity.

Generation 1 — semantic units. HuBERT (Hsu et al., 2021), wav2vec 2.0 (Baevski et al., 2020), w2v-BERT (Chung et al., 2021), and WavLM (Chen et al., 2022) (2020–2021) trained self-supervised speech encoders, with discrete tokens obtained via k-means clustering over hidden states at roughly 25–50 Hz. Whisper (Radford et al., 2023) is the dominant supervised counterpart used as a streaming encoder for the continuous-embedding route (§A.1, route (ii)). Property: semantically faithful (the unit sequence preserves what was said) but acoustically lossy (reconstruction loses speaker identity and prosody). Suitable for understanding tasks; unsuitable as the sole representation when the system must also speak naturally. SpeechGPT (Zhang et al., 2023) and AnyGPT (Zhan et al., 2024) are canonical generation-1 LLM-coupled systems.

Generation 2 — neural audio codecs. SoundStream (Zeghidour et al., 2021), EnCodec (Défossez et al., 2022), and DAC (Kumar et al., 2023) reframed audio tokenization as a signal-compression problem: an encoder–quantizer–decoder is trained with residual vector quantization (RVQ) to compress the waveform to roughly 1–3 kbps with high reconstruction fidelity. A typical configuration is 75 Hz frame rate with 8 RVQ codebooks per frame. AudioLM (Borsos et al., 2023) and VALL-E (Wang et al., 2023) are the canonical generation-2 LM-coupled systems on the generation side. Property: acoustically faithful but semantically weak — the eight codebooks of a single frame have no canonical ordering by importance, so an LLM trained on the flattened token stream learns acoustic detail before semantic content, hurting downstream language behavior.

Generation 3 — semantic–acoustic fused. SpeechTokenizer (Zhang et al., 2024a), Mimi (the codec used by Moshi (Défossez et al., 2024)), and the CosyVoice tokenizer (Du et al., 2024) added an explicit semantic-then-acoustic ordering to the RVQ stack: the first codebook is distilled or constrained to carry semantic content (in Mimi's case, distilled from WavLM (Chen et al., 2022)), and the remaining codebooks carry acoustic residuals. WavTokenizer (Ji et al., 2025) represents the parallel single-codebook line that compresses to similarly low bitrates without the multi-codebook structure. Property: the first codebook of generation 3 behaves approximately like a text token, which lets an LLM read or write speech with the same training pipeline as text. Mimi specifically operates at 12.5 Hz with 8 codebooks, totalling 1.1 kbps — low enough that an autoregressive Transformer can produce real-time speech. Generation 3 is the upstream condition for Moshi and for the broader L2 token-level wave of 2024; without it, the L2 row of the audit table in §3.2 would not exist.

A.3 The K-Tokens-per-Frame Problem

A direct consequence of generations 2 and 3 both using RVQ is that every audio frame produces K tokens rather than one. For Mimi, $K = 8$. A naive autoregressive treatment serializes the eight codebooks along the time axis, multiplying the sequence length and the inference cost by eight; for ten seconds of audio at 12.5 Hz, this turns 125 frames into 1000 tokens. The L2 sub-designs surveyed in §3.2 are, to a first approximation, different answers to "how should the K codebooks of each frame be arranged in the autoregressive sequence?" Moshi's multi-stream RQ-Transformer predicts all K codebooks at the same time step using a small inner Transformer; OmniFlatten flattens the codebooks along time; SyncLLM packs each chunk's tokens into a fixed-length block. These are not architec-

tural curiosities at the LLM level; they are direct responses to the K-tokens-per-frame constraint imposed by the tokenizer.

A.4 Two Recent Departures

Two 2025–2026 directions broaden the tokenizer assumption space and matter for the L2 design discussion in §3.2.

Flow-matching streaming detokenizer. Kimi-Audio (Ding et al., 2025) decouples the LLM input side from the LLM output side: the LLM continues to consume discrete tokens, but the audio synthesis side uses a chunk-wise streaming flow-matching detokenizer rather than an RVQ-decoder. This widens the design space because the LLM no longer has to predict the same token representation it consumes, and naturalness on the output side can be improved without re-training the LLM. The pattern is too new to have settled into a sub-design but already appears in several 2026 system reports.

Codec-free representations. SALMONN-omni (Yu et al., 2026) drops the audio codec entirely and operates on continuous embeddings throughout, bringing route (ii) of §A.1 into the L2 design space. The trade-off is the inverse of generation 3 tokenization: the model retains more acoustic detail at the cost of losing the "audio as language" framing that made the LLM training pipeline reusable in the first place. SALMONN-omni's appearance is what makes the L2 row of §3.2 structurally heterogeneous rather than uniformly token-based.

A.5 Reading Guide

For the rest of the paper, three takeaways from this appendix are load-bearing. *First*, when an L2 system is described as "token-level full-duplex", the precise meaning depends on which generation of tokenizer it uses: generation 3 lets the same Transformer carry semantic and acoustic content, which is why the L2 wave clusters in late 2024 rather than earlier. *Second*, the L2 sub-design names used informally in §3.2 (multi-stream, flatten, chunk-alternation) are upstream-determined by the K-tokens-per-frame constraint, not invented top-down at the LLM level. *Third*, route (ii) (continuous embeddings) and the codec-free designs are not exceptions to the L2 framing of §3.2 but legitimate L2 systems whose decision point is on the output token stream of a non-RVQ representation. Throughout the per-system breakdowns of Appendix C.7 we use "L2 token-level" to mean the decision is at the discrete-output layer, whether that layer is RVQ-coded or continuous.

B Pre-Full-Duplex Era: Detailed Historical Account

This appendix expands the compressed pre-FD account of §2, including the modular pipelines that preceded the LLM era, the end-to-end speech-text wave, and the GPT-4o pivot.

B.1 Modular Pipelines Before the LLM Era

Before "full-duplex" became a banner of the LLM era, turn-taking had been a research subject for half a century and a deployed product for several years. The cascaded speech-dialogue stack (ASR→NLU→DM→NLG→TTS) treated turn-taking as a scheduling decision driven by voice activity detection: when energy crossed a threshold and stayed silent for ∼300 ms, the system declared the user's turn over and began responding. This worked acceptably for command-and-control interactions and badly for everything else.

The academic foundation traces back to Sacks, Schegloff, and Jefferson's 1974 *Language* paper (Sacks et al., 1974), which formalized turn-taking as a system of turn-constructional units and turn-allocation rules and remains the most-cited article in that journal. A modern computational lineage runs through Skantze and collaborators (Skantze, 2021), whose neural turn-taking-prediction work reframed speaker change as a continuous prediction problem rather than a VAD threshold and was the basis of much subsequent work on end-of-turn detection.

On the industrial side, the most visible pre-LLM full-duplex moment was **Google Duplex**, demonstrated by Sundar Pichai at Google I/O 2018, where an AI system phoned a hair salon and a restaurant and conducted multi-turn appointment-booking conversations in remarkably human-sounding speech (Leviathan and Matias, 2018). Duplex was never published as an academic paper but it set the popular expectation that "AI can carry on a phone conversation" six years before GPT-4o, and its underlying L0 stack — cascaded ASR + dialogue manager + TTS tuned for spontaneous timing — defined the blueprint that industrial systems would refine in the years that followed. In Asia, Ant Group's 2021 Interspeech paper (Jin et al., 2021) described a three-module pipeline

(speech-aware VAD, supervised end-of-turn classifier, response planner) running on real outbound calls, and Alibaba's DAMO Academy (Lin et al., 2022) followed in 2022 with a KDD paper that added speaker-conditioned VAD (pVAD) to filter background voices and reframed end-of-turn as a learned binomial decision over partial sentences, deployed in Alibaba's intelligent customer-service product. Together with Duplex, these systems established the L0 modular blueprint — VAD plus EoT plus dialogue manager, each a separate model — that LLM-era systems would later either inherit or argue against.

B.2 The End-to-End Wave and the GPT-4o Pivot

The first pivot was technical, not commercial: speech became a token. Discrete audio tokenizers (Zhang et al., 2024a; Défossez et al., 2022) made it possible to feed waveforms into Transformer language models the same way text is fed, and a wave of end-to-end spoken dialogue models followed. SpeechGPT (Zhang et al., 2023) fused an LLM with speech encoding and decoding; AudioGPT (Huang et al., 2024) extended the multimodal recipe; dGSLM (Nguyen et al., 2023) — chronologically earliest, in 2022 — showed that two parallel speech streams could be jointly modelled without text at all, learning turn-taking rhythm and even laughter from the Fisher corpus. None of these were marketed as "full-duplex" in the modern sense. Adjacent lines of work from the same window — audio-understanding LLMs such as Qwen2-Audio (Chu et al., 2024) and integrated speech-interaction systems such as the broader Step-Audio line (Huang et al., 2025) — share the token-level coupling between LLM and speech but target understanding or single-turn synthesis rather than duplex dialogue; we cite them here as historical context for the L1/L2 patterns that emerged around the same time.

