REVISION-BOUND GROUNDING FOR STATEFUL TOOL USE IN OPEN-SOURCE FULL-DUPLEX VOICE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Full-duplex voice agents can listen, speak, reason, and act concurrently, but this creates a state-consistency problem: user intent may change while earlier reasoning or tool execution remains in flight, allowing stale evidence or actions to affect the interaction. We introduce *revision-bound grounding*, a causal interface that binds asynchronous evidence, action proposals, authorizations, and tool outcomes to their originating dialogue revision and action trace, admitting evidence and pending actions only while they remain authoritative, while preserving verified external effects in observed world state. We instantiate this interface in MOSHI-AGENT, which decouples continuous speech from asynchronous reasoning and tool execution while maintaining explicit dialogue, action, and observed world state. We also construct a tool-grounded duplex corpus of 599K conversations totaling 12.1K hours and train MOSHIAGENT-SPEECH to incorporate delayed evidence during ongoing interaction. Across six audio-agent benchmark families and TOOLRACEBENCH, a scenario-based benchmark for stateful tool use, revision-bound execution with fixed MOSHIAGENT-SPEECH improves Full-Duplex-Bench v3 argument accuracy from 65.0 to 77.6 and strict Pass@1 from 56.0 to 67.0, while increasing TOOLRACEBENCH Pass@1 from 34.2 to 83.2. The complete system achieves 32.14 Macro Pass@1 on τ-Voice, surpassing GPT-Realtime-2.1 mini.

1 INTRODUCTION

Full-duplex voice agents move spoken interaction from a turn-serialized exchange to a continuous setting in which listening, speaking, reasoning, retrieval, and tool execution may proceed on overlapping timelines. This capability is essential for natural pauses, backchannels, overlap, and interruption, but it also changes the reliability problem for stateful agents. A reliable agent must not only understand the latest user utterance or select an appropriate external tool. It must also maintain causal consistency among the evolving dialogue, the actions authorized by that dialogue, and the world state later observed after those actions.

Recent full-duplex systems increasingly decouple latency-critical speech generation from asynchronous retrieval, reasoning, and tool use (Défossez et al., 2024; Roy et al., 2026; Chien et al., 2026; Huang et al., 2026; NVIDIA, 2026). In parallel, stateful agent runtimes provide mechanisms for authorization, provenance, and transactional control over external actions (Chen et al., 2026b; Santos-Grueiro, 2026a;b; Xu et al., 2026). These two lines of work address complementary parts of the problem: full-duplex architectures determine when asynchronous computation can occur, while agent runtimes constrain how external effects are authorized and recorded. However, neither by itself specifies which evidence, proposal, invocation, or result should remain authoritative when the dialogue state that produced it has already been superseded.

Despite this progress, reliable full-duplex agents still face two coupled challenges at the boundary between continuous interaction and asynchronous execution. First, user intent may change while earlier reasoning or tool actions remain in flight. For example, a user may ask to set an alarm for 9 a.m. and then interrupt with "actually, make it 10" before the original request completes. Even if the speech model recognizes the revision and the external action executes correctly, the system may still commit or report the obsolete 9 a.m. change if stale work remains authoritative. Second, preventing such failures requires stronger grounding between the speech and execution layers than transcripts or final tool results alone can provide. The speaker must track whether an action is proposed, authorized,

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

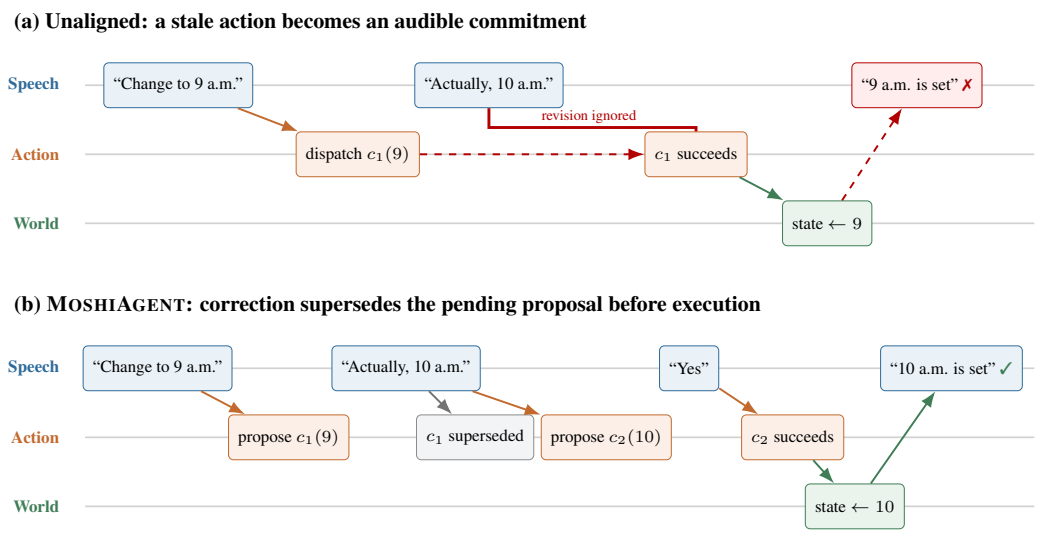

Figure 1: **Revision-bound grounding aligns concurrent speech, action, and world state.** (a) In a naive full-duplex composition, a user revision can arrive after a stale action has been dispatched, allowing the obsolete action to succeed and be reported. (b) MOSHIAGENT binds pending actions to the active dialogue revision, supersedes the stale proposal before execution, and reports completion only after the confirmed action succeeds and its effect is observed in world state.

invoked, observed, or superseded, while the executor must know which dialogue revision supports the action and whether that intent has changed. Without this shared causal state, speech and execution can drift apart even when each component behaves correctly in isolation.

We address these challenges with MOSHIAGENT, a full-duplex voice agent built around the co-design of continuous speech interaction and revision-bound execution. MOSHIAGENT couples a continuously interacting Streaming Speaker with asynchronous reasoning and tool execution through a shared revision-bound grounding interface that determines which evidence and actions remain authoritative as user intent evolves. The interface binds returned evidence, action proposals, authorizations, and tool outcomes to the dialogue revision and exact invocation from which they originate. This allows the speaker to incorporate admitted evidence without blocking ongoing interaction, while preventing superseded reasoning or actions from becoming authoritative. In this way, MOSHIAGENT supports concurrent speaking, reasoning, and acting while maintaining causal consistency between evolving user intent, external actions, and observed world state.

To realize this co-design on the speech side, we construct a tool-grounded duplex corpus of 599K conversations totaling 12.1K hours and train MOSHIAGENT-SPEECH to recognize when external support is needed and to incorporate delayed evidence during continuous interaction. We evaluate the resulting speech–execution co-design across six audio-agent benchmark families and TOOLRACEBENCH, a scenario-based benchmark for stateful tool use. With MOSHIAGENT-SPEECH fixed, replacing a direct tool loop with revision-bound execution improves Full-Duplex-Bench v3 argument accuracy from 65.0 to 77.6 and strict Pass@1 from 56.0 to 67.0, raises τ-Voice Macro Pass@1 from 18.6 to 32.1, and increases TOOLRACEBENCH Pass@1 from 34.2 to 83.2. The same speech and execution stack operates with four reasoning backends without backend-specific speech retraining. Our contributions are summarized below:

- We introduce MOSHIAGENT, a full-duplex voice-agent framework that co-designs continuous speech interaction with revision-bound execution through a shared grounding interface. The framework enables asynchronous reasoning and tool use while maintaining causal consistency between evolving user intent, external actions, and observed world state.
- We construct open resources for training and evaluating grounded full-duplex agents, including a tool-grounded duplex training corpus of 599K conversations totaling 12.1K hours and TOOLRACEBENCH, a benchmark targeting stateful tool use.

- We conduct extensive evaluations across seven audio-agent benchmark families, four reasoning backends, controlled speech–execution comparisons, and component ablations. The results demonstrate that the proposed speech–execution co-design substantially improves stateful task completion and tool-use reliability, including improving Full-Duplex-Bench v3 strict Pass@1 from 56.0 to 67.0, τ-Voice Pass@1 from 18.6 to 32.1, and TOOLRACEBENCH Pass@1 from 34.2 to 83.2.

2 RELATED WORK

2.1 FULL-DUPLEX SPEECH AND ASYNCHRONOUS REASONING

Recent speech language models increasingly support end-to-end, streaming, and full-duplex interaction, including overlap, backchanneling, and interruption (Xie & Wu, 2024; Zeng et al., 2024; Xu et al., 2025a; Wang et al., 2025; Zhang et al., 2025; Défossez et al., 2024; Roy et al., 2026). Corresponding benchmarks evaluate these capabilities under increasingly interactive spoken settings (Lin et al., 2025; 2026). A complementary line moves slower semantic computation outside the latency-critical interaction path through background reasoning, asynchronous retrieval, and pluggable deliberation (Christakopoulou et al., 2024; Srinivas et al., 2026; Zou et al., 2026; Chien et al., 2026; Huang et al., 2026; Liang et al., 2026). These works primarily address *when* slower computation should occur and how its output can re-enter an ongoing interaction. We address the complementary question of *which* state remains authoritative when user speech is revised while reasoning, retrieval, or tool execution remains in flight. MOSHIAGENT introduces revision-bound grounding to admit asynchronous evidence according to the dialogue revision and action trace from which it originates.

