# CAR-VOICE-BENCH: A BENCHMARK FOR IN-CAR FULL-DUPLEX VOICE ASSISTANTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In-car voice assistants are integral to modern vehicle control, navigation, and telematics. During natural interactions, users routinely exhibit dynamic conversational behaviors, such as concurrent speech, barge-in interruptions, and abrupt goal modifications. Despite their prevalence, existing automotive benchmarks fail to jointly evaluate full-duplex spoken dialogue, executable tool use, and rigorous outcome verification. To address this deficiency, we propose CAR-Voice-Bench, a unified evaluation framework integrating domain-specific tasks, full-duplex interaction, and a replayable testbed. The benchmark comprises 300 curated tasks covering six practical challenges: driving safety, turn management, tool unavailability, disambiguation, simple, and complex. We systematically evaluate three representative models under both standard and dialectal speech conditions with injected background noise, independently testing each configuration three times. Among them, OpenAI's `gpt-realtime-2.1` achieves the best overall performance, attaining a pass@1 reward of 51.61%.

## 1 INTRODUCTION

As speech models become more capable of understanding and interaction, evaluation has expanded from recognition accuracy to semantic understanding, dialogue, and task execution. AISHELL-1 (Bu et al., 2017) provides a Mandarin speech corpus and recognition baseline; Common Voice (Ardila et al., 2019) is a crowdsourced multilingual speech corpus; and FLEURS (Conneau et al., 2022) provides multilingual speech evaluation resources. SUPERB (Yang et al., 2021) evaluates speech representations on diverse downstream tasks, while Dynamic-SUPERB (Huang et al., 2023) evaluates instruction-driven and dynamic multitask speech capabilities. For spoken language understanding, Fluent Speech Commands (Lugosch et al., 2019) maps speech to command intents, SLURP (Bastianelli et al., 2020) covers multidomain assistant requests with intent and entity annotations, and SLUE (Shon et al., 2021) evaluates recognition, named-entity recognition, and sentiment analysis on natural speech.

Audio-language benchmarks further extend evaluation to audio understanding and response quality. AudioBench (Wang et al., 2024) covers speech, audio-scene, and paralinguistic understanding; AIR-Bench (Yang et al., 2024) evaluates generative comprehension over audio; and MMAU (Sakshi et al., 2024) focuses on multi-task audio understanding and reasoning across speech, environmental sounds, and music. VoiceBench (Chen et al., 2024) evaluates voice assistants on knowledge, instruction following, safety, and robustness across speakers, environments, and colloquial expressions. For multi-turn dialogue, MultiWOZ (Budzianowski et al., 2018) and SpokenWOZ (Si et al., 2023) support task-oriented state tracking, cross-turn information integration, and end-to-end task completion.

Tool-use benchmarks evaluate tool selection, argument generation, and multi-step execution. API-Bank (Li et al., 2023) provides executable APIs and tool-use dialogues, while ToolBench, introduced with ToolLLM (Qin et al., 2023), constructs tasks over real-world APIs and evaluates tool-use trajectories. BFCL (Patil et al., 2025) evaluates function selection and argument generation, including parallel calls, cases where tools should not be called, and stateful multi-turn tasks. Extending tool-use evaluation to spoken interaction, $\tau^2$-bench (Barres et al., 2025) evaluates conversational agents in dual-control environments, while $\tau$-Voice (Ray et al., 2026) extends this setting to full-duplex speech with temporally aligned speech and tool events.



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.