The second pivot was commercial. On May 13, 2024, OpenAI's GPT-4o demonstration (OpenAI, 2024) showed conversational interactions in which users could interrupt the model mid-sentence and have it yield the floor smoothly. The wider community read this as proof that an end-to-end speech-to-speech model could be *simultaneously* listening and speaking, and a replication race followed within months: Mini-Omni (Xie and Wu, 2024a), LLaMA-Omni (Fang et al., 2025), and GLM-4-Voice (Zeng et al., 2024) all shipped open implementations claiming end-to-end speech dialogue. Despite the end-to-end framing, a strict evaluation against the substantive-FD criterion shows these models largely functioned in a half-duplex manner: they could be *interrupted* via a keyword trigger or a frontend VAD, but their decoder produced one stream at a time. Two bridge papers from this transitional window — a full-duplex LLM scheme (Wang et al., 2024a) and the Duplex-Model proposal from THUNLP (Zhang et al., 2024b) — attempted to paste a VAD-based scheduler onto an LLM without unifying the token layer; a token-level full-duplex system did not appear publicly until the next era.

B.3 The Full-Duplex Proliferation: Detailed System Stories

The third era opened with two papers one month apart. In August 2024, LSLM (Ma et al., 2025) demonstrated that a decoder-only TTS model could fuse a streaming SSL encoder in real time, with middle-layer fusion outperforming early or late variants. Kyutai publicly unveiled Moshi at a Paris launch event on July 3, 2024, and released both an interactive web demo and, in September 2024, the model weights and technical report (Défossez et al., 2024). Moshi paired the Mimi neural audio codec (12.5 Hz, 8 RVQ codebooks) with an RQ-Transformer that emitted three simultaneous streams (user-audio prediction, assistant audio, and an "inner monologue" text track), with roughly 200 ms reported latency. Its training recipe is itself part of the historical record: pre-training on millions of hours of public audio, multi-stream post-training driven by speaker-diarized data, full-duplex fine-tuning on the Fisher corpus, and instruction tuning on a large body of TTS-synthesized dialogues (Défossez et al., 2024). Together LSLM and Moshi reset what "learned" full-duplex was expected to mean.

Within six months, the design space had fragmented. OmniFlatten (Zhang et al., 2025b) flattened four streams (user/assistant $\times$ speech/text) into a single sequence with a special silent-speech token; SyncLLM (Veluri et al., 2024) divided time into fixed chunks (160–240 ms variants evaluated, with 160 ms as the baseline) and alternated user and assistant token blocks with synchronization markers; Mini-Omni2 (Xie and Wu, 2024b) added a "Stop Omni" keyword interruption. GLM-4-Voice (Zeng et al., 2024) shipped in the same window with a single-codebook tokenizer and a trillion-

token pretraining mix; it shares the L2 architectural shape but does not explicitly claim full-duplex behavior.

Two further developments in 2025 are worth flagging because they recur in the framework discussion. First, MinMo (Chen et al., 2025b) from Alibaba's FunAudioLLM team established the L1 industrial pattern for full-duplex dialogue: an LLM backbone whose hidden state is consumed by an external duplex-decision predictor. Notably, Qwen2.5-Omni (Xu et al., 2025) (and its successor Qwen3.5-Omni (Team, 2026)) use a structurally identical pattern — their Thinker–Talker decomposes generation by having the Talker read the Thinker's hidden state — without claiming full-duplex behavior; they target streaming text+speech generation, where the "two simultaneous outputs" are text and speech modalities rather than overlapping user/assistant audio. The recurrence of the L1 architectural shape across these different product goals is what we treat as evidence that L1 is a structural attractor in §3, distinct from any specific FD claim. Second, late 2025 saw a modular comeback. FireRedChat (Chen et al., 2025a) from Xiaohongshu and FlexDuo (Liao et al., 2025) re-implemented the modular blueprint with stronger components; X-Talk (Liu et al., 2025) from SJTU's X-LANCE group went further and published an explicit advocacy paper arguing that the progression toward token-level designs is not the only viable trajectory, and that modular speech-to-speech systems deserve a serious long-term seat at the table. The era is not closed; it is contested.

Through late 2025 and 2026 the proliferation continued at industrial scale and along several non-mainstream directions. Fun-Audio-Chat (Team et al., 2025) from the Tongyi Fun team extended the MinMo predictor lineage with a larger dialogue model and richer multi-turn supervision; Covo-Audio (Wang et al., 2026b) appeared as a parallel industrial system in early 2026; SoulX-Duplug (Yan et al., 2026) re-cast the modular blueprint as a plug-and-play streaming state-prediction module that can be attached to a frozen LLM; StepFun released Step-Audio R1.1 (Wu et al., 2025), a real-time variant built around a dual-brain articulation–reasoning split. Two further L2 systems from this window broaden the L2 design space along axes that earlier L2 systems took as fixed: DuplexMamba (Lu et al., 2025) replaces the Transformer decoder with a Mamba / state-space-model backbone, and SALMONN-omni (Yu et al., 2026) drops the audio codec entirely and operates on continuous embeddings. Work like PersonaPlex (Roy et al., 2026) (voice/role control) and MoshiRAG (Chien et al., 2026) (asynchronous knowledge retrieval) extends dialogue capabilities beyond the core full-duplex setting and lies outside the scope of this survey.

C Six Acid-Test Cells: Per-Cell Behavior Detail

This appendix expands each of the six canonical full-duplex interaction scenarios of Figure 4, describing the expected system behavior, common failure modes, and the architectural mechanisms typically required.

C.1 Standard Turn $(T1, I1, R6)$

Behavior. Clean handoff at a transition-relevance place (TRP), with a 200–600 ms inter-turn gap.
Coverage. Every system since cascaded SDS handles this; it is the baseline of any duplex evaluation.

C.2 Latched Zero-Gap $(T2, I1, R6)$

Behavior. Near-zero handoff. The assistant must already have begun generating before the user finishes (streaming generation + early end-of-turn prediction).
Coverage. Requires semantic VAD plus speculative TTS. Moshi, MinMo, and FireRedChat reach this; cascaded baselines miss because their EOT decision happens after silence has been observed.

C.3 Cooperative Barge-In $(T3, I4, R2)$

Behavior. The user asserts the floor while the assistant is still talking. The assistant must stop within roughly 200 ms for the interaction to feel substantively full-duplex.
Pitfall. Apparent-FD systems (chunk-aligned designs, keyword-triggered interruption in command-mode systems) keep generating across the chunk boundary — they fail the causal-influence criterion (Appendix E).

C.4 Backchannel During System Speech $(T3, I2, R1)$

Behavior. Short overlapping signals from the user ("uh-huh", "mm", "*dui*") are listener support, not a floor claim — the assistant must keep going, possibly with subtle prosodic accommodation.
Pitfall. Naïve acoustic VAD treats any user voice as barge-in, fragmenting the system mid-clause.

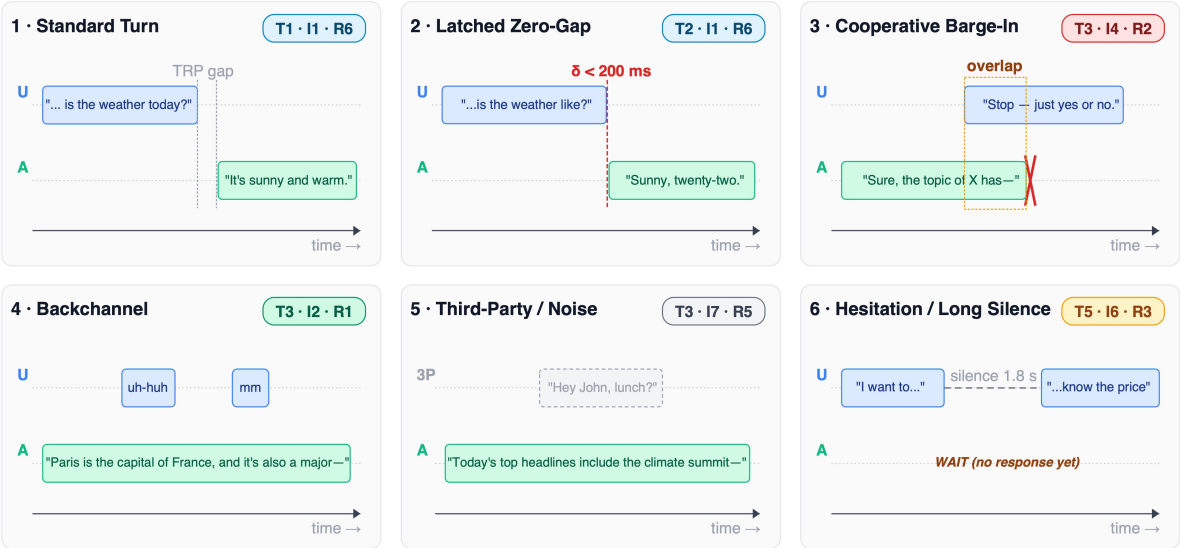

Figure 4: Six canonical full-duplex interaction scenarios (referenced from §4). Each panel concretizes one (T, I, R) ontology cell on User (U) / Assistant (A) audio timelines.