2.2 TOOL-USING AGENTS AND STATEFUL EXECUTION

Tool-using agents have progressed from interleaved reasoning and action toward large-scale function calling and multi-turn evaluation in environments whose state evolves across tool interactions (Yao et al., 2023; Schick et al., 2023; Patil et al., 2024; Qin et al., 2024; Liu et al., 2025; Patil et al., 2025; Lu et al., 2025; Yao et al., 2025; Barres et al., 2026). Spoken agents and benchmarks increasingly extend these capabilities to interactive speech, tool use, and verifiable task completion (Maben et al., 2025; Chen et al., 2025a; Liang et al., 2026; NVIDIA, 2026; Zhang et al., 2026; Chen et al., 2026a; Jain et al., 2026; Pahwa et al., 2026; Lin et al., 2026; Ray et al., 2026). Reliable execution additionally requires authority, provenance, correlation, and state-consistent commit semantics. Recent agents apply these principles through contextual authorization, argument provenance, state validation, and transactional or commit-time controls (Shi et al., 2026; Qin et al., 2026; Fan et al., 2026; Liu et al., 2026; Chen et al., 2026b; Santos-Grueiro, 2026a). These approaches generally assume an explicit reasoner–tool boundary after an action has been proposed. Full-duplex speech introduces an additional concurrency problem because the user evidence supporting that action may change before execution completes. MOSHIAGENT bridges these settings by binding conversational revisions, proposals, confirmations, exact invocations, and accepted outcomes through a shared causal execution boundary.

3 REVISION-BOUND GROUNDING FOR FULL-DUPLEX AGENTS

3.1 PROBLEM SETTING AND AUTHORITATIVE STATE

A full-duplex session is not a sequence of completed turns. User speech, agent speech, reasoning, retrieval, and tool execution may proceed concurrently. Consequently, user intent may change while work initiated from earlier dialogue remains in flight. An asynchronous result can therefore be correct for the state that produced it but no longer authoritative for the current interaction. This means that completing an asynchronous operation is not, by itself, sufficient for its result to affect the current session. We therefore distinguish event arrival from event admission. Arrival makes an asynchronous event available to the runtime, whereas admission determines whether it may affect authoritative state or subsequent user-facing behavior.

At event index t, we represent authoritative session state as

$$S_t = (D_t, A_t, W_t), \tag{1}$$

where D_t is dialogue state, A_t is action state, and W_t is observed world state. These components are coupled but need not update in lockstep. As shown in Figure 2, the Streaming Speaker remains on the

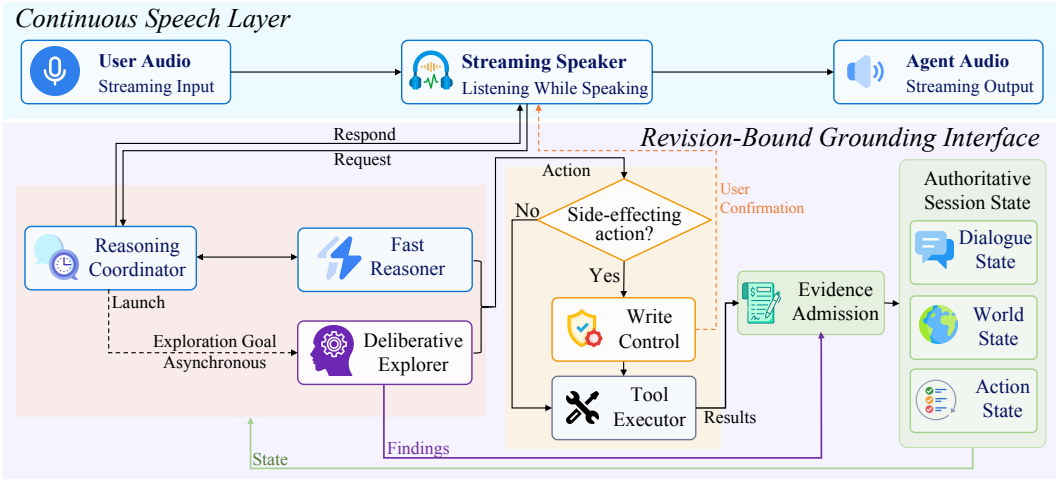

Figure 2: **Architecture of MOSHIAGENT.** A continuous Streaming Speaker is decoupled from asynchronous reasoning and tool execution. The grounding interface determines which returned evidence may enter authoritative session state, while Write Control gates side-effecting actions and binds user authorization to the exact proposal being confirmed.

latency-critical interaction path, while reasoning and tool execution proceed asynchronously behind a shared revision-bound grounding interface. The interface mediates returned evidence and proposed tool actions before they may update authoritative session state or trigger external side effects.

3.2 REVISIONS AND SEMANTIC SUPERSESSION

Revision-bound grounding associates asynchronous work with the user evidence from which it was derived and evaluates its continued authority against subsequent user evidence.

User-Evidence Revisions. The dialogue state D_t maintains a monotonically increasing user-evidence revision r over its authoritative user-side transcript. Every asynchronous operation records its originating revision r_o, which identifies the user evidence available when that operation was initiated. The authority of an operation is evaluated against user evidence that arrives after r_o. An operation remains authoritative when the subsequent evidence is compatible with the intent under which it was initiated. It becomes superseded when the subsequent evidence withdraws that intent or changes an action-relevant target, argument, or scope. Compatible evidence may therefore extend the dialogue without invalidating work derived from an earlier revision.

Proposal-Relative Supersession. For a side-effecting action that requires user authorization, the runtime represents the pending proposal as

$$p = (\mathrm{pid}, C_p, r_p), \tag{2}$$

where pid is the proposal identity, C_p is the exact canonical action specification bound to the user-facing proposal, and r_p is the user-evidence revision at which that proposal was presented. Let ΔU_p denote the authoritative user speech observed after revision r_p. The runtime evaluates the relation between this evidence and the pending proposal as

$$g(\Delta U_p, C_p) \in \{\texttt{accept}, \texttt{reject}, \texttt{modify}, \texttt{unclear}\}. \tag{3}$$

`accept` denotes explicit authorization of the complete action represented by C_p and authorizes only that proposal. `reject` denotes refusal or withdrawal and revokes the pending proposal. `modify` denotes a change to an action-relevant target, argument, or scope of C_p, superseding the pending proposal and its derived candidates and triggering replanning from the updated user evidence. `unclear` covers evidence that does not resolve the proposal, including questions, unrelated speech, incomplete audio, and unresolved mixed intent, and leaves the proposal pending.

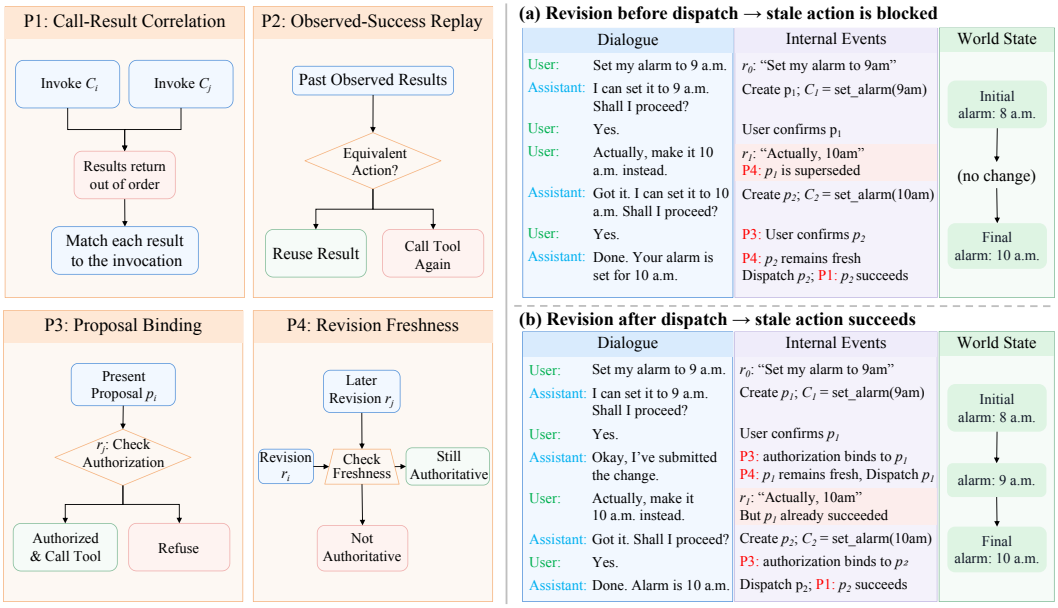

Figure 3: **Revision-bound grounding under asynchronous execution.** Left, P1–P4 bind asynchronous results, reusable observations, user authorization, and pending work to their respective invocation, canonical action, proposal, and dialogue revision. Right, a revision supersedes an undispatched action, whereas after dispatch, any verified effect is incorporated into world state and execution continues from that state under the revised intent.

To resolve this decision in practice, the runtime combines deterministic rules with semantic classification. Deterministic rules handle unambiguous acceptance and rejection, while a semantic classifier handles proposal modifications and mixed intent using C_p together with the post-proposal user evidence ΔU_p. The classifier evaluates a snapshot of the proposal and user evidence. Its verdict is committed only if the captured proposal identity and user-evidence revision still match the corresponding live session state.

3.3 Revision-Bound Grounding and Execution

Revision-bound grounding controls whether asynchronous evidence and actions may affect authoritative session state. We enforce four session-local properties:

P1 Call-Result Correlation. A result is admitted only for the invocation that produced it.