In-car voice assistance represents an intrinsically challenging, safety-critical domain where drivers frequently barge in or shift goals mid-turn. While CAR-bench (Kirmayr et al., 2026) applies multi-turn tool interaction to navigation, charging, and vehicle control, and evaluates execution consistency, disambiguation, and capability boundaries, no public benchmark, to the best of our knowledge, jointly evaluates automotive workflows, full-duplex spoken interaction, and verifiable execution outcomes. To bridge this critical gap, we introduce CAR-VOICE-BENCH, marrying the domain fidelity of CAR-bench with the full-duplex paradigm of $\tau$-Voice through localized Chinese in-cabin tasks, systematic data scaling, and joint process-outcome metrics. Localized to Chinese vehicular contexts, CAR-VOICE-BENCH encompasses 300 curated tasks spanning six core challenges: driving safety, turn management, capability boundaries (tool unavailability), pragmatic disambiguation, basic tasks, and complex cross-service workflows, evaluated under noisy standard and dialectal speech conditions. Our closed-loop framework pairs a controllable spoken-user simulator with audio agents in a reproducible vehicle environment, validating interactions via triadic checks: Natural Language (NL) compliance, Database (DB) state transitions, and intermediate Action traces, alongside fine-grained turn-timing diagnostics.

Empirical evaluations across models highlight the necessity of this multi-dimensional protocol. `gpt-realtime-2.1`, `qwen-audio-3.0-realtime-plus`, and `Seeduplex` achieve pass rates of 26.39%, 10.05%, and 10.56% on tool-unavailable tasks; 74.24%, 74.35% and 20.45% on basic tasks; and 73.53%, 46.05%, and 15.29% on complex coordination, respectively. Critically, correct terminal DB states often conceal flawed execution trajectories or conversational non-compliance. Under dialectal speech, overall pass@1 degrades by 14.33, 0.94 and 2.22 percentage points, indicating that under our end-to-end configuration, `gpt-realtime-2.1` suffers a more pronounced degradation under dialectal speech conditions.

## 2 CAR-VOICE-BENCH: DATASET AND EVALUATION

### 2.1 COMPARISON WITH RELATED BENCHMARKS

Table 1 outlines the comparative coverage of CAR-bench, $\tau$-Voice, and CAR-Voice-Bench. Specifically, "dedicated turn tasks" refer to benchmark instances that explicitly operationalize barge-ins and backchannels as distinct evaluation targets. Although the $\tau$-Voice framework supports full-duplex simulated audio timeline, its publicly released benchmark does not encompass evaluation data for automotive voice assistants; conversely, CAR-bench lacks a framework for full-duplex voice interaction.

### 2.2 DATASET CREATION PROCESS

Building upon the operational scenarios, business scopes, and rule designs of CAR-bench, CAR-Voice-Bench constructs Chinese in-cabin voice tasks. While preserving a subset of tasks from CAR-bench, we redesign the action schemas for the vast majority of them. The reformulated tasks explicitly formalize prerequisites and sequential constraints across interrelated operations, enabling

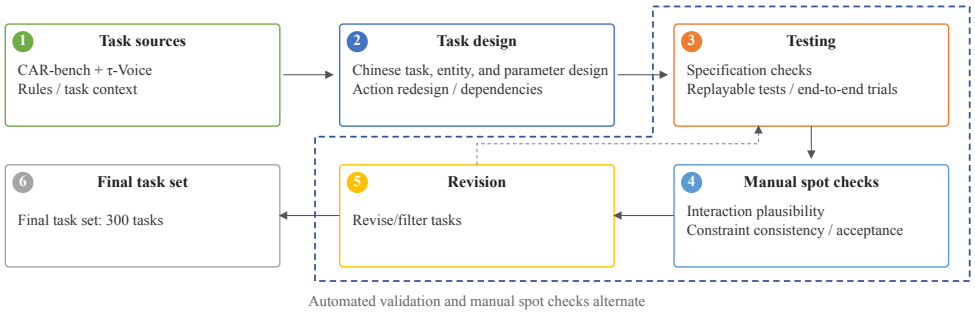
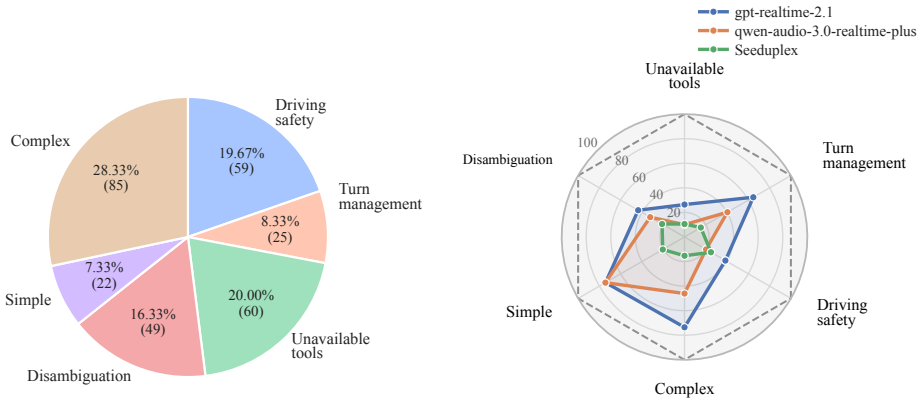