Distinguishing $I2$ from $I4$ requires intent classification, not just energy detection.

C.5 Third-Party / Noise $(T3, I7, R5)$

Behavior. Speech that overlaps the assistant but does not come from the addressed user (third party, TV, environmental speech) must be filtered out and ignored.
Mechanism. Personal VAD (pVAD), speaker embeddings, or an LLM-level relevance gate (sometimes called "semantic denoise"). FireRedChat's pVAD module and the Ant Group / Alibaba L0 pipelines target precisely this cell.

C.6 Hesitation with Long Silence $(T5, I6, R3)$

Behavior. Acoustic silence does not imply end-of-turn. Semantic content "I want to…" is incomplete; the system should wait rather than treat the silence as a turn-yielding cue.
Pitfall. Fixed-threshold VAD (e.g., ∼500 ms) fires prematurely. Semantic end-of-turn prediction (Skantze, 2021) or an LLM-conditioned predictor is required.

C.7 Per-System Architectural Detail

We organize the per-system breakdown by the L-layer at which the duplex decision is made (§3). Each entry summarizes architecture, training recipe where disclosed, and the ontology cells the system credibly handles. Approximate substantive latency δ^* is reported when published.

C.7.1 L0 Module-Level Systems

L0 systems factor the duplex decision out of the generative model into one or more lightweight modules (typically VAD, end-of-turn (EoT) classifier, dialogue manager) that run continuously alongside an LLM-based or template-based generator. The LLM-era L0 systems inherit the topology established by the 2021 Interspeech and 2022 KDD industrial papers (§B); only the generator was replaced.

FireRedChat (Chen et al., 2025a). Three serial modules drive every transition. (i) **pVAD**: a streaming personalized VAD with a speaker-condition channel, used to suppress background noise and competing speakers. (ii) **EoT**: an end-of-turn classifier over streaming ASR partial results, reaching ∼96% accuracy on Chinese and ∼95% on English in the paper's evaluation. (iii) **Dialogue Manager**: schedules the LLM/AudioLLM backend (FireRedChat supports both cascaded and semi-cascaded backends interchangeably). The pVAD covers $T3 \cdot I7$ (third-party voice); the EoT covers $T5 \cdot I6$ (semantic hesitation). $T3 \cdot I4$ (cooperative barge-in) is handled acoustically and consequently fails on the $T3 \cdot I2$ backchannel cell. Reported barge-in response time $T_{90} \approx 170$ ms; reported end-to-first-response latency $P_{50} \approx 2.34$ s, $P_{95} \approx 3.02$ s (comparable to or better than the open frameworks LiveKit and Ten).

FlexDuo (Liao et al., 2025). A pluggable streaming module that decouples duplex control from the dialogue system. Notably introduces a third state called IDLE, but with a semantics distinct from the classical no-activity state: their IDLE is a buffer the system enters *while still speaking*, used to filter noise and irrelevant audio without triggering LISTEN or yielding the floor (see §5 for our framework's disambiguation).

X-Talk (Liu et al., 2025). A position paper from the SJTU X-LANCE group arguing that the L0 modular blueprint is competitive on latency, interpretability, and engineering cost, and is not merely a transitional stage before L2. Reframes the L0 trajectory as a parallel design path rather than a legacy.

SoulX-Duplug (Yan et al., 2026). A plug-and-play streaming state-prediction module that attaches to a frozen LLM, restating the L0 modular blueprint with modern streaming components. Representative of the late-2025 modular renaissance.

Easy Turn (Li et al., 2026). Strictly a turn-taking detector rather than a full SDS: jointly fine-tuned on acoustic and linguistic modalities, predicting four states (complete / incomplete / backchannel / wait) that align directly with ontology cells $(T1, I1, R6)$, $(T5, I6, R3)$, $(T3, I2, R1)$, and $(T1, I5, R6)_{\text{delayed}}$. Releases a 1145-hour open turn-taking corpus, currently the largest of its kind, and provides a drop-in replacement for legacy EoT modules.

The L0 ceiling. All L0 systems share a structural latency floor: even when the decision module fires in $\sim$100 ms, the downstream LLM forward pass plus first-chunk TTS decoding pushes $\delta^*_{\text{respond}}$ to roughly 500 ms. This is the physical motivation for the L1 step.

C.7.2 L1 Hidden-State-Level Systems

L1 systems retain modular separation but feed the decision module the LLM's hidden state $\mathbf{h}_t$, reusing the language model's semantic competence.

MinMo (Chen et al., 2025b). Voice Encoder + Input Projector $\rightarrow$ Qwen LLM (LoRA), with the LLM's hidden state branching into (i) an Output Projector $\rightarrow$ Voice-Token LM $\rightarrow$ Token2Wav synthesis path, and (ii) a *Full-Duplex Predictor* (single-layer Transformer with linear softmax, randomly initialized) that emits a binary decision in real time. The predictor operates in two modes: when the system is silent it discriminates "respond now" vs "keep waiting" ($T1$ vs $T5 \cdot I6$); when the system is speaking it discriminates barge-in ($T3 \cdot I4$) from backchannel ($T3 \cdot I2$). Training reuses a 4000-h dialogue mix (real + simulated) under a multi-stage curriculum (speech-to-text $\rightarrow$ text-to-speech $\rightarrow$ speech-to-speech $\rightarrow$ duplex interaction alignment), with heuristic-rule auto-labeling for turn-taking events. The paper reports a speech-to-text latency of approximately 100 ms and a full-duplex latency of approximately 600 ms in theory, 800 ms in practice.

Freeze-Omni (Wang et al., 2024b). Trains audio I/O around a *frozen* LLM via a three-stage protocol (ASR, TTS, then a multi-task duplex stage on 60 k multi-turn QA samples), using $\sim$8 GPUs total. Architecturally L1, but because the LLM weights are not updated by the duplex objective the model lives on the L0/L1 boundary — a low-cost realization of the L1 pattern that trades ontology coverage for training simplicity.

Fun-Audio-Chat (Team et al., 2025). Extends the MinMo predictor lineage with a larger dialogue model and richer multi-turn supervision; same overall L1 architectural shape. Released by the Tongyi Fun team in late 2025.

Covo-Audio (Wang et al., 2026b). A parallel industrial system released in early 2026; L1-shaped within the publicly disclosed architecture.

Qwen2.5-Omni / Qwen3.5-Omni Thinker–Talker (non-FD, L1-shaped). Qwen2.5-Omni (Xu et al., 2025) and its successor Qwen3.5-Omni (Team, 2026) are not marketed as full-duplex: their *Thinker* (a text LLM) feeds its hidden representation to a *Talker* (a dual-track autoregressive audio-token model) that streams text and speech outputs concurrently — two output modalities, not overlapping user/assistant audio. The Thinker–Talker decomposition is nevertheless *architecturally identical* to MinMo's LLM–Predictor pattern (external module reading LLM hidden state), which we treat in §3 as evidence that L1 is a broader structural attractor than the FD niche alone.

L1 convergence. MinMo and Qwen2.5-Omni were proposed by different teams two months apart and arrived at structurally identical designs. To-

gether with Freeze-Omni's frozen-LLM degenerate variant and the Tongyi Fun lineage, L1 has stabilized as the industrial default for systems that want LLM-quality dialogue without committing to a full token-level redesign.