P2 Observed-Success Replay. A previously observed outcome may be reused for an equivalent canonical action only if it was successful and remains applicable.

P3 Proposal Binding. Authorization applies only to the exact proposal presented for confirmation.

P4 Revision Freshness. Evidence or actions remain authoritative only while subsequent user evidence has not superseded the intent under which they were initiated.

Operationally, every asynchronous operation records its session identity and originating user-evidence revision. Tool operations additionally carry invocation and canonical-action identities, while side-effecting actions carry the proposal identity used for authorization. For non-side-effecting calls, P1 prevents reordered results from being attributed to the wrong request, while P2 permits reuse only of previously observed successful outcomes that remain applicable. For side-effecting calls, P3 binds confirmation to the exact proposal and P4 revalidates its authority immediately before dispatch, preventing rejected or modified proposals from executing under stale authorization. Evidence authority and action authority remain distinct: admitting or verbalizing evidence does not itself authorize an external side effect. This distinction also determines how the agent handles user revisions that race with an already authorized action. If a user revision arrives after an action has already been dispatched, the external effect may still occur and, once verified, remains part of W_t.

However, that outcome cannot satisfy the revised request or justify a completion claim for it. The agent instead continues from the updated dialogue and observed world state, as illustrated in Figure 3.

4 LEARNING GROUNDED FULL-DUPLEX INTERACTION

From Tool Trajectories to Grounded Duplex Speech. We construct and validate the grounded duplex corpus through four processing stages followed by a human quality audit:

1. **Trajectory Normalization.** We normalize heterogeneous tool-use data from customer service, retrieval, function calling, and multi-step agent interaction into ordered trajectories of user utterances, assistant responses, tool invocations, and tool results, while preserving cross-step dependencies and source provenance.
2. **Grounded Response Construction.** For each grounded turn, we summarize the available tool observations into a factual reference and generate a spoken continuation supported by it. Structural, grounding, and speech-suitability filtering removes invalid examples, followed by deterministic deduplication within and across sources. This yields 637,130 text conversations.
3. **Speech Rendering and Alignment.** We render the conversations as role-separated stereo speech with IndexTTS2 (Zhou et al., 2025), using a fixed assistant voice and a user voice fixed within each conversation. Word-level alignment places both channels on a shared 24-kHz timeline. Audio and alignment checks retain 623,558 conversations, and a strict reference–response correspondence check retains 614,599 for content screening.
4. **Decontamination.** We freeze seven evaluation families with 11,426 entries and screen all model-visible training text for shared sources, benchmark entities, exact and near-duplicate matches, and duplicate user-turn sequences. Removing the union of 15,188 flagged conversations leaves 599,411 conversations totaling 12,106.7 hours. Independent remapping finds no residual matches under the registered exclusion rules.
5. **Human Audit.** We audit 200 conversations stratified by data source and duration. Of these, 184 pass and 16 receive minor ratings, with no major or critical cases.

Appendix A details data provenance, speech processing, decontamination, and human review.

Speech Model Training. We initialize MOSHIAGENT-SPEECH from MoshiRAG (Chien et al., 2026) and train all parameters of its speech language model on our grounded duplex corpus, while keeping the Mimi codec frozen. Training retains the native autoregressive text-and-audio objective over continuous duplex contexts, with supporting evidence introduced asynchronously during the interaction. Supporting evidence is presented at variable points in the ongoing duplex context, exposing the model to asynchronous returns without tying evidence use to a fixed pause or turn boundary. The resulting Streaming Speaker learns to identify when external support is needed, incorporate returned evidence into ongoing speech, and sustain grounded multi-turn progression across dialogue and tool-mediated steps, while semantic tool selection and execution remain behind the revision-bound grounding interface and client-executed tool boundary. At deployment, MOSHIAGENT-SPEECH remains fixed while reasoning backends and execution policies operate through the same interface.

5 EXPERIMENTS

5.1 TOOLRACEBENCH

TOOLRACEBENCH is a scenario-based benchmark built on τ-Voice for stateful tool use. We construct it by identifying representative state-sensitive interaction patterns in τ-Voice and retaining their dialogue context, task state, tool definitions, task entities, and action semantics. We then turn these interaction windows into independently replayable scenarios by controlling conditions that determine whether an action or observation remains valid, including user corrections, authorization changes, prior execution status, invocation or proposal identity, concurrent requests, delayed or reordered results, and dependencies between task steps. Each scenario specifies an initial state, the relevant dialogue, a fixed event trace, and a behavioral oracle over observable actions, external state, and user-facing behavior. Paired scenarios vary decision-relevant conditions while preserving the

Table 1: **Main comparison across voice-agent capabilities.** We report stateful task completion on τ-Voice, tool use on Full-Duplex-Bench v3 (FDB-v3) and AU-BFCL, audio reasoning on Big Bench Audio (BBA) and VoiceBench, and full-duplex response latency. "–" denotes an unreported or unavailable comparable result, whereas N/A denotes an inapplicable evaluation.

	Real-world Task	Tool Use		Audio Reasoning		End-to-end Latency	
System	τ-Voice Macro Pass@1 ↑	FDB-v3 Pass@1 ↑	AU-BFCL Average ↑	BBA Accuracy ↑	VoiceBench Overall ↑	Smooth Turn Latency (ms) ↓	Interruption Latency (ms) ↓
Hosted Real-Time APIs (architecture undisclosed)							
GPT-Realtime-2.1 (high)	45.7	–	–	95.8	–	–	–
GPT-Realtime-2.1	38.0	–	–	86.8	–	–	–
GPT-Realtime-2.1 mini (high)	29.4	–	–	74.7	–	–	–
GPT-Realtime-2.1 mini	22.5	–	–	62.7	–	–	–
Gemini 3.1 Flash Live (high)	37.7	54.0	–	96.6	–	940.0	651.0
Gemini 3.1 Flash Live	26.2	48.0	–	71.3	–	567.0	337.0
Qwen3.5-Omni-Plus-Realtime	53.7	50.0	–	98.7	–	2140.0	–
Full-Duplex Speech Models							
Moshi	N/A	N/A	N/A	4.4	29.5	265.0	257.0
PersonaPlex-7B-v1	N/A	N/A	N/A	19.1	30.6	170.0	240.0
VoiceChat-11B	18.5	33.0	56.1	40.9	45.8	448.0	480.0
Non-Duplex Audio-Language Models							
GPT-4o-mini-audio-preview	N/A	N/A	86.7	65.0	82.8	N/A	N/A
Ultravox-v0.6-LLaMA-3.3-70B	N/A	N/A	84.4	75.2	81.8	N/A	N/A
Full-Duplex Agent Systems							
MoshiRAG	N/A	N/A	N/A	37.2	–	180.0	1020.0
DuplexCascade	N/A	N/A	N/A	–	65.4	1724.0	1225.0
DuplexOmni	16.7	32.0	54.3	77.2	41.5	246.0	510.0
MoshiAgent (ours)	**32.1**	**67.0**	**73.7**	**98.8**	**73.5**	**184.0**	**388.0**

Table 2: **Generalization across reasoning backends.** We vary only the reasoning backend, with no backend-specific speech retraining.

	Real-world task	Tool use		Audio reasoning		End-to-end Latency	
Reasoning Backend	τ-Voice Macro Pass@1 ↑	FDB-v3 Pass@1 ↑	AU-BFCL Average ↑	BBA Accuracy ↑	VoiceBench Overall ↑	Smooth Turn Latency (ms) ↓	Interruption Latency (ms) ↓
Gemma-4-31B-it	**32.1**	**67.0**	73.7	**98.8**	**73.5**	184	388
Muse-Glimmer-30B	28.9	61.0	**77.1**	97.7	72.7	**146**	325
LFM2.5-2.6B	20.8	50.0	68.7	88.7	52.3	196	297
Gemma-4-26B-A4B-it	22.5	48.0	62.5	98.2	67.4	147	**252**

underlying task context, enabling controlled evaluation of asynchronous result attribution, outcome reuse, authorization, user revision, multi-step task progression, and concurrent tool use. Whereas τ-Voice evaluates whether a complete dialogue reaches the target environment state, TOOLRACEBENCH evaluates whether the agent remains behaviorally consistent at state-sensitive decision points within those interactions. Full construction details and representative scenarios are provided in Appendix D.

5.2 EXPERIMENTAL SETUP

Other Benchmarks. We evaluate MOSHIAGENT across six audio-agent benchmark families covering complementary aspects of spoken interaction. For stateful task completion in real-world, we use τ-Voice across its Airline, Retail, and Telecom domains. For tool use, we evaluate on Full-Duplex-Bench v3 (Lin et al., 2026) and the audio version of BFCL-v3 in AU-Harness (Nguyen et al., 2026). We further use Big Bench Audio (Hill-Smith & Cameron, 2024) and VoiceBench (Chen et al., 2026a) to measure general audio understanding and reasoning, and Full-Duplex-Bench 1.0 to evaluate interactive full-duplex behavior. We follow the official evaluation procedure and metric of each benchmark.

Systems. Our system is MOSHIAGENT, whose Streaming Speaker is instantiated by MOSHIAGENT-SPEECH and coupled to the revision-bound grounding interface described in Section 3. Unless otherwise stated, we use Gemma-4-31B-it (Team et al., 2026) as the reasoning backend. We compare against hosted real-time voice APIs with undisclosed internal architectures, native full-duplex speech models, non-duplex audio-language models, and modular full-duplex agent systems.