Figure 1: CAR-Voice-Bench data construction and filtering process.

Table 1: Comparison of data coverage across three benchmarks.

| Benchmark | In-car tasks | Multiturn tool use | Full-duplex speech | Dedicated turn tasks | Tasks |
|---|---|---|---|---|---|
| CAR-bench (Kirmayr et al., 2026) | ✓ | ✓ | — | — | 240 |
| $\tau$-Voice (Ray et al., 2026) | — | ✓ | ✓ | — | 278 |
| **CAR-Voice-Bench** | ✓ | ✓ | ✓ | ✓ | **300** |

Note: ✓ denotes targeted benchmark coverage; – denotes lack of dedicated task data




(a) Task category distribution

(b) Performance by category (pass@1, %)

Figure 2: Task distribution and model performance by category. (a) Distribution of the 300 tasks across six categories. (b) Pass@1 scores of the three models by task category, averaged equally over the standard and dialectal speech conditions.

the evaluator not only to inspect the final database state, but also to verify the causal dependencies spanning essential queries, authorization gates, and consequential write actions. We detail this data construction workflow in Figure 1.

Task expansion employs a pipeline combining AI-assisted generation and verification with human spot-checks. First, structured information such as driving states, available capabilities, user goals, and business constraints is used to generate task drafts. Subsequently, through task specification validation, environment replay testing, and multi-turn end-to-end dry runs, the task's executability, constraint consistency, voice interaction plausibility, and scoring determinacy are verified. The final benchmark suite comprises 300 tasks grouped into six categories according to their primary evaluation objectives, with the task distribution and model performance by category shown in Figure 2.

### 2.3 Evaluation Methodology

Task completion is rigorously audited across three dimensions, dynamically configured to match each task's operational requirements:

- **Natural-Language Assertions (NL).** Evaluates dialogue responses against privileged tool logs using an LLM judge, auditing informational coverage, clarification requests, and capability limitation disclaimers. NL scoring is active on 173 tasks. Basic and complex coordination tasks retain NL assertions in their specifications, but NL scoring is disabled for these categories in the reported evaluation; among turn management tasks, only backchannel incorporates NL into task success.

- **Database-State Verification (DB).** Assesses end-state consistency by replaying reference and actual execution traces from equivalent initial checkpoints. DB verification is mandatory for all 300 tasks.
- **Action-Trace Compliance (Action).** Audits procedural fidelity, verifying required tool invocations, argument schema correctness, execution status, and prerequisite conversational dependencies. While 280 tasks undergo standard action matching, 20 interruption tasks utilize a customized action validator to accommodate dynamic goal revisions without bifurcating the evaluation schema.

**Explicit action-dependency checks.** For actions conditioned on explicit prerequisites, the evaluator enforces causal and temporal precedence over the execution trace: the required prior operation must conclude successfully, with its result visible before any dependent call is dispatched.

Reported metrics follow a hierarchical aggregation protocol: scores are computed at the individual task level, macro-averaged over the applicable included tasks within each run, and finally averaged over three evaluation trials. Each metric uses its own eligibility criteria, and macro-averaging assigns equal weight to each applicable task rather than increasing its weight with the number of tool calls.