C.7.3 L2 Token-Level Systems

L2 systems eliminate the explicit decision module: turn-taking decisions are encoded in the token-level generative process itself. We organize L2 systems by their internal sub-mode (multi-stream, flatten, chunk-alternation, channel-fusion).

dGSLM (pre-history) (Nguyen et al., 2023). The first textless dual-channel speech LM: dual-tower Transformer with cross-attention, trained on the 2000-hour Fisher corpus with next-frame prediction, no text. Established that turn-taking rhythm and paralinguistic signals (laughter, sighs) can be learned end-to-end without text. Not a practical dialogue system (no instruction following) but the direct architectural ancestor of Moshi's multi-stream design.

LSLM (Ma et al., 2025). Released August 2024, one month before Moshi — the first published LLM-era dual-channel duplex system. Uses a token-based decoder-only TTS for the speaking channel and a streaming SSL encoder for the listening channel, fused at one Transformer layer (early/middle/late variants studied; middle layer optimal). The fusion-based duplex modelling is a distinct L2 sub-mode from Moshi's parallel-stream RQ-Transformer.

Moshi (Défossez et al., 2024). Combines (i) the **Mimi** neural audio codec (12.5 Hz, 8 RVQ codebooks; first codebook distilled from WavLM for semantic content, codebooks 2–8 for acoustic residuals), (ii) an **RQ-Transformer** (Temporal Transformer in the time dimension plus a Depth Transformer over the eight codebooks), and (iii) **Inner Monologue** (a text token stream temporally interleaved before the audio tokens, optionally with an acoustic delay). At every 80 ms frame, the model simultaneously emits text tokens, a predicted user-audio token, and an assistant-audio token. Because the model predicts both streams at every frame, "user begins speaking" is naturally absorbed by the user-stream prediction and the assistant stream collapses to silence tokens without any explicit VAD/EoT module. The paper reports a theoretical latency of 160 ms and approximately 200 ms in practice. Training spans pre-training on roughly 7 M hours of public audio with Whisper-transcript supervision, multi-stream post-training driven by speaker-diarized data, full-duplex fine-tuning on the Fisher corpus, and instruction tuning on synthetic dialogues generated by their own TTS.

OmniFlatten (Zhang et al., 2025b). Flattens four streams (user/assistant × speech/text) into a single sequence with a special `silent_speech_token`, then trains a standard GPT-style autoregressive model with no architectural modification, with text-stream tokens leading their corresponding speech-stream tokens. Demonstrates that full-duplex behavior does not require a bespoke decoder — careful token design over a standard LM is sufficient. The paper reports low end-to-end latency without committing to a single number.

SyncLLM (Veluri et al., 2024). Divides time into fixed-size chunks (the paper evaluates 160 ms, 200 ms, and 240 ms variants; Figure 2 of the paper uses 160 ms as the baseline) and alternates predicting the assistant's chunk and the user's chunk within a single sequence, marked by periodic synchronization tokens (`[S0]`/`[S1]`); uses deduplicated HuBERT tokens at 25 Hz with interpolation. By the causal-influence criterion of §5, transitions occur only at chunk boundaries: within a chunk, the system cannot causally respond to the user's current audio. SyncLLM is therefore the canonical example of an L2-shaped but *apparent* (not substantive) full-duplex system.

Mini-Omni / Mini-Omni2 (Xie and Wu, 2024a,b). Mini-Omni v1 generates text and audio tokens in parallel (same time step, two heads) on top of an arbitrary text LLM via the "Any Model Can Talk" adaptation recipe, trained on the VoiceAssistant-400K instruction corpus. Mini-Omni2 adds vision and a keyword-triggered interruption ("Stop Omni") with state tokens `irq`/`n-irq` for command-mode barge-in detection. Outside the trigger keyword the model has no causal response to user audio — archetypal apparent-FD behavior with strong open-source ecosystem value.

DuplexMamba (Lu et al., 2025). An L2 system that replaces the Transformer decoder with a Mamba / state-space-model backbone while keeping the duplex-in-token-sequence pattern. Demonstrates that L2 synchronization is backbone-

agnostic and not tied to attention-based architectures.

SALMONN-omni (Yu et al., 2026). A standalone full-duplex speech LLM that operates on continuous embeddings rather than discrete audio-codec tokens. The duplex decision is still token-level autoregressive (continuous-valued token stream), keeping it within L2 by our definition; the contribution is to demonstrate that the L2 pattern does not strictly require a discrete codec tokenizer.

GLM-4-Voice (non-FD, L2-shaped) (Zeng et al., 2024). A streaming end-to-end speech chatbot with a single-codebook tokenizer at 175 bps (12.5 Hz, derived from an ASR model with a VQ bottleneck), pretrained on roughly a trillion tokens through unsupervised, interleaved, and supervised stages. Architecturally adjacent to L2 with explicit turn structure preserved; the paper does not claim full-duplex behavior. Included here as evidence that the L2 architectural shape recurs even outside the FD-claiming niche.

Step-Audio R1.1 (Wu et al., 2025). A 2026 real-time variant from the Step-Audio family with a dual-brain articulation–reasoning split. Public information is limited at the time of writing; included for timeline completeness.

C.7.4 L3 Representation-Level: Why Empty

L3 would require user and assistant streams to be encoded in a shared continuous latent rather than negotiated through discrete tokens. Three obstacles keep the L3 row empty in Table 1: (i) discrete audio tokenizers (Mimi, SpeechTokenizer, X-Codec, CosyVoice 2 family) have become the de-facto ecosystem, raising the cost of switching to continuous representations; (ii) continuous-latent generative models (diffusion, flow matching) remain immature for streaming autoregressive use — recent flow-matching detokenizers operate downstream of a discrete model, not at the modelling layer; (iii) no agreed mathematical formulation of what L3 should look like exists yet. Three plausible directions are continuous audio latents with flow-matching modelling, JEPA-style dual-stream prediction, and world-model-conditioned dialogue. We discuss these in §8 as a concrete architectural agenda.

C.7.5 Cross-System Comparison Tables

Table 1 in the main text gives the L-layer / FSM-state matrix. Table 3 (this appendix) gives the system × acid-test-cell coverage matrix for the seven cells most discussed in the paper.

C.7.6 Training-Recipe Cross-Comparison

A convergent four-stage training pattern emerged across L1/L2 systems through late 2024 and 2025: (1) general text LLM pre-training (often reused from an off-the-shelf backbone such as Qwen, LLaMA, or GLM); (2) speech-to-text alignment (ASR objective); (3) text-to-speech alignment (TTS objective); (4) duplex interaction alignment on full-duplex dialogue data, optionally followed by (4b) reinforcement learning with an interaction-quality reward. The data scarcity at stage 4 is the operative constraint: Moshi reports Fisher (2000 h) plus large-scale synthesis; MinMo reports 4000 h (3000 h real, 1000 h simulated); most other systems are dominated by synthesis. Auto-labeling at the duplex stage is universally adopted in some form — MinMo's heuristic rules, OmniFlatten's silent-speech tokens, and SyncLLM's periodic anchors are three independent recipes, each enabling thousand-hour-scale training without manual turn-taking annotation. We treat this scarcity, and the strong dependence on auto-labeling, as the core of the data frontier discussed in §8 and Appendix D.

D Data: Extended Audit and Research Agenda

This appendix expands the data-bottleneck thesis stated in §6. The body section summarizes the thesis and three pieces of evidence; this appendix supplies (i) the three training-data type typology, (ii) the full corpus inventory, (iii) the public-vs-industrial track split, (iv) the synthesis-pipeline survey, (v) the cell-coverage matrix, (vi) six concrete research directions, and (vii) an explicit scope statement for the thesis.