Table 3: **Controlled comparison of speech and execution policies.** We cross the base MoshiRAG speaker and MOSHIAGENT-SPEECH with either a direct tool loop or revision-bound execution while keeping the reasoning backend, tool schemas, and evaluation inputs fixed. We report end-to-end task metrics together with performance on TOOLRACEBENCH.

		FDB-v3					τ-Voice				TOOLRACEBENCH
Speaker	Execution	Arg. Acc.	Pass@1	Easy	Medium	Hard	Airline	Retail	Telecom	Pass@1	Pass@1
MoshiRAG	Direct Tool Loop	60.5	50.0	66.7	55.9	23.3	44.7	5.3	0.9	16.9	9.7
MoshiRAG	Revision-Bound Execution	60.5	46.0	63.9	52.9	16.7	46.7	6.4	1.2	18.1	29.2
MOSHIAGENT-SPEECH	Direct Tool Loop	65.0	56.0	66.7	58.8	40.0	48.0	6.1	1.8	18.6	34.2
MOSHIAGENT-SPEECH	Revision-Bound Execution	**77.6**	**67.0**	**69.4**	**70.6**	**60.0**	**49.3**	**31.3**	**15.8**	**32.1**	**83.2**

Table 4: **Component ablations.** We remove individual components from the complete MOSHIAGENT system while keeping the remaining stack fixed, and report stateful task completion on τ-Voice together with TOOLRACEBENCH Pass@1.

Configuration	Airline	Retail	Telecom	τ-Voice Pass@1	TOOLRACEBENCH Pass@1
Full system	**49.3**	**31.3**	**15.8**	**32.1**	**83.2**
w/o Reasoning Coordinator	36.0	27.5	14.3	25.9 (-6.2)	49.0 (-34.2)
w/o Deliberative Explorer	40.7	28.7	12.6	27.3 (-4.8)	62.1 (-21.1)
w/o Authoritative Session State	42.7	27.8	14.0	28.2 (-4.0)	67.8 (-15.4)
w/o Revision-Bound Write Control	38.7	27.2	13.2	26.3 (-5.8)	50.7 (-32.6)
w/o Evidence Admission	43.3	28.4	10.5	27.4 (-4.7)	42.6 (-40.6)
w/o Reasoner History	48.0	5.6	5.3	19.6 (-12.5)	59.7 (-23.5)

5.3 END-TO-END EVALUATION

Overall Performance. Table 1 compares MOSHIAGENT with representative real-time voice APIs, full-duplex speech models, audio-language models, and modular voice-agent systems across the six benchmark families. Using Gemma-4-31B-it as the default reasoning backend, MOSHIAGENT achieves 32.14 Macro Pass@1 on τ-Voice, 67.0 Pass@1 on Full-Duplex-Bench v3, 73.7 on AU-BFCL, 98.8 accuracy on Big Bench Audio, and 73.5 overall on VoiceBench. The same system therefore supports stateful task completion, tool use, and general audio reasoning without requiring benchmark-specific architectures.

Full-Duplex Latency and Responsiveness. End-to-end capability does not come at the cost of serializing the speech interaction around reasoning. MOSHIAGENT reaches a smooth-turn latency of 184 ms and an interruption latency of 388 ms while retaining the same revision-bound execution path used for tool-mediated tasks. This behavior is consistent with the system decomposition in Section 3: the Streaming Speaker remains on the acoustic critical path, while reactive and deliberative computation can proceed asynchronously and return evidence without stopping or restarting speech generation. Thus, longer-running semantic work changes when evidence becomes available rather than forcing the conversation itself to wait.

Generalization across Reasoning Backends. As shown in Table 2, we keep MOSHIAGENT-SPEECH and the revision-bound execution stack fixed and replace only the reasoning backend. All four operate through the same interface without backend-specific speech retraining, although their capability profiles differ. Gemma-4-31B-it gives the strongest results on τ-Voice, Full-Duplex-Bench v3, Big Bench Audio, and VoiceBench among the reported backends. Muse-Glimmer-30B performs best on AU-BFCL and yields the lowest smooth-turn latency, while Gemma-4-26B-A4B-it gives the lowest interruption latency. The 2.6B LFM2.5 backend remains compatible with the same speech and execution stack, but with lower task-level performance. These results demonstrate backend portability rather than backend invariance: the grounding interface can remain fixed across reasoning models, while reasoning quality continues to affect end-to-end capability.

5.4 ABLATIONS AND GROUNDING ANALYSIS

Controlled Speech and Execution Comparison. Table 3 crosses two speakers with two execution policies while holding the reasoning backend, tool schemas, and evaluation inputs fixed. On TOOL-

RACEBENCH, revision-bound execution improves MoshiRAG from 9.7 to 29.2 and MOSHIAGENT-SPEECH from 34.2 to 83.2. Speech training also provides substantial gains under either execution policy, improving the direct tool loop from 9.7 to 34.2 and revision-bound execution from 29.2 to 83.2. These results show that the trained speaker and revision-bound execution make complementary contributions to robustness under asynchronous tool races. Their effects on end-to-end task performance are less uniform. With MoshiRAG, revision-bound execution decreases Full-Duplex-Bench v3 Pass@1 from 50.0 to 46.0 while increasing τ-Voice Macro Pass@1 from 16.9 to 18.1. With MOSHIAGENT-SPEECH, it instead increases Full-Duplex-Bench v3 argument accuracy from 65.0 to 77.6 and strict Pass@1 from 56.0 to 67.0, while raising τ-Voice Macro Pass@1 from 18.6 to 32.1. Revision-bound execution provides consistent gains on the controlled race evaluation, while stronger end-to-end gains emerge when paired with a speaker trained to incorporate asynchronous evidence.

Component Ablations on Real-World Tasks. We remove individual components from the complete system while keeping the remaining stack unchanged. Table 4 shows that every removal reduces τ-Voice Macro Pass@1. Removing Reasoner History causes the largest drop, from 32.1 to 19.6, with most of the degradation concentrated in Retail and Telecom. Removing the Reasoning Coordinator and Revision-Bound Write Control reduces Macro Pass@1 to 25.9 and 26.3, respectively, while removing the Deliberative Explorer, Evidence Admission, or Authoritative Session State yields scores between 27.3 and 28.2. These results indicate that performance depends on both multi-timescale reasoning and the state and execution mechanisms that keep asynchronous computation aligned with the active dialogue revision. Here, Reasoner History denotes only the conversational history supplied to the reasoning backend, and the Streaming Speaker retains its native conversational context throughout. TOOLRACEBENCH reveals a different sensitivity profile. The full system achieves 83.2, while every component removal substantially reduces performance. Removing Evidence Admission causes the largest drop to 42.6, followed by the Reasoning Coordinator at 49.0 and Revision-Bound Write Control at 50.7. Removing Reasoner History, the Deliberative Explorer, or Authoritative Session State yields 59.7, 62.1, and 67.8, respectively. The contrast with τ-Voice is informative. Reasoner History is most important for end-to-end task completion, whereas Evidence Admission, Write Control, and coordination are more critical under asynchronous tool races.

Residual Failure Analysis. We inspect all 599 failed τ-Voice trajectories of the complete MOSHIAGENT system and assign each a single primary cause, defined as the earliest unrecovered error on the causal path to task failure. *Argument Construction* (232, 38.7%) and *Action Selection* (192, 32.1%) dominate, together accounting for 70.8% of failures. *Evidence Utilization* contributes 64 cases (10.7%), while *Response Generation* accounts for 90 (15.0%). Only 21 cases (3.5%) arise from *Intent Interpretation* or *Task-State Tracking*. Notably, no failure is primarily attributed to *Grounding Enforcement*, and a separate P1–P4 audit finds no grounding violation as the first decisive error. Common failures instead involve incorrect identifiers or transaction arguments, incomplete compound actions, unsupported action choices, misuse of valid tool observations, and incorrect or premature spoken realization. The remaining τ-Voice gap therefore lies primarily in semantic action selection, argument construction, evidence use, and response generation rather than revision-bound grounding.

6 CONCLUSION

In this paper, we study a reliability challenge in full-duplex voice agents, where user revisions can race with asynchronous reasoning and tool execution. We introduce revision-bound grounding, which binds asynchronous evidence and actions to the dialogue revision from which they originate. We instantiate this design in MOSHIAGENT and train MOSHIAGENT-SPEECH on 599K tool-grounded duplex conversations totaling 12.1K hours. Across seven audio-agent benchmark families, MOSHIAGENT supports stateful task completion, tool use, audio reasoning, and responsive full-duplex interaction. With MOSHIAGENT-SPEECH fixed, revision-bound execution improves Full-Duplex-Bench v3 argument accuracy by 12.6 points and strict Pass@1 by 11.0 points, while the complete system reaches 83.2 Pass@1 on TOOLRACEBENCH. These results highlight the importance of keeping evolving user intent, external actions, and observed outcomes causally aligned.

AI USE STATEMENT

AI models, including LLMs, are used as part of the systems studied in this work and in our synthetic data generation pipeline. In particular, AI models are used to generate synthetic dialogue and tool-interaction data for the tool-grounded duplex corpus used in training. Details of the models, data generation procedures, and experimental use of these data are provided in the main paper and appendix. LLMs were additionally used for language editing and polishing of the manuscript. All research methodology, experimental analysis, and scientific conclusions were reviewed and verified by the authors, who take full responsibility for the final manuscript.

REFERENCES

Liquid AI. Lfm2.5-2.6b: Agents everywhere. *Liquid AI Blog*, 2026. www.liquid.ai/blog/lfm2-5-2-6b.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik R Narasimhan. $tauˆ2$-bench: Evaluating conversational agents in a dual-control environment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=OC2z7iSQKa.