- **Macro Parameter Accuracy.** For each applicable task, we compute the proportion of parameter-correct calls among eligible calls whose tool names match the task requirements, including accepted information-equivalent read calls. We then average these task-level proportions with equal weight across applicable tasks.
- **Macro Action Coverage.** The proportion of required procedural actions that satisfy equivalence verification.
- **First-Correct-Action Latency.** The duration from the user speech offset to the resolution of the first goal-advancing tool invocation.
- **Mean Logical-Action Latency.** The mean duration from the user speech offset to the execution of each requisite logical operation.
- **All-Action Completion Latency.** The total elapsed duration between the user speech offset and the completion of all required operations.
- **Normalized Extra Interaction Turn.** Quantifies relative conversational turn overhead, computed exclusively over fully successful trajectories (Reward $= 1.00$ with passed Action verification and valid records) where actual turns equal or exceed the minimal reference length. For task $i$, let $A_i$ and $B_i$ denote actual and reference turn counts, respectively. The normalized extra interaction turn is defined as

$$E_i = \frac{A_i - B_i}{B_i} \tag{1}$$

## 3 CAR VOICE AGENT HARNESS

Extending the $\tau$-Voice full-duplex evaluation framework, our testbed integrates a spoken-user simulator, a native audio assistant, a full-duplex orchestrator, and an automotive environment. Audio exchanged within the simulation framework uses 16 kHz, 16-bit mono PCM, with provider-specific resampling handled by the adapters. In addition, we incorporate conversational response-ownership validation, alongside automated session-cleanup routines and comprehensive interaction-evidence logging.

The user simulator formulates responses based on task objectives, dialogue history, and interaction state, rendering them into speech via TTS(text-to-speech). During streaming, task-specified acoustic corruptions—including stationary background noise, transient interference, channel distortions, and packet loss—are superimposed onto the user audio. Operating over this continuous stream, the assistant generates spoken output and dispatches tool calls. The orchestrator drives the simulation over discrete time steps, relaying audio chunks and structured events while delivering tool execution feedback in the subsequent step.

For tasks configured with turn-management scripts, the script orchestrates user vocalization, floor yielding, and session termination: specified actions are executed once their trigger conditions are

met; otherwise, the system remains idle while streaming ongoing utterances to completion. For the formal evaluation, autonomous and stochastic interruption or backchannel decisions are disabled. The 25 turn-management tasks are fully controlled by deterministic scripts and do not fall back to a dynamic turn-taking policy; other tasks follow their predefined interaction schedules.

### 3.1 Vehicle Environment and Tool Feedback

The complete tool set includes 64 tools that span vehicle control, navigation, information retrieval, and communication. The environment maintains static metadata, vehicular, and navigation states, along with domain data such as locations, weather, contacts, and calendar events. Task configurations dictate allowable tools, parameter spaces, and response visibility: tool and argument constraints are enforced through interface pruning coupled with runtime invocation validation, while response restrictions are applied by masking targeted field values with `unknown`.

Domain policies are supplied to the assistant via system instructions. Tool invocations undergo a multi-stage validation pipeline: the system first verifies tool and argument availability against the task configuration, followed by schema conformance covering argument naming, mandatory presence, and typing. Upon passing these entry-level checks, the specific tool function validates underlying domain logic and safety constraints before executing the corresponding action. Unsuccessful calls emit structured error diagnostics back to the assistant and log internal traces where relevant.

Database (DB) state evaluation initializes two parallel environments from the same task configuration to reconstruct states from the model's trace and the reference trajectory, respectively, comparing their final database hashes. During replay, successful non-mutating calls, including read-only calls, are skipped. Failed non-mutating calls are retained when they may have produced diagnostic side effects, and replayed calls are checked against their recorded runtime results. For intent-replacement scenarios, the expected state reflects the newly redirected user intent.