D.1 Three Training-Data Types

We classify training data for full-duplex systems into three types by the completeness of dialogue state they carry:

- **Type A — Single-stream speech**: single-channel audio without dialogue structure (LibriSpeech, Common Voice, ASR corpora). Used at the speech-text alignment stage; sup-

System	Standard $T1I1R6$	Latched $T2I1R6$	Barge-in $T3I4R2$	Backchnl. $T3I2R1$	Third-party $T3I7R5$	Hesitation $T5I6R3$	Concurrent $T4I1R1$
FireRedChat	✓	△ EoT	✓	·	✓pVAD	✓EoT	·
MinMo	✓	△ Pred.	✓	✓eval.	·	△ partial	·
Freeze-Omni	✓	·	✓	·	·	·	·
Moshi	✓	△	△ supported	△ training	·	·	△ arch.
LSLM	✓	·	✓	·	·	·	·
OmniFlatten	✓	·	✓	✓	·	·	✓
SyncLLM	✓	·	·	✓	·	·	✓
Mini-Omni2	✓	·	△ keyword	·	·	·	·

Table 3: Ontology-cell coverage matrix for representative full-duplex systems, populated strictly from each paper's own claims (abstract, methodology, evaluation). ✓ the paper explicitly claims or evaluates the capability; △ partial / qualified support (mentioned but not evaluated, partial mechanism, training-data only, command-mode keyword, or architectural plausibility without dedicated evaluation); · the paper does not claim the capability in any section we could verify. Cell shorthand: **Standard** = standard turn; **Latched** = latched zero-gap; **Barge-in** = cooperative barge-in; **Backchnl.** = backchannel during system speech; **Third-party** = third-party / noise rejection; **Hesitation** = hesitation with long silence; **Concurrent** = sustained concurrent speech. *Absence of a claim does not imply that the system fails the cell — only that the paper does not document the capability.* Three patterns are visible: (i) no surveyed paper explicitly claims sustained concurrent speech ($T4 \cdot I1 \cdot R1$), and only OmniFlatten and SyncLLM claim "overlapping speech" as an architecturally supported behavior; (ii) $T3 \cdot I7$ (third-party rejection) is claimed only by FireRedChat via its pVAD module; (iii) $T3 \cdot I2$ (backchannel during system speech) is more widely claimed than we initially assumed — MinMo provides actual back-channeling accuracy in Table 22 of its paper (~70–80%); OmniFlatten and SyncLLM list backchanneling as supported behaviors; Moshi mentions backchannels only in its synthetic-training-data prompts.

plies the LLM's speech-token vocabulary but does not teach duplex dynamics.

- **Type B — Single-stream dialogue / instruction**: single-channel question-answer or instruction data (VoiceAssistant-400K, InstructS2S-200K). Used to teach "answer in speech" but not duplex timing.
- **Type C — Two-stream time-synchronous dialogue**: two-channel recordings (user and assistant on separate channels) with frame-level time alignment. Fisher, Switchboard, CANDOR, Easy Turn, internal call-center corpora. **This is the only data type that directly teaches full-duplex behavior.**

Type C is the binding constraint. Real-world collection requires two human speakers, dual-channel hardware, frame-level timestamping, content annotation, and privacy compliance; per-hour cost is order-of-magnitude higher than single-stream ASR data. The remainder of this appendix focuses on Type C.

D.2 Public Type-C Corpora

Table 4 lists the principal public conversational corpora available for full-duplex training. Four observations: (i) the LDC telephone era (Switchboard (Godfrey et al., 1992), CallHome, Fisher (Cieri et al., 2004)) remains the largest contiguous source of two-channel time-synchronous English audio; (ii) multi-party meeting corpora (ICSI (Janin et al., 2003), AMI (Carletta, 2007)) are abundant in turn-taking phenomena but use multi-channel rather than two-speaker setups and are therefore better suited to diarization-flavored training; (iii) the modern conversational era is best represented by CANDOR (Reece et al., 2023) for English video-call audio and MagicData-RAMC (Yang et al., 2022) for Mandarin mobile-phone audio; (iv) the newest additions are explicitly turn-taking-aware: Easy Turn (Li et al., 2026) provides turn-class labels for end-of-turn classifier training, and HumDial Track 2 (Wang et al., 2026a) provides the first public corpus designed around the full-duplex sub-scenarios (interruption, rejection, third-party speech, pause handling). The combined total of ∼5 k h is one to two orders of magnitude smaller than the industrial Track-B totals discussed in §D.3.

D.3 Two Tracks: Public vs Industrial

Full-duplex training data does not flow through a single shared pipeline. It splits structurally into two tracks with divergent evolution, accessibility, and

Corpus	Hours	Era	Lang.	Format
Two-speaker telephone (LDC era)				
Switchboard	∼260 h	1992	EN	2-channel
CallHome	∼120 h	1990s	EN/ZH/JP/AR/DE/ES	2-channel
Fisher	∼2000 h	2003–04	EN	2-channel
Multi-party meeting				
ICSI Meeting	∼72 h	2003	EN	multi-channel
AMI Meeting	∼100 h	2007	EN	multi-channel + video
Modern conversational (video / mobile)				
MagicData-RAMC	180 h	2022	ZH	2-channel
CANDOR	∼850 h	2023	EN	2-channel (Zoom)
Turn-taking labeled / FD-by-design				
Easy Turn train	1145 h	2025	EN+ZH	turn-class labeled
HumDial Track 2	107 h	2026	EN+ZH	2-channel, 100+ spk

Table 4: Representative public conversational corpora that full-duplex systems have used or could plausibly use for training. Total ∼5 k h dominated by English telephone audio; Chinese-language public coverage is concentrated in MagicData-RAMC and HumDial Track 2 (under 300 h combined). Of these, Fisher is the empirically dominant training source — used by dGSLM (Nguyen et al., 2023) and Moshi (Défossez et al., 2024) — with Easy Turn and HumDial Track 2 the principal post-2024 additions.

coverage bias — a split the rest of the field's data discourse should acknowledge explicitly.

Track A — Public. Driven by academic groups, public funding, and open-source releases; on the order of ∼1 k h per release; mostly commercially usable; biased toward $T1$ / $T2$ / $T5$ cells (clean turns, latched handoffs, hesitation) with $T3$ at natural-distribution frequency and $T4$ essentially absent; updated on a five-to-ten-year cadence (Fisher 2003 → CANDOR 2020 → Easy Turn 2025).

Track B — Industrial proprietary. Continuously fed by deployed products (call centers, smart speakers, in-app voice). Scale is generally not disclosed but is widely believed to be one to two orders of magnitude larger than Track A. Coverage skews toward the deploying product's domain (e.g., call-center conversations are rich in $T3$/$T4$ overlaps but specific to that vertical). Licensing is closed; reproducibility is zero.

The asymmetry of Track B over Track A on raw scale, combined with Track B's product-specific bias, has a load-bearing implication: an open-source full-duplex system trained only on Track-A data will, in the short term, neither match Track-B systems on overall behavior nor be auditable against them on cell-by-cell coverage. The data agenda in §D.6 below is in large part an attempt to make Track A useful in spite of this gap.

D.4 Synthesis Pipelines

Because real Type-C data is scarce and expensive, most modern systems augment with synthetic two-channel dialogues. We catalogue five recipes seen in the recent literature, in order of ontology-cell expressiveness:

1. **Dual-role TTS**: an LLM generates a turn-by-turn script; a multi-speaker TTS synthesizes the two channels. Covers $T1$ only; clean turn-taking with no overlap.
2. **Overlap injection**: dual-role TTS plus randomly injected onset overlaps at turn boundaries. Adds $T2$ (latched) and partial $T3 \cdot I4$ (barge-in). MinMo's heuristic $T \sim \mathcal{N}(0.6, 0.4^2)$ inter-turn distribution is an instance of this.
3. **Backchannel insertion**: detects short backchannel utterances in source dialogue, randomly inserts them into the assistant's speaking span. Adds $T3 \cdot I2$ but requires source dialogue actually containing back-channels — a circular dependency on Type-C data.
4. **Silent-token augmentation**: OmniFlatten's (Zhang et al., 2025b) `silent_speech_token` explicitly marks "the system should be silent here" in the flattened sequence, making $T5$ expressible at the token level without modifying the audio data.

5. **Time-anchor self-supervision**: SyncLLM's (Veluri et al., 2024) `[S0]`/`[S1]` markers at chunk boundaries; teaches the model an explicit sense of within-chunk timing. A streaming-architecture support tool rather than a cell-expanding data augmentation.

Two cells remain unreachable by any of these recipes alone: $T4$ (sustained concurrent speech), because synthesizing two coherent simultaneous monologues is itself unsolved; and $T3 \cdot I7$ (third-party / extraneous voice), because injecting a third speaker requires either real environmental recordings or a non-trivial pipeline.

D.5 Ontology Cell × Training Corpus Coverage

Table 5 maps the seven ontology cells most discussed in this paper against the representative public Type-C *training* corpora; we deliberately exclude the FDB-v1.5 benchmark from this matrix because it is an evaluation resource rather than a training source. Three patterns dominate:

- $T4$ **(sustained concurrent) is the single most under-served cell across all public training data.** Combined with the equally thin evaluation coverage of $T4$ in current benchmarks (Appendix E), this explains why every system in Appendix C.7 fails this cell.
- $T3 \cdot I7$ **(third-party voice) is newly addressed by HumDial Track 2.** Before HumDial, this cell was absent from public training data; HumDial Track 2 (Wang et al., 2026a) explicitly includes a Rejection scenario covering "Third-party Speech" and "Speech Directed at Others" as sub-tasks, making this the first public corpus on which $T3 \cdot I7$ can be directly trained.
- **The CANDOR/MagicData/HumDial trio addresses the modern-acoustic / multilingual gap that Fisher and Switchboard leave.** A 2026 open-source FD training mix that combines Fisher (telephone, scale) with CANDOR (video-call acoustics), MagicData-RAMC (Mandarin), Easy Turn (turn-class labels), and HumDial Track 2 (rejection sub-scenarios) covers six of the seven ontology cells; the remaining gap is $T4$.