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

Mingyang Chen, sunhaoze, Tianpeng Li, Fan Yang, Hao Liang, KeerLu, Bin CUI, Wentao Zhang, Zenan Zhou, and Weipeng Chen. Facilitating multi-turn function calling for llms via compositional instruction tuning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 42742–42782, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/69c49f75ca31620f1f0d38093d9f3d9b-Paper-Conference.pdf.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. VoiceBench: Benchmarking LLM-based voice assistants. *Transactions of the Association for Computational Linguistics*, 14:378–398, 2026a. doi: 10.1162/tacl.a.628. URL https://aclanthology.org/2026.tacl-1.18/.

Zheng Chen, Hanqing Liu, Duling Xu, Dong Dong, Jialin Li, Bangzheng Pu, and Jidong Zhai. Cordon: Semantic transactions for tool-using llm agents, 2026b. URL https://arxiv.org/abs/2606.17573.

Chung-Ming Chien, Manu Orsini, Eugene Kharitonov, Neil Zeghidour, Karen Livescu, and Alexandre Défossez. MoshiRAG: Asynchronous knowledge retrieval for full-duplex speech language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=4aI2vOyyHH.

Konstantina Christakopoulou, Shibl Mourad, and Maja Mataric. Agents thinking fast and slow: A talker-reasoner architecture. In *NeurIPS 2024 Workshop on Open-World Agents*, 2024. URL https://openreview.net/forum?id=xPhcP6rbI4.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue, 2024. URL https://arxiv.org/abs/2410.00037.

Linfeng Fan, Ziwei Li, Yuan Tian, Yichen Wang, Rongsheng Li, and Xiong Wang. The granularity mismatch in agent security: Argument-level provenance solves enforcement and isolates the llm reasoning bottleneck, 2026. URL https://arxiv.org/abs/2605.11039.

Fixie AI. Ultravox-v0.6-LLaMA-3.3-70B. Hugging Face model, 2025. URL https://huggingface.co/fixie-ai/ultravox-v0_6-llama-3_3-70b. Released August 18, 2025.

Glaive AI. Glaive function calling v2. Hugging Face dataset, 2023. URL https://huggingface.co/datasets/glaiveai/glaive-function-calling-v2.

Google DeepMind. Gemini 3.1 Flash Audio (Flash Live, TTS). Model Card, March 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-flash-audio/. Includes Gemini 3.1 Flash Live.

Micah Hill-Smith and George Cameron. Evaluating audio reasoning with Big Bench Audio. Hugging Face Blog, December 2024. URL https://huggingface.co/blog/big-bench-audio-release.

Muye Huang, Lingling Zhang, Xingyu Yu, Lei Shi, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Jun Liu. Duplexomni: Real-time listening, seeing, thinking, and speaking for full-duplex interaction, 2026. URL https://arxiv.org/abs/2606.09186.

Dhruv Jain, Harshit Shukla, Gautam Rajeev, Ashish Kulkarni, Chandra Khatri, and Shubham Agarwal. Voiceagentbench: Are voice assistants ready for agentic tasks?, 2026. URL https://arxiv.org/abs/2510.07978.

Tianle Liang, Yifu Chen, Shengpeng Ji, Yijun Chen, Zhiyang Jia, Jingyu Lu, Fan Zhuo, Xueyi Pu, Yangzhuo Li, and Zhou Zhao. VoxMind: An end-to-end agentic spoken dialogue system. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10117–10139, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.459. URL https://aclanthology.org/2026.acl-long.459/.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H. Liu, and Hung-Yi Lee. Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. In *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8, 2025. doi: 10.1109/ASRU65441.2025.11433838.

Guan-Ting Lin, Chen Chen, Zhehuai Chen, and Hung yi Lee. Full-duplex-bench-v3: Benchmarking tool use for full-duplex voice agents under real-world disfluency, 2026. URL https://arxiv.org/abs/2604.04847.

Weiwen Liu, Xu Huang, Xingshan Zeng, xinlong hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong WANG, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Wang Xinzhi, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8EB8k6DdCU.

Yanming Liu, Xinyue Peng, Jiannan Cao, Xinyi Wang, Songhang Deng, Jintao Chen, Jianwei Yin, and Xuhong Zhang. ToolGate: Contract-grounded and verified tool execution for LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 9653–9684, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.470. URL https://aclanthology.org/2026.findings-acl.470/.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 54463–54482. Curran Associates, Inc., 2024. doi: 10.52202/079017-1725. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/61cce86d180b1184949e58939c4f983d-Paper-Datasets_and_Benchmarks_Track.pdf.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Leander Melroy Maben, Gayathri Ganesh Lakshmy, Srijith Radhakrishnan, Siddhant Arora, and Shinji Watanabe. Aura: Agent for understanding, reasoning, and automated tool use in voice-driven tasks. In *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–4, 2025. doi: 10.1109/ASRU65441.2025.11434725.

Meta Superintelligence Lab. Muse Glimmer 30B. Hugging Face Model Card, August 2026. URL https://huggingface.co/meta-models/Muse-Glimmer-30B. Model: meta-models/Muse-Glimmer-30B.

Hoang Nguyen, Sidharth Surapaneni, Akshay Kalkunte, Jash Mehta, Aman Tiwari, Oluwanifemi Bamgbose, Khyati Mahajan, Jash Shah, Shruthan Radhakrishna, Sathwik Tejaswi Madhusudhan, Vikas Yadav, and Sai Rajeswar. Au-harness: An open-source toolkit for holistic evaluation of audio llms, 2026. URL https://arxiv.org/abs/2509.08031.

NVIDIA. NVIDIA NemotronLabs VoiceChat 11B. NVIDIA NGC model card, 2026. URL https://catalog.ngc.nvidia.com/orgs/nim/nvidia/models/nemotron-labs-voicechat. Version 1.0.0; official model release July 16, 2026. This is a model card, not a standalone paper.

NVIDIA Corporation. Nemotron-sft-agentic-v2. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/nvidia/Nemotron-SFT-Agentic-v2. Includes tool-calling, search, and customer-service subsets.

NVIDIA Corporation. Nemotron-rl-agentic-function-calling-pivot-v1. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/nvidia/Nemotron-RL-Agentic-Function-Calling-Pivot-v1.

OpenAI. GPT-4o Mini Audio. OpenAI API Documentation, 2025. URL https://developers.openai.com/api/docs/models/gpt-4o-mini-audio-preview.

OpenAI. GPT-Realtime-2.1. OpenAI API Documentation, 2026a. URL https://developers.openai.com/api/docs/models/gpt-realtime-2.1. Model ID: gpt-realtime-2.1.

OpenAI. GPT-Realtime-2.1 Mini. OpenAI API Documentation, 2026b. URL https://developers.openai.com/api/docs/models/gpt-realtime-2.1-mini. Model ID: gpt-realtime-2.1-mini.

Ramit Pahwa, Apoorva Beedu, Parivesh Priye, Rutu Gandhi, Saloni Takawale, Aruna Baijal, and Zengli Yang. Audio2tool: Speak, call, act – a dataset for benchmarking speech tool use, 2026. URL https://arxiv.org/abs/2604.22821.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 126544–126565. Curran Associates, Inc., 2024. doi: 10.52202/079017-4020. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e4c61f578ff07830f5c37378dd3ecb0d-Paper-Conference.pdf.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Manoj Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-2184. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5e3661f7fe4c8ac5652d62eb3d3c96ea-Paper-Datasets_and_Benchmarks_Track.pdf.

Suliu Qin, Haomin Zhuang, Yujun Zhou, Yufei Han, and Xiangliang Zhang. Airguard: Guarding agent actions with runtime authority control, 2026. URL https://arxiv.org/abs/2605.28914.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 9695–9717, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/28e50ee5b72e90b50e7196fde8ea260e-Paper-Conference.pdf.

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

Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. When2Call: When (not) to call tools. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3391–3409, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.174. URL https://aclanthology.org/2025.naacl-long.174/.

Rajarshi Roy, Jonathan Raiman, Sang-Gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. Personaplex: Voice and role control for full duplex conversational speech models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16137–16141, 2026. doi: 10.1109/ICASSP55912.2026.11463413.

Igor Santos-Grueiro. Temporary authority, permanent effects: Commit-time authorization for llm agents, 2026a. URL https://arxiv.org/abs/2607.10487.

Igor Santos-Grueiro. Context-to-execution integrity for llm agents, 2026b. URL https://arxiv.org/abs/2607.06000.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.

Tianneng Shi, Jingxuan He, Zhun Wang, Hongwei Li, Linyu Wu, Wenbo Guo, and Dawn Song. Progent: Securing ai agents with privilege control, 2026. URL https://arxiv.org/abs/2504.11703.

Vidya Srinivas, Zachary Englhardt, Vikram Iyer, and Shwetak Patel. Thinking while speaking: Inference-time knowledge transfer for responsive and intelligent conversational voice agents, 2026. URL https://arxiv.org/abs/2511.07397.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Qwen Team. Qwen3.5-omni technical report, 2026. URL https://arxiv.org/abs/2604.15804.

Xiong Wang, Yangze Li, Chaoyou Fu, Yike Zhang, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen LLM. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63345–63354. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wang25aw.html.

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

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

Jinghan Xu, Longze Fan, Zeyuan Wang, Xinjin Li, and Hankai Liu. Beyond single-use tokens: Durable authorization state for replay-resistant llm agent actions, 2026. URL https://arxiv.org/abs/2608.01710.