### 3.2 Full-duplex Scheduling and Turn Control

The orchestrator coordinates audio chunks and system events at discrete 200ms intervals across all models, defining simulation scheduling granularity rather than constraining model response latency. Tool calls are executed within the current simulation tick, and their results are delivered to the assistant at the next tick. The system advances the delivery of spoken text in proportion to audio playback progress. For natural language evaluation, it reconstructs a linearized dialogue transcript based on the chronological sequence and temporal containment of audio segments. In parallel, tool invocations and their execution outcomes are compiled into a dedicated structured context, which is supplied alongside the transcript to the evaluator model.

To maintain reproducible evaluation conditions, we disable autonomous and stochastic user interruptions/backchannels, benchmarking these dynamics solely via 25 scripted turn-management tasks (20 for intent replacement and 5 for backchannels). In these scenarios, user turn behaviors are entirely deterministic: the script idles when triggers are unmet and concludes the session upon completion without falling back to heuristic turn rules.

In certain intent-replacement tasks, a language model generates constrained clarifications while the script retains temporal control over speech emission. The controller directly arbitrates assistant audio streams (e.g., continuing, pausing, or terminating playback) without synthesizing extra user turns. Strict information isolation is maintained throughout: The language model used by the user simulator does not directly receive the assistant's tool calls, tool results, or environment database state; the script controller can observe structured tool-call events to evaluate its trigger conditions.

### 3.3 Auxiliary Language Models and Evaluation Components

We adopt `gpt-5.2` to simulate user utterances from task goals and dialogue contexts, and to perform natural language (NL) evaluation by judging conversation transcripts against task assertions and tool traces.

Dialogue turns count only substantive, completed user speech, excluding backchannels. Upon reaching the turn limit, the environment grants the assistant a final turn, terminating only after audio buffering finishes and in-flight tool results resolve; any subsequent user speech is discarded. A

Table 2: Model configurations

| Provider | Model |
|---|---|
| Doubao | Seeduplex (1.2.6.1) |
| OpenAI | gpt-realtime-2.1 |
| Qwen | qwen-audio-3.0-realtime-plus |

run also concludes if the user or script dispatches a stop signal, the simulation exceeds step/error ceilings, or timeouts occur across model inference, audio streaming, or tool execution.

# 4 Experimental Protocol

## 4.1 Experimental Setup

The evaluation uses the three model configurations shown in Table 2. All three models support streaming full-duplex speech input and output. We evaluate them under two acoustic presets—noisy standard speech and noisy dialectal speech—yielding six configurations, each evaluated in three independent runs.

The dialect preset assigns 159 tasks to Sichuan Chinese and 141 tasks to Northeastern Chinese, covering all 300 tasks. For each task, the user persona, behavioral settings, and noise parameters are held constant across the standard and dialectal speech conditions. Background noise files are specified by the task presets. The benchmark therefore compares performance under standard and dialectal speech conditions, rather than evaluating every task in every dialect.

Background and transient noise are drawn from the $\tau$-Voice audio assets. Continuous background noise is mixed using a nominal SNR envelope centered at $15\,\mathrm{dB}$ with smooth variations bounded by $\pm 3\,\mathrm{dB}$. Transient noise is scheduled per audio chunk using a Poisson event probability with a nominal rate of $\lambda = 1\,\mathrm{min}^{-1}$; new events are suppressed while a previous burst is playing. Each burst's nominal SNR is sampled uniformly from $[-5, 10]\,\mathrm{dB}$. Noise amplitudes are scaled relative to a fixed reference speech RMS rather than the RMS of each utterance, so the specified SNRs represent nominal mixing targets. These settings provide controlled acoustic perturbations rather than physically calibrated in-cabin sound fields.