D.6 Data Research Directions

The audit above motivates six concrete data-side research directions. We list them in approximate decreasing tractability for the open-source community:

1. **Third-party / background voice synthesis (short-term)**: combine an existing single-stream voice library, multi-speaker TTS, and room-impulse-response convolution to produce $T3 \cdot I7$ training data. A pure engineering task; estimated person-month-scale to assemble a multi-thousand-hour synthetic corpus.
2. **Ontology-driven synthesis pipeline (short-term)**: extend the five-recipe survey above into a single open tool with per-cell injection probabilities (e.g., $p_{T_2}, p_{T_3 \cdot I_4}, p_{T_3 \cdot I_2}, p_{T_3 \cdot I_7}, p_{T_4}$), so that any LLM-generated script can be expanded into two-channel dialogue with a target cell distribution rather than the natural distribution.
3. **Cross-dataset alignment and combination (short-term)**: align Easy Turn ($T1$/$T5$), FDB-v1.5 ($T3$), and Fisher ($T1$/$T2$ + natural $T3$) under the ontology framework into a single "ontology-typed" open corpus.
4. T_4 **targeted real-data collection (medium-term)**: design real-data collection protocols for sustained concurrent speech — conference calls, group discussion, debate, family-room recordings. Pure synthesis cannot reach this cell at current state of the art; real collection is required.
5. **Agent-vs-agent self-play data generation (medium-to-long-term)**: let two full-duplex models converse with each other under ontology-typed sampling and filtering. Risks amplifying biases of current models, so should be combined with the synthesis pipeline of direction 2 as an ontology constraint.
6. **Training objectives that compensate for data scarcity (short-to-medium-term)**: recent work argues that better training objectives can substitute for missing data — reinforcement learning with conversational rewards, or auxiliary turn-aware objectives that imprint turn structure on ordinary dialogue data. This is the only direction that does not require new

Cell	Fisher	CANDOR	MagicData-RAMC	Easy Turn	HumDial T2
$T1 \cdot I1 \cdot R6$ Standard	✓	✓	✓	✓	✓
$T2 \cdot I1 \cdot R6$ Latched	✓	✓	✓	△	△
$T3 \cdot I4 \cdot R2$ Barge-in	✓	✓	✓	·	✓
$T3 \cdot I2 \cdot R1$ Backchannel	✓	✓	✓	✓	✓
$T3 \cdot I7 \cdot R5$ Third-party	·	△	·	·	✓
$T5 \cdot I6 \cdot R3$ Hesitation	✓	✓	✓	✓	✓
$T4 \cdot I1 \cdot R1$ Concurrent	·	△	·	·	·

Table 5: Ontology-cell coverage of representative public Type-C training corpora. ✓ well represented (Fisher, CANDOR, and MagicData-RAMC contribute $T3$ overlap cells at the natural conversational rate; Easy Turn contributes $T1/T3 \cdot I2/T5$ via designed turn classes; HumDial Track 2 (Wang et al., 2026a) contributes $T3 \cdot I4$ and $T3 \cdot I2/I7$ via designed interruption/rejection sub-scenarios); △ partial (CANDOR has occasional Zoom-incident third-party speech and brief concurrent stretches; Easy Turn and HumDial cover $T2$ only as a degenerate end-of-turn case); · not represented for structural reasons (telephone and single-party-recording corpora structurally exclude third-party voice; no public corpus contains sustained concurrent monologue). HumDial Track 2 is the first public corpus to designate $T3 \cdot I7$ third-party speech as an explicit training target. The FDB-v1.5 benchmark (Lin et al., 2026b) is an evaluation resource and is treated separately in Appendix E.

data collection and is therefore the highest-leverage short-term direction for open-source teams.

D.7 Scope of the Thesis

The data-bottleneck thesis of §6 is a strong but bounded claim, and four boundaries are worth stating explicitly so that it is not over-read as "architecture does not matter".

The thesis applies only at L0–L2. The hierarchy of §3 has mature representatives at L0, L1, and L2. L3 representation-level full-duplex has no representative yet, and reaching it will require architectural innovations (continuous-latent autoregressive streaming, JEPA-style dual-stream prediction, world-model-conditioned dialogue) that are not data problems but modelling-paradigm problems.

Architecture and data are coupled, not independent. The thesis does not claim that architecture choice has no effect on data efficiency. Multi-stream designs may convert $T3 \cdot I4$ data into capability more efficiently than flattened designs; channel-fusion designs may be more sample-efficient in low-data regimes. The thesis claims that the *ceiling* of behavior is set by data; architecture sets the gain on the path from data to ceiling.

Engineering cost may favor the architecture path. In an industrial setting, the cost of acquiring sufficient Type-C data may be comparable to or exceed the cost of architecture iteration. The thesis is a statement about research priorities, not about every team's optimal engineering choice.

Evaluation paradigms may shift the picture. Current benchmarks (FDB family, MTR-Duplex) measure largely static turn-taking behaviors. If the field moves to multi-turn, agent-versus-examiner, adversarial evaluation, then systems with stronger reasoning traces (e.g., a Thinker–Talker decomposition) may demonstrate capability gaps that no amount of additional Type-C data closes. The thesis is best read as conditional on the current evaluation regime.

Within these bounds, the bottleneck thesis explains, with no further machinery, the cross-system performance variance reported in Appendix C.7 and the open-source-versus-industrial gap that §D.3 documents.

E Evaluation: Extended Benchmark Survey and Realization-Gap Analysis

This appendix expands the evaluation analysis stated in §7. The body section establishes (a) why conventional metrics fail, (b) the two-series FDB / FD-Bench disambiguation, and (c) the realization-gap concept; this appendix supplies the per-benchmark notes, the cell-coverage matrix, an extended discussion of the realization gap, and the three structural evaluation gaps in detail.

E.1 Survey of Full-Duplex Benchmarks

Naming disambiguation. The literature contains two independent, simultaneously active benchmark series with confusingly similar names. We name

them explicitly here because cross-referencing between them is non-trivial and uncited mentions of "FD-Bench" in recent papers are sometimes ambiguous.

- **Series A — Full-Duplex-Bench (FDB).** A versioned line by Lin et al.: v1 (Lin et al., 2025b) → v1.5 (Lin et al., 2026b) → v2 (Lin et al., 2025a) → v3 (Lin et al., 2026a), with v1.5 establishing the overlap-handling protocol most other recent work compares against. Focus: behavior-level evaluation, per-cell metrics.
- **Series B — FD-Bench.** An independent benchmarking pipeline by Peng et al. (Peng et al., 2025) (data generator, test-case construction, scoring scripts) rather than a single fixed test set. Focus: system-level pipeline evaluation including throughput, latency, and robustness.

The two series are not interchangeable. They measure different things along different axes and their numbers are not directly comparable. Future work that needs to cite "FD-Bench" generically should explicitly name the team or the version.

Benchmark-by-benchmark notes. We summarize the resources that we treat as load-bearing in this paper:

- **Talking Turns** (Arora et al., 2025). Trains a supervised judge model on Switchboard human-human dialogue and uses it to score whether a full-duplex system's per-frame behavior (speak / pause / backchannel) matches the human distribution. Strength: anchored in real human turn-taking statistics. Limitation: covers $T1$, $T2$, and the $T3 \cdot I4$ slice of the ontology; silent on $T3 \cdot I2$ (backchannel-as-listener) and $T3 \cdot I7$ (third-party voice).
- **FDB v1** (Lin et al., 2025b) (Series A). Four axes — Pause, Backchannel, Turn-taking, Interruption — with automated metrics for each. Coverage: $T5 \cdot I6 \cdot R3$, $T3 \cdot I2 \cdot R1$, $T1 \cdot I1 \cdot R6$, $T3 \cdot I4 \cdot R2$. Single-turn, English-dominant, static test set.
- **FDB v1.5** (Lin et al., 2026b). The key v1.5 contribution is the decomposition of overlap into four distinct scenarios — user interruption, user backchannel, talking-to-others, background speech — which align nearly one-to-one with the $T3$ row of the T × I ontology. The *stop-latency* metric (how quickly the system yields after a user onset) is, to our knowledge, the closest existing public metric to a substantive-FD test. A reported finding of v1.5 is that current systems split into two dominant strategies — response-first (fast but easily confused by backchannels) and floor-holding (stable but slow on legitimate barge-in) — which is itself a useful framing for the ontology audit of §3.2.
- **FDB v2** (Lin et al., 2025a) (Series A). Replaces the static test set with a multi-turn automated examiner that drives the system through four task families (daily / correction / entity-tracking / safety) at two paces (fast / slow). This is a paradigmatic shift from passive test set to active conversational partner. Reported finding: every system in scope fails on the combination of concurrent speech ($T4$) plus mid-turn correction ($T3 \cdot I3$) plus multi-turn entity tracking. The failure pattern coincides with the data-missing cells of Appendix D.5, providing the most direct independent evidence we have for the data-bottleneck thesis.
- **FDB v3** (Lin et al., 2026a) (Series A). Extends FDB to tool-using voice agents: it scores how a system handles tool calls under disfluent user behavior (hesitation, mid-call interruption, recovery). Treats the FD system as a component of a larger agent stack rather than as a dialogue partner in isolation; appropriate for voice-agent evaluation, less so for core FD architecture comparison.
- **MTR-Duplex** (Zhang et al., 2025a). Multi-round full-duplex evaluation focused on conversational content quality — dialogue features, dialogue quality, instruction following, safety. Complementary to FDB v2: FDB v2 tests *when* the system speaks, MTR-Duplex tests *what* it says.
- **FD-Bench** (Peng et al., 2025) (Series B). Pipeline-style: provides a configurable generator for test scenarios rather than a fixed test set. Distinct in form and intent from Series A; appropriate when evaluating engineering robustness rather than behavioral coverage.
- **Easy Turn test set** (Li et al., 2026). The test

slice of the Easy Turn corpus measures four-class turn classification: complete / incomplete / backchannel / wait. This is the first public open-source resource that lets independent systems report a comparable accuracy number on $T1$/$T5$ turn classes, with a published SOTA baseline.

- **HumDial Challenge 2026** (Wang et al., 2026a). Announced as releasing (i) a dual-channel real human-dialogue corpus, (ii) a public leaderboard, and (iii) a standardized evaluation protocol. The signature open-evaluation event of 2026 for full-duplex.

Survey companions. Ji et al. (2024) survey end-to-end spoken dialogue models more broadly; their evaluation chapter overlaps in scope with this appendix but predates the FDB v1.5 / v2 wave. Chen and Yu (2025) review turn-taking specifically and provide useful pre-FDB context.

E.2 Ontology-Cell Coverage of Benchmarks

Table 6 maps the principal benchmarks against the same seven ontology cells used in Appendix D.5. Three patterns are visible:

- $T1$ **Standard and** $T3 \cdot I4$ **Barge-in are over-served** relative to other cells: every benchmark in the table covers both, often with multiple metrics each. Comparing systems on these cells alone is unlikely to yield new information.
- $T4$ **Concurrent remains under-served on the evaluation side too** — only FDB v2's examiner partially probes it. The training-data scarcity for $T4$ documented in Appendix D.5 is therefore mirrored on the evaluation side: even teams that solve the data problem would currently lack a clean way to demonstrate the solution.
- **FDB v1.5 is the single most useful benchmark for the** $T3$ **row** by virtue of its overlap-scenario decomposition. Its *stop-latency* metric is the closest existing operationalization of the substantive-FD criterion of §5, and is the metric the rest of this appendix recommends as the empirical anchor.

E.3 The Realization Gap

The hierarchy of §3 describes a system's *architectural capacity* — what cells the design admits in principle. The benchmarks of §E.1 measure the system's *demonstrated behavior* — what cells the trained system actually serves on the test set. The discrepancy between the two is what we call the *realization gap*, and it is the conceptual contribution this appendix recommends taking forward.

The realization gap is empirically non-trivial. Two L2 systems with the same architectural capacity can have very different demonstrated behavior on the same cell — Mini-Omni2 and OmniFlatten both belong to the L2 flattened family, but the former demonstrates neither backchannel nor concurrent capability on FDB while the latter claims both. The cause is downstream of architecture and upstream of evaluation: it is the cell-coverage distribution of training data, the subject of Appendix D. The realization gap thus operationalizes the link between Appendix C.7 and Appendix D: *architecture sets the ceiling, training data realizes some fraction of it, and evaluation reports that fraction*.

A useful evaluation report for a full-duplex system therefore has three components, not one: (i) the architectural L-layer of the system (from Appendix C.7); (ii) the cell-coverage profile of its training data (from Appendix D); and (iii) the demonstrated per-cell behavior on the benchmarks of §E.1. A single aggregate score collapses these three into a number that obscures the diagnostic question — "is this system failing because of architecture, data, or evaluation construct?" — that an evaluation should answer.

This is the appendix's conceptual recommendation. We do not propose new metrics or a new benchmark; we propose that existing benchmarks be reported under this three-component frame.

E.4 Structural Gaps in the Current Regime

We close with three structural gaps that the current evaluation regime exhibits. None of them is fully closable in the short term, but each suggests a concrete direction for the next round of benchmarks.

Static test sets versus genuinely adaptive interaction. All public FDB versions before v2, all of Easy Turn, and all of Series B's pipeline scenarios are pre-recorded: the system's response is logged but does not change what the test set asks next. Real conversation is recursive in exactly the opposite direction — the human partner adapts to what the system just did. FDB v2's automated examiner is a partial fix but the examiner itself follows a scripted task family rather than mounting a

Cell	Talking Turns	FDB v1	FDB v1.5	FDB v2	Easy Turn
$T1 \cdot I1 \cdot R6$ Standard	✓	✓	✓	✓	✓
$T2 \cdot I1 \cdot R6$ Latched	✓	△	△	✓	△
$T3 \cdot I4 \cdot R2$ Barge-in	✓	✓	✓	✓	·
$T3 \cdot I2 \cdot R1$ Backchannel	·	✓	✓	△	✓
$T3 \cdot I7 \cdot R5$ Third-party	·	·	✓	△	·
$T5 \cdot I6 \cdot R3$ Hesitation	△	✓	△	✓	✓
$T4 \cdot I1 \cdot R1$ Concurrent	·	·	·	△	·

Table 6: Ontology-cell coverage of the principal evaluation resources. ✓ the cell is directly tested by an explicit metric or sub-task in the benchmark; △ the cell is partially tested, either as a subset of a broader category or through a metric that does not fully isolate it; · the cell is not tested. FDB v1.5's row reflects its overlap-into-four-scenarios decomposition, including *talking-to-others* / *background speech* which together cover $T3 \cdot I7$. FDB v2 partially reaches $T4$ via its examiner's concurrent-speech probes. No public benchmark currently tests $T4$ as a primary metric; this is the single largest hole in the evaluation landscape.

genuinely adaptive adversary.

Language and acoustic distribution. Roughly 95 % of public benchmark audio is English, and the acoustic distribution is dominated by telephone and Zoom-style recordings. Industrial systems serve mobile-microphone audio in noisy environments in Chinese and other languages, but there is no standardized way to report results in those conditions. The result is that the open-source / industrial gap discussed in Appendix D.3 is reinforced rather than measured by the current benchmarks.

Absence of explicit substantive-versus-apparent probes. No public benchmark currently includes a causal-influence test of the kind described in §5 — a controlled counterfactual experiment that asks whether the system's mid-turn behavior is genuinely sensitive to the user audio, or whether it is following a fixed schedule. Chunk-alternating designs (e.g., the SyncLLM family) can score well on stop-latency without passing such a test, and the field would benefit from a benchmark that explicitly distinguishes the two.

Taken together, the three gaps suggest that the next round of evaluation work should aim for an adaptive multi-turn protocol with explicit causal-influence probes and multilingual / multi-acoustic test conditions. None of these directions requires inventing new metrics; they require recombining the protocols already represented in the benchmarks of §E.1 with the data agenda of §D.6.

F Decision State Machine: Formal Definitions and Per-System Traces

This appendix gives the explicit transition table behind the state machine introduced in §5, and works through six per-system traces that illustrate how the same five-state, eleven-transition graph redescribes systems whose actual implementations live at very different layers of the stack. Nothing here departs from the main-text definitions; the goal is to make the FSM concrete enough to serve as an ablation checklist in future work.