Zhangchen Xu, Adriana Meza Soria, Shawn Tan, Anurag Roy, Ashish Sunil Agrawal, Radha Poovendran, and Rameswar Panda. Toucan: Synthesizing 1.5m tool-agentic data from real-world mcp environments, 2025b. URL https://arxiv.org/abs/2510.01179.

Chen Yang, Ran Le, Yun Xing, Zhenwei An, Zongchao Chen, Wayne Xin Zhao, Yang Song, and Tao Zhang. Toolmind technical report: A large-scale, reasoning-enhanced tool-use dataset, 2025. URL https://arxiv.org/abs/2511.15718.

Jianing Yang, Yusuke Fujita, and Yui Sudo. Duplexcascade: Full-duplex speech-to-speech dialogue with vad-free cascaded asr-llm-tts pipeline and micro-turn optimization, 2026. URL https://arxiv.org/abs/2603.09180.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. tau-bench: A benchmark for tool- agent- user interaction in real-world domains. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 9965–10017, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b126cc38b8638e07bef37e7b2bb72bf-Paper-Conference.pdf.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot, 2024. URL https://arxiv.org/abs/2412.02612.

Haoyang Zhang, Jun Chen, Donghang Wu, Yuxin Li, Yuxin Zhang, Xiangyu Tony Zhang, Che Liu, Qingjian Lin, Yizhou Peng, Hexin Liu, Eng Siong Chng, Chao Yan, Boyong Wu, Yechang Huang, Xuerui Yang, and Fei Tian. Duplexsla: A full-duplex spoken language model with synchronized speech, language, and action, 2026. URL https://arxiv.org/abs/2605.20755.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chao-Hong Tan, Zhihao Du, and ShiLiang Zhang. OmniFlatten: An end-to-end GPT model for seamless voice conversation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14570–14580, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.709. URL https://aclanthology.org/2025.acl-long.709/.

Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. Indextts2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech, 2025. URL https://arxiv.org/abs/2506.21619.

Wenhao Zou, Yuwei Miao, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Jingwen Xu. Lts-voiceagent: A listen-think-speak framework for efficient streaming voice interaction via semantic triggering and incremental reasoning, 2026. URL https://arxiv.org/abs/2601.19952.

APPENDIX

A DATA CONSTRUCTION AND QUALITY CONTROL

This appendix provides the complete construction protocol for the tool-grounded duplex corpus described in Section 4. It covers source processing, trajectory normalization, grounded response construction, speech synthesis and alignment, evaluation decontamination, and human quality review. Figure 4 summarizes the complete process.

A.1 DATA SOURCES AND TRAJECTORY NORMALIZATION

We construct the corpus from heterogeneous tool-use datasets covering customer service, retrieval, function selection, argument generation, multi-step tool use, and tool-grounded response generation. Each source is converted into a common ordered trajectory of user utterances, assistant responses, tool invocations, and tool results. We preserve source provenance and cross-step dependencies while removing source-specific message formats.

Each normalized trajectory retains the source identifier, source-local sample identifier, task domain or topic, system context and tool definitions, ordered interaction events, upstream provenance, and generated grounding fields. Provenance records include the original split, row identifier, and parent-data metadata required for subsequent source-level filtering and auditing.

Table 5 summarizes the retained sources and their final contribution to the training corpus. APIGen-MT (Prabhakar et al., 2025) and the τ-bench (Yao et al., 2025) training split are removed during evaluation isolation because they share task databases with the target τ evaluations. When2Call (Ross et al., 2025) contributes negative examples without grounded tool results and therefore does not enter the final grounded reference-response corpus.

After normalization and text-level quality filtering, we perform deterministic deduplication using normalized user-turn sequences. Deduplication is applied first within each source and then across the merged mixture. For a conversation x with eligible user turns $u_1, \ldots, u_k$, we define

$$S(x) = [N(u_1), N(u_2), \ldots, N(u_k)], \tag{4}$$

where $N(\cdot)$ normalizes Unicode, case, whitespace, and punctuation. When multiple conversations share a signature, we retain the version with more grounded turns, breaking further ties by total turn count. This procedure produces 637,130 text conversations, with an average of 4.21 turns and 1.72 grounded turns per conversation. Of these, 268,036 conversations contain multiple grounded turns.

License Review. We conduct a source-level license review for every upstream dataset represented in the final training corpus. Toucan-1.5M (Xu et al., 2025b), ToolACE (Liu et al., 2025), ToolMind/GraphSyn (Yang et al., 2025), and Glaive function-calling-v2 (Glaive AI, 2023) are distributed under Apache-2.0, while Nemotron-SFT Agentic, the four Nemotron-RL agent/tool-use datasets (NVIDIA Corporation, 2025; 2026), xLAM function-calling (Liu et al., 2024), and BUTTONInstruct (Chen et al., 2025b) are distributed under CC BY 4.0. The search subset inherits its provenance and license from Nemotron-SFT Agentic. For each retained source, we record the upstream dataset, license, attribution, and transformation history, and preserve the corresponding license text, source attribution, and modification notices required for derivative research use.

A.2 GROUNDED CONVERSATION CONSTRUCTION

For each normalized trajectory, we construct a factual reference from the tool evidence available at the corresponding point in the interaction and generate a spoken continuation supported by that reference. The construction is designed to preserve the causal relation between accumulated tool evidence and the conversational response it supports.

The processing sequence is as follows. Recoverable placeholders are first resolved, while examples with unrecoverable task-critical arguments are discarded. Accumulated tool observations are then compressed into a factual reference without introducing recommendations or unsupported explanations. Dates, monetary values, identifiers, and named entities receive additional grounding checks.

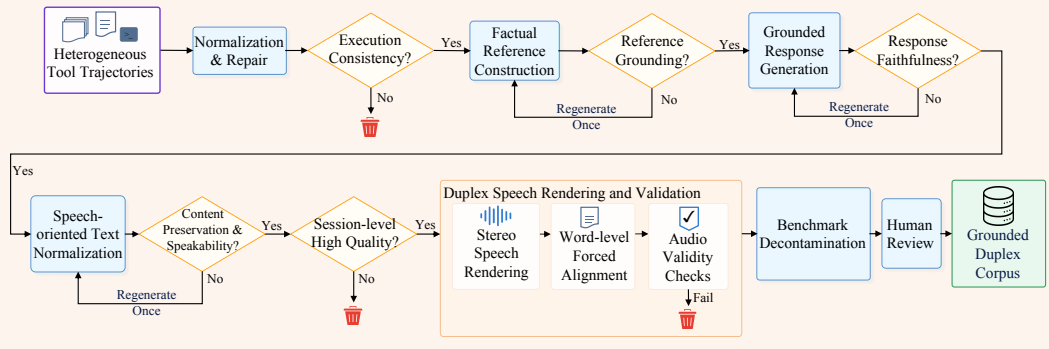

Figure 4: **Detailed grounded-duplex data construction pipeline.** Heterogeneous tool trajectories are normalized, converted into factual references and grounded spoken responses, rendered as duplex speech, and filtered for audio validity and evaluation contamination. Regeneration is bounded to a single retry where indicated. Human review is a post-construction audit rather than a filtering stage over the full corpus.

Table 5: **Training-data sources and final corpus composition.** Counts refer to the final content-clean training corpus. Percentages are rounded independently to three decimal places.

Source	Upstream Data	Processing Summary	License	Conversations	Share
Toucan	Toucan-1.5M	Quality, modality, and tool-result filtering	Apache-2.0	253,317	42.261%
Nemotron	Nemotron-SFT Agentic	Unified interactive-agent and tool-calling splits	CC BY 4.0	190,269	31.743%
GraphSyn	ToolMind GraphSyn	Hidden reasoning removed, tool interactions retained	Apache-2.0	47,553	7.933%
xLAM	xLAM function calling	Queries and function outputs expanded to grounded turns	CC BY 4.0	45,423	7.578%
Nemotron-RL	Four agent/tool-use datasets	Step-wise trajectories merged and schemas normalized	CC BY 4.0	39,175	6.536%
Glaive	Glaive function-calling-v2	Missing execution results completed where recoverable	Apache-2.0	9,655	1.611%
BUTTON	BUTTONInstruct	Tool execution results completed where recoverable	CC BY 4.0	7,837	1.307%
Search	Nemotron-SFT search	Retrieval trajectories normalized	CC BY 4.0	5,698	0.951%
ToolACE	ToolACE	Function-call DSL parsed and multi-turn structure recovered	Apache-2.0	484	0.081%
Total				**599,411**	**100%**

The response is generated as a natural spoken interaction containing a pre-evidence continuation, a reference-grounded continuation, and, when appropriate, a subsequent conversational continuation. We verify that the grounded portion is fully supported by its reference. A failed grounding check permits one constrained regeneration. Examples that remain unsupported, structurally malformed, or unsuitable for spoken interaction are removed.

Text generation uses Gemma-4-31B-IT with temperature 1.0, top-p 0.95, and top-k 64. Generated conversations subsequently pass a sequence of speech-oriented transformations summarized in Table 6.

A.3 Duplex Speech Rendering and Alignment

We synthesize each retained conversation once using IndexTTS2 with deterministic sampling at 24 kHz, and encode the resulting audio as 16-bit FLAC. Each conversation is rendered in stereo, with the assistant assigned to channel 0 and the user to channel 1. Conversational timing is determined by the durations of the synthesized speech segments together with stage-specific silence intervals.

We perform word-level forced alignment separately on the role-appropriate channel for each speech segment. Segment-relative timestamps are converted to the absolute conversation timeline and stored together with word identity and speaker role. We retain only conversations with readable audio, valid metadata, non-empty alignment, monotonic segment timing, and valid channel assignments. These checks retain 623,558 conversations.

A subsequent strict correspondence check ensures that grounding references and their supported responses remain structurally aligned. Sample identity and grounded-turn counts must match, references must agree after whitespace normalization, and all grounded response spans must be non-empty. This step retains 614,599 conversations. The underlying audio is unchanged by this operation.

Table 6: **Speech-oriented text processing.** Each stage preserves task-relevant factual content while making the trajectory suitable for spoken interaction.

Stage	Purpose	Failure handling
Spoken reformulation	Convert template-like text into natural telephone speech	Revert if task-critical content is lost
Mechanism-leakage repair	Remove explicit references to APIs, schemas, or backend mechanisms	Reject unresolved leakage
Speakability conversion	Verbalize dates, numbers, identifiers, and symbols	Preserve input for further screening if conversion fails
Reference repair	Repair malformed or truncated factual references	No new facts may be introduced
Residual screening	Detect unreadable numeric or symbolic remnants	Reject unresolved cases
Form-dump repair	Convert field-value dumps into natural utterances	Accept only if factual fidelity is preserved
Response merging	Merge adjacent redundant assistant responses	Apply only to semantic restatements
Session-level screening	Detect robotic, unsupported, stuck, or truncated conversations	Repair minimally or remove the conversation

Table 7: Evaluation set composition across benchmark families.

Evaluation Family	Entries
Big Bench Audio	1,000
τ-Voice	278
ToolRaceBench	298
BFCL-v3 Audio	1,240
Full-Duplex-Bench v3	100
Full-Duplex-Bench 1.0	727
VoiceBench	7,783
Total	**11,426**

A.4 Evaluation Isolation and Decontamination

We define the evaluation reference set before screening the training corpus. It contains 11,426 entries from the seven benchmark families used in our evaluation, as summarized in Table 7.

Decontamination combines four complementary checks. **Source-level isolation** removes training sources that share task databases with the target evaluation. **Entity-level isolation** matches normalized entities drawn from the evaluation databases, including names, orders, flights, products, and account identifiers. **Exact matching** compares normalized user turns, assistant turns, factual references, and grounded responses against benchmark text. **Near-duplicate matching** measures query-specific word 5-gram coverage after excluding common boilerplate.

The exclusion signals overlap and therefore cannot be summed directly. Table 8 reports their individual incidence. Taking the union over sample keys removes 15,188 conversations and yields a final corpus of 599,411 conversations totaling 12,106.7 hours. To verify the final corpus independently, a separate pipeline remaps all retained examples to text and reruns the source, entity, exact-match, 5-gram, and duplicate-key checks. The results of these verification checks are shown in Table 9.

A.5 Human Quality Audit

We conduct a stratified human audit of 200 conversations totaling 3.391 hours. Sampling is stratified by source and duration, with duration bins defined as short below 30 seconds, medium from 30 to 90 seconds, and long at least 90 seconds. Up to eight examples are selected from each source-duration cell using a fixed hash ordering. Each example is rated from 1 to 5 for audio integrity, text fidelity, turn timing, naturalness, and reference faithfulness. Annotators additionally record channel correctness, personally identifiable information, safety concerns, benchmark resemblance, and an overall severity label of pass, minor, major, or critical. Table 10a summarizes the resulting quality ratings and severity outcomes.

Forty examples form an agreement set and are independently annotated by two reviewers before adjudication. The remaining 160 examples are assigned between the reviewers. All disagreements involving major or critical severity, privacy, benchmark contamination, or other binary risk labels are adjudicated by a third reviewer. Numerical 1-to-5 ratings remain the mean of the first two reviewers unless a separate numerical disagreement is explicitly recorded. Four severity disagreements required adjudication, and no numerical rating required adjudication. Inter-annotator agreement is reported in

Table 8: **Content-decontamination signals.** Counts overlap across rows. The final removal count is computed from their union over conversation keys.

Exclusion signal	Matches
Shared τ-database sources	10,631
Historical-reference τ entities	5,081
Model-visible τ entities	2,103
Benchmark exact or near duplicates	1,762
Duplicate user-turn sequences	1,460
Union removed	**15,188**
Final corpus	**599,411**

Table 9: Independent decontamination verification.

Verification Criterion	Residual Matches
APIGen-MT or τ training samples	0
τ entity matches	0
Normalized benchmark exact matches	0
Word 5-gram coverage ≥ 0.5	0

Table 10b. For severity labels, we report unweighted Cohen's κ. For the ordinal 1-to-5 ratings, we report quadratic-weighted Cohen's κ. Channel correctness, PII, safety, and benchmark-resemblance labels achieve 100% agreement but have zero category variance, so no κ is reported.

B Data Processing Prompts

Several stages of corpus construction use Gemma-4-31B-IT for generation, routing, or semantics-preserving rewriting. Unless otherwise stated, all calls use temperature 1.0, top-p 0.95, and top-k 64. Runtime content is replaced below by bracketed variables such as `<REFERENCE>` and `<DIALOGUE_HISTORY>`.

B.1 Grounded Supervision Construction

Factual Reference Construction. For each grounded turn, we summarize the tool observations available at that point into a short factual reference. The reference is intended to contain only information that may support the subsequent spoken response. The model receives the dialogue history together with the available tool interactions and produces a single-line reference of at most approximately fifty words.

The prompt is:

```
You write a short factual reference for chatbot, moshi. You
see the full conversation so far, including past reference
documents, and should give a short document that helps moshi
to generate its answer to the last user turn. Be as concise
as possible. Each reference document should contain no more
than 50 words.
Rules:
- Output exactly one line: 'Reference:' followed by
concise facts (about 50 words max). No line breaks inside
the reference.
- Answer directly; no vague summaries, meta commentary, or
talking about yourself.
- Plain text: no markdown (*, #). Prefer spelled-out words
over dense symbols for numbers or simple math.
- Simple punctuation only.
```

Table 10: **Human quality audit.** The audit contains 200 source- and duration-stratified conversations. Forty examples are independently double annotated for agreement analysis.

(a) Overall quality

Metric	Result
Pass	184
Minor	16
Major	0
Critical	0
Audio integrity	4.650
Text fidelity	4.640
Turn timing	4.480
Naturalness	4.343
Reference faithfulness	4.543
Correct channel assignment	200 / 200
PII, safety, or benchmark resemblance	0

(b) Annotation agreement

Dimension	Exact agreement	Agreement coefficient
Severity	90.0%	0.792
Audio integrity	85.0%	0.851
Text fidelity	90.0%	0.883
Turn timing	80.0%	0.790
Naturalness	87.5%	0.864
Reference faithfulness	87.5%	0.861

```
Start of the conversation:
<DIALOGUE_HISTORY>
Reference:
```

The system context additionally instructs the model to use the available function-calling context rather than guess when information must be retrieved, and prohibits backend-mechanism language in the generated reference. For source branches using strengthened grounding, we add the following constraints:

```
Grounding (most important):
- Use ONLY facts that appear in the tool results above.
NEVER invent, infer, compute, or compare anything that was
not retrieved.  If the data needed to answer is not in the
tool results, say so plainly instead of making something up.
- If you summarize a list, the count must be EXACTLY right.
If the results are truncated or open-ended, write 'several'
or 'including ...', never a specific total.
- Use EVERY value the tool results DID return.  When
several results come back for a multi-part question they
correspond to the requested items IN ORDER, so attribute
each accordingly.  Only decline data that is genuinely
absent, empty, or returned an error.
```

A deterministic numeric grounding check is applied separately. Numbers in the reference must either occur in the user request or retrieved observations, or be arithmetically derivable from them.

Grounded Spoken Response Generation. Given the dialogue history and the factual reference, the model generates the next spoken assistant turn. During data construction, the turn is decomposed into a pre-evidence lead, a grounded body, and an optional tail. This decomposition is used only to

construct and align supervision and does not imply that deployment responses must expose the same textual segmentation.

```
You are generating moshi's next turn in an ongoing,
multi-turn spoken conversation between a human and the
chatbot moshi.  You see the prior conversation history; a
Reference document provides the knowledge for moshi's answer
to the latest question.
moshi's turn is an augmented turn with three labeled parts:
- (lead):  a natural opener that connects to what the user
just asked.  It needs no external information.  It MUST NOT
reveal that moshi consulted anything or is stalling.
- (body)(subtopic:  ...):  the knowledge-intensive part.
Convey the Reference's facts in the same order, spoken
naturally; do not add anything that is not in the Reference.
Talk like a CHAT AGENT rather than reading a database
output.
- (tail):  an optional short closing remark that needs no
external information; it may be empty.
Guidelines:
- The text will be read aloud by a text-to-speech model, so
it must sound like natural spoken language.
- Convert prose numbers, dates, and symbols to readable
words.  Spell identifiers and confirmation codes
letter-by-letter and digit-by-digit, tracking the Reference
exactly.
- Use only basic punctuation; no markdown, bullet points, or
tables.
- This is an ongoing conversation:  do not greet again,
answer the latest question directly, and avoid unnecessary
back-and-forth.
- Keep the body within thirty words.
- Label each part explicitly as (lead), (body)(subtopic:
...), and (tail).
VOICE RULE: You are the agent speaking to the caller.  Never
say the words 'tool', 'function', 'API', or 'system'.  If
the reference indicates information is unavailable, decline
naturally.
<DIALOGUE_HISTORY>
Reference:  <REFERENCE>
(augmented)
```