F.1 Notation

A state is one of $\mathcal{S} = \{\text{IDLE}, \text{LISTEN}, \text{SPEAK}, \text{WAIT}, \text{DUAL}\}$. A transition is written $s \xrightarrow{(T_i, I_j)/R_k} s'$ to mean: when the system is in state s and the current input is of temporal class T_i and intent class I_j, the system takes the response of class R_k and moves to state s'. Triggers and actions are drawn from the $\text{T} \times \text{I} \times \text{R}$ ontology of §4. A transition with no R component is a perceptual transition (the system updates its state based on input without producing output). Self-loops are written $s \xrightarrow{(T_i, I_j)/R_k} s$.

F.2 The Eleven Transitions

Table 7 enumerates the eleven transitions. The transitions split into three families introduced in §5.1: *onset transitions* (τ_1, τ_2) bring the system out of IDLE; *turn-handoff transitions* (τ_3–τ_7) move between single-speaker states; *overlap transitions* (τ_8–τ_{11}) describe what happens when both speakers are active. The overlap family is the duplex-specific one: a half-duplex system implements τ_8 as an unconditional fast handoff to LISTEN and does not distinguish τ_9 from τ_{10} or τ_{11}, which is the formal sense in which it "cannot tell backchannel from interruption."

#	From	To	Trigger (T, I)	Action R	Notes
τ_1	IDLE	LISTEN	(T_1, I_1) user onset	—	Front-end VAD or speaker-conditioned VAD detects user speech.
τ_2	IDLE	SPEAK	(T_1, I_5) proactive timer	R_6 respond	System initiates a turn after a silence threshold.
τ_3	LISTEN	SPEAK	(T_1, I_1) EoT confirmed	R_6 respond	End-of-turn predictor fires; standard handoff.
τ_4	LISTEN	WAIT	(T_5, I_6) user hesitation	R_3 wait	Semantic EoT says the user has not finished.
τ_5	WAIT	SPEAK	(T_1, I_1) EoT re-confirmed	R_6 respond	User completes their utterance.
τ_6	WAIT	LISTEN	(T_3, I_1) user resumes	—	User keeps talking after the pause.
τ_7	SPEAK	IDLE	(T_1, I_1) system EoT	—	System finishes its turn cleanly.
τ_8	SPEAK	DUAL	(T_3, I_*) overlap detected	—	User audio onset while system is speaking; I disambiguated in DUAL.
τ_9	DUAL	LISTEN	(T_3, I_4) floor-claim	R_2 yield	System stops speaking and listens.
τ_{10}	DUAL	SPEAK	(T_3, I_2) backchannel	R_1 continue	System keeps speaking; brief listener-acknowledgement absorbed.
τ_{11}	DUAL	SPEAK	(T_3, I_7) third-party / noise	R_5 ignore	System keeps speaking; filters out non-addressee audio.

Table 7: The eleven transitions of the full-duplex decision state machine. Triggers are written as (T, I) pairs from the ontology of §4; actions are R classes. Transitions τ_1, τ_6, τ_7, and τ_8 are perceptual — they update state from input without producing an output action. Sustained T_4 concurrent speech is the DUAL self-loop with no transition out, omitted from the row count because every system in Appendix C.7 fails to enter it stably.

F.3 Cell-to-Transition Mapping

The state machine and the T × I × R ontology compose: each non-trivial ontology cell maps to exactly one transition, and each transition realizes a small set of cells.

- $T_1 \cdot I_1 \cdot R_6$ *Standard* = τ_3.
- $T_2 \cdot I_1 \cdot R_6$ *Latched* = τ_3 executed with zero or negative latency between user EoT and system speech onset.
- $T_3 \cdot I_4 \cdot R_2$ *Cooperative barge-in* = τ_8 followed by τ_9.
- $T_3 \cdot I_2 \cdot R_1$ *Backchannel* = τ_8 followed by τ_{10}.
- $T_3 \cdot I_7 \cdot R_5$ *Third-party* = τ_8 followed by τ_{11}.
- $T_5 \cdot I_6 \cdot R_3$ *Hesitation* = τ_4 followed by τ_5 (or τ_6).
- $T_4 \cdot I_1 \cdot R_1$ *Concurrent* = τ_8 followed by a sustained DUAL self-loop.

A system that lacks one of the eleven transitions also lacks every cell that depends on it. This is the precise sense in which the state machine functions as an ablation inventory: each transition is an independently testable behavior, and a benchmark score that fails on one cell can be localized to the missing or incorrectly implemented transition.

F.4 Per-System Traces

We work through six traces against the table above to illustrate how systems at different L-layers realize the same FSM.

Trace 1 — Standard turn-taking (every system). IDLE $\xrightarrow{\tau_1}$ LISTEN $\xrightarrow{\tau_3}$ SPEAK $\xrightarrow{\tau_7}$ IDLE. This is the half-duplex baseline trace. Every system in Appendix C.7 traverses it; differences live in the timing of τ_3 and τ_7, not in the trace itself.

Trace 2 — Hesitation handling (MinMo). LISTEN $\xrightarrow{\tau_4}$ WAIT $\xrightarrow{\tau_5}$ SPEAK. MinMo (Chen et al., 2025b) fires τ_4 when its semantic EoT predictor recognizes that the user has not yet finished (e.g., trailing-off intonation or an incomplete syntactic frame). FireRedChat (Chen et al., 2025a) executes the same trace at L0 via its EoT classifier; the difference is implementation layer, not the trace shape. An L2 system without an explicit semantic-

EoT module typically collapses τ_4 into a delayed τ_3 — which is observable as a longer mean response latency on T_5-heavy test cases but not as a separate transition.

Trace 3 — Cooperative barge-in (Moshi). SPEAK $\xrightarrow{\tau_8}$ DUAL $\xrightarrow{\tau_9}$ LISTEN. Moshi (Défossez et al., 2024) executes τ_8 implicitly because the user-audio stream is always being predicted in parallel with assistant audio; the system is therefore in DUAL continuously and τ_9 amounts to suppressing the assistant audio stream while continuing to attend to the user stream. The transition latency is dominated by the codec frame rate (12.5 Hz → 80 ms minimum) plus a small recognition delay.

Trace 4 — Backchannel-during-speech (OmniFlatten). SPEAK $\xrightarrow{\tau_8}$ DUAL $\xrightarrow{\tau_{10}}$ SPEAK. OmniFlatten (Zhang et al., 2025b) uses an explicit `silent_speech_token` on the assistant stream to indicate "stay silent here" and continues the assistant utterance after the user's brief acknowledgement. The trace is short — typically a few tokens at the flattened sequence's granularity. A system that lacks τ_{10} collapses this trace into $\tau_8 + \tau_9$ (backchannel mistaken for floor-claim, system yields incorrectly).

Trace 5 — Third-party voice (FireRedChat). SPEAK $\xrightarrow{\tau_8}$ DUAL $\xrightarrow{\tau_{11}}$ SPEAK. FireRedChat's (Chen et al., 2025a) speaker-conditioned VAD (pVAD) detects that the overlapping audio is not from the addressed user; the dialogue manager routes this to τ_{11} rather than τ_9. The decision is made at the module layer rather than the token layer, but the FSM trace is identical to Moshi's structurally; the difference is which component is responsible for the I_7 classification.

Trace 6 — Sustained concurrent (no system). SPEAK $\xrightarrow{\tau_8}$ DUAL $\xrightarrow{\text{(self-loop)}}$ DUAL $\cdots$. The state machine admits a DUAL self-loop on $T_4 \cdot I_1 \cdot R_1$, but no system in Appendix C.7 stably sustains it. The architectural cause varies (chunk-alternating designs cannot represent it; multi-stream designs can represent it but fail to learn it), but in every case the trace collapses to a $\tau_8 + \tau_9$ or to acoustic degradation within a few frames. Sustained T_4 is the open transition of the state machine.

F.5 What the State Machine Is Not

We close with three explicit non-claims, so that the formalism is not over-read.

First, the state machine is not a controller. It describes the externally observable behavior of a full-duplex system; nothing in it specifies how a particular system computes its next transition. An L0 system computes it from module outputs, an L1 system from a sidecar predictor, an L2 system from token-level decoding — and the FSM is silent on which.

Second, the transitions are not equally hard. τ_3 and τ_7 (clean turn-taking) are essentially solved by every system in the audit; τ_{10}, τ_{11}, and the sustained DUAL self-loop on T_4 are not. A flat per-transition checklist would obscure this; the cell-coverage matrix of Appendix C.7 is the appropriate companion view.

Third, the eleven transitions are not exhaustive over all conceivable dialogue events — they cover the cells the ontology of §4 identifies as currently load-bearing. A future ontology revision that introduced, for instance, an I_8 (sustained meta-conversational signal such as long-form laughter) would add a corresponding transition; the framework is open in this direction.