B.2 SPEECH-FORM NORMALIZATION

Speech-Form Rendering. All text that is actually rendered as speech is normalized into a form suitable for TTS. The transformation is deliberately conservative: it may change how numbers, identifiers, symbols, dates, or codes are spoken, but it may not add, drop, or paraphrase factual content. When a grounding reference is available, it is used only to determine the type and exact rendering of a value already present in the line.

```
You convert ONE line text into the exact words a
TEXT-TO-SPEECH voice would SAY ALOUD. Keep every word, name
and fact EXACTLY -- do not rephrase, summarize, add, or drop
anything; only turn written/number/symbol forms into spoken
words.  Output ONLY the rewritten line.
NUMBERS -- decide IDENTIFIER vs QUANTITY:
- IDENTIFIERS such as order, account, confirmation,
reservation, ticket, case, SKU, ZIP, or phone numbers are
read DIGIT BY DIGIT.
- Quantities, prices, dates, times, percentages, and years
are rendered in natural spoken form.
```

```
CODES & SPELLING:
- Opaque identifiers and codes are spelled letter-by-letter
and digit-by-digit.
- Email addresses, domains, and structured identifiers are
verbalized using spoken separators such as 'at', 'dot',
'underscore', and 'dash'.
SYMBOLS:
- Render symbols as their spoken forms when they carry
meaning in context.
- Remove residual markup or symbols that a speaker would not
pronounce.
LAYOUT:
- Flatten written list formatting into natural spoken text.
- Remove markup, reasoning tags, or emoji.
The result must contain no digit and no symbol other than
ordinary punctuation.
Use this FACTUAL REFERENCE ONLY to know each number/code's
exact value and what KIND of thing it is.  Do NOT add any
number, code, name or price from the reference that is not
ALREADY in the line.
REFERENCE: <REFERENCE>
Line:  <LINE>
```

C BASELINES AND REASONING BACKENDS

Baseline Systems. We compare MOSHIAGENT against representative systems spanning four classes of voice interaction architectures, following the grouping used in Table 1. For hosted real-time APIs, we evaluate GPT-Realtime-2.1 (OpenAI, 2026a) and GPT-Realtime-2.1 mini (OpenAI, 2026b), Gemini 3.1 Flash Live (Google DeepMind, 2026), and Qwen3.5-Omni-Plus-Realtime (Team, 2026). These systems expose end-to-end real-time voice interfaces, while their internal speech, reasoning, and execution architectures are not publicly specified in sufficient detail for component-level comparison. We therefore treat them as black-box end-to-end baselines and report only metrics for which a comparable evaluation is available.

For native full-duplex speech models, we include Moshi (Défossez et al., 2024), PersonaPlex-7B-v1 (Roy et al., 2026), and VoiceChat-11B (NVIDIA, 2026). These systems directly model interactive spoken dialogue and provide representative comparisons for low-latency duplex generation, but do not necessarily expose the same stateful tool-execution interface as MOSHIAGENT.

We additionally compare against non-duplex audio-language models, GPT-4o-mini-audio-preview (OpenAI, 2025) and Ultravox-v0.6-LLaMA-3.3-70B (Fixie AI, 2025). These baselines provide strong audio understanding and reasoning references, but do not operate under the same continuously interactive full duplex setting.

Finally, we compare against modular full-duplex agent systems: MoshiRAG (Chien et al., 2026), DuplexCascade (Yang et al., 2026), and DuplexOmni (Huang et al., 2026). This group is the closest architectural comparison to MOSHIAGENT because it combines continuous or streaming speech interaction with additional semantic computation outside the acoustic generation path.

Reasoning Backends. Unless otherwise stated, MOSHIAGENT uses Gemma-4-31B-it (Team et al., 2026) as its default reasoning backend. To test whether the speech and execution design depends on a particular reasoner, we additionally evaluate Muse-Glimmer-30B (Meta Superintelligence Lab, 2026), LFM2.5-2.6B (AI, 2026), and Gemma-4-26B-A4B-it (Team et al., 2026). Across this comparison, we keep MOSHIAGENT-SPEECH, the revision-bound execution stack, tool schemas, and evaluation inputs fixed, and replace only the reasoning backend.

D TOOLRACEBENCH CONSTRUCTION

TOOLRACEBENCH is a 298-scenario benchmark built on τ-Voice for stateful tool use. It preserves the task semantics, domain environments, tool capabilities, and state transitions of τ-Voice while reorganizing critical interaction windows into independently replayable scenarios. The benchmark targets decisions that arise when dialogue, tool execution, and external outcomes evolve concurrently, including result attribution, outcome reuse, authorization, user revision, multi-step task progression, and concurrent tool use.

TOOLRACEBENCH complements the end-to-end task evaluation in τ-Voice. Whereas τ-Voice evaluates whether a complete dialogue reaches the target environment state, TOOLRACEBENCH evaluates whether the agent behaves correctly at individual state-sensitive decision points within such interactions.

D.1 SCENARIO CONSTRUCTION

We identify representative interaction patterns through manual analysis of τ-Voice task contexts. For each selected context, we retain the relevant dialogue, task state, tool definitions, task entities, and action semantics. We then construct a controlled interaction window by varying conditions that can change whether an action or observation remains valid. These variations include user corrections, authorization changes, prior execution status, invocation or proposal identity, concurrent requests, delayed or reordered results, and dependencies between task steps.

Each scenario specifies four elements. The *initial state* records the task and interaction state before the evaluated window. The *dialogue* contains the user evidence relevant to the decision. The *event trace* specifies the order in which tool outcomes, user revisions, and other asynchronous events become available. The *behavioral oracle* specifies the observable conditions required for a correct outcome.

D.2 REPRESENTATIVE BENCHMARK SCENARIOS

D.2.1 EXAMPLE 1

Initial State. Two independent order-status queries are pending for `W2378156` and `W9999999`.

Dialogue.

```
User: Please check orders W2378156 and W9999999.
```

Event Trace.

```
c1: get_order_details(W2378156)
c2: get_order_details(W9999999)

c2 returns first:
status = cancelled

c1 returns later:
status = delivered
```

Behavioral Oracle. Each returned result must remain associated with the invocation that produced it, independent of completion order. Thus, `c1` may support only a status for `W2378156`, and `c2` may support only a status for `W9999999`. A valid response may report that `W2378156` was delivered and `W9999999` was cancelled. Attributing the results according to completion order constitutes a failure.

D.2.2 EXAMPLE 2

Initial State. The user requests the status of `W2378156`. Between repeated requests, the task state relevant to the query does not change.

Dialogue.

```
User: What is the status of order W2378156?
Assistant: Let me check.

User: Could you check W2378156 again?
```

For the failure-control variant, the second request is instead:

```
User: Please try again.
```

Event Trace. In the successful variant,

```
c1: get_order_details(W2378156)

c1 returns:
status = delivered

Assistant: Order W2378156 has been delivered.
```

In the failure-control variant,

```
c1: get_order_details(W2378156)

c1 returns:
temporary failure

Assistant: I could not retrieve it.
```

Behavioral Oracle. A previously observed successful result may be reused for the same request while the relevant task state remains unchanged, or the agent may issue the same read again. In either case, any reported status must be supported by a successful observation for `W2378156`. Reusing a successful result does not require a second client dispatch, although replay utilization is recorded only as a diagnostic. An unsuccessful outcome cannot support an order status. After an explicit retry following a temporary failure, the agent must obtain a successful observation before reporting the status.

D.2.3 EXAMPLE 3

Initial State. A cancellation proposal is pending for order `W2378156`, with reason `no longer needed`. No cancellation has yet been dispatched.

Dialogue.

```
User: Yes, cancel order W2378156.
```

Event Trace. The user confirmation is received while the specified cancellation proposal is pending. A write invocation may subsequently be prepared for dispatch.

Behavioral Oracle. The confirmation authorizes only the pending cancellation of `W2378156` with the task-relevant arguments represented by that proposal. It must not authorize an invocation with a different action, target, or other task-relevant arguments. Dispatching such a modified invocation on the basis of the same confirmation constitutes a failure.

D.2.4 EXAMPLE 4

Initial State. A replacement is pending for order `W1000001`, replacing both the bottle and lamp using `credit_card_1234`. The replacement has not yet been dispatched.

Dialogue.

```
User: No bottle, just the lamp.
```

Event Trace. The user correction is received before the pending two-item replacement reaches the external tool boundary.

Behavioral Oracle. After the correction is observed, the original two-item replacement must not be dispatched. Any subsequently dispatched replacement must reflect the revised scope and include only the lamp. Dispatching the superseded two-item replacement constitutes a failure.

D.2.5 EXAMPLE 5

Initial State. A return is pending for item `6065192424` in order `W9571698`, with refund destination `gift_card_7250692`. The user has confirmed the proposal, but no external write has yet been dispatched.

Dialogue.

```
User: Yes, go ahead.

User: Actually, wait.
```

Event Trace. The confirmation is received first. The withdrawal is then received before the prepared return reaches the external tool boundary.

Behavioral Oracle. Once the withdrawal is observed, the earlier confirmation no longer authorizes the pending return. The return must remain uncommitted unless it is authorized again under the updated user instruction. Dispatching the return after observing the withdrawal on the basis of the earlier confirmation constitutes a failure.