**Execution Reliability and Inclusion Protocol.** In this benchmark, infrastructure errors refer to runs marked as Infra Error by the evaluation framework when execution or evaluation exceptions prevent a valid scored result from being returned. These include API or session errors and session-cleanup failures that prevent further attempts. Such runs are recorded without a Reward score and excluded from performance metrics, whereas task failures with valid scores remain included as Reward=0. Infrastructure errors occurred during the initial evaluation of both `gpt-realtime-2.1` and `qwen-audio-3.0-realtime-plus`, prompting us to retest the affected task-run instances. Retesting resolved the single infrastructure error observed for `gpt-realtime-2.1`. For `qwen-audio-3.0-realtime-plus`, however, 109 task-run instances still lacked valid scores after retesting: 63 instances involving 41 distinct tasks under the standard speech condition, and 46 instances involving 34 distinct tasks under the dialectal speech condition. Because the final aggregate metrics include only tasks with valid scores in all three runs, 259 and 266 tasks were retained for `qwen-audio-3.0-realtime-plus` under the two conditions, respectively.

For a fixed model and speech condition, let $N$ be the number of included tasks, $n = 3$ the number of valid runs per task, and $c_i$ the number of runs of task $i$ with final Reward$= 1.00$. For $1 \le k \le n$, the metrics are defined below, with $\binom{a}{k} = 0$ when $a < k$. For each task, choose $k$ of its $n$ runs without replacement, calculate the fraction of combinations containing at least one success, and average across tasks:

$$\mathrm{pass@}k = \frac{1}{N} \sum_{i=1}^{N} \left[ 1 - \frac{\binom{n-c_i}{k}}{\binom{n}{k}} \right] \tag{2}$$

Table 3: Overall Reward-based pass metrics across three independent evaluation runs.

| Noisy speech, standard | | | | |
|---|---|---|---|---|
| Model | Tasks | pass@1(%) | pass@3(%) | pass$^3$(%) |
| gpt-realtime-2.1 | 300 | **58.78** | **78.33** | **38.67** |
| qwen-audio-3.0-realtime-plus | 259 | 35.78 | 51.74 | 20.85 |
| Seeduplex | 300 | 18.67 | 31.00 | 8.00 |
| **Noisy speech, dialectal** | | | | |
| Model | Tasks | pass@1(%) | pass@3(%) | pass$^3$(%) |
| gpt-realtime-2.1 | 300 | **44.44** | **66.33** | **21.67** |
| qwen-audio-3.0-realtime-plus | 266 | 34.84 | 54.14 | 16.54 |
| Seeduplex | 300 | 16.44 | 29.00 | 7.00 |
| Equal-weight average across speech conditions | | | | |
| Model | Tasks | pass@1 (%) | pass@3 (%) | pass$^3$ (%) |
| gpt-realtime-2.1 | — | **51.61** | **72.33** | **30.17** |
| qwen-audio-3.0-realtime-plus | — | 35.31 | 52.94 | 18.70 |
| Seeduplex | — | 17.56 | 30.00 | 7.50 |

Note: task counts include only tasks with three complete evaluation results that enter the reported statistics. Equal-weight averages assign 50% to each condition's unrounded metric.

Under the same sampling procedure, calculate the fraction of combinations in which all $k$ selected runs succeed, and average across tasks:

$$\text{pass}^k = \frac{1}{N} \sum_{i=1}^{N} \frac{\binom{c_i}{k}}{\binom{n}{k}} \tag{3}$$

The definitions of pass@$k$ and pass$^k$ are given in Equation 2 and Equation 3, respectively. The reported pass@1 is the mean success rate across three independent evaluation runs. For each model and speech condition, we first calculate each task's success rate over its three runs and then average these rates across the included tasks.

### 4.2 Experimental Analysis

Table 3 presents three-run results across models and acoustic conditions across standard and dialectal presets. On each model's valid task set, `gpt-realtime-2.1` achieves the top scores across pass@1, pass@3, and pass$^3$, followed by `qwen-audio-3.0-realtime-plus` and `Seeduplex`. Transitioning from standard to dialectal speech induces pass@1 degradations of 14.33, 0.94 and 2.22 percentage points, respectively.

Table 4 reports model performance across the six task categories, using equal weighting across speech presets.

**Driving-Safety Tasks.** Agents must query safety states and reject dangerous operations. Joint Reward is 38.42% for `gpt-realtime-2.1`, 24.86% for `Seeduplex`, and 20.26% for `qwen-audio-3.0-realtime-plus`. The latter achieves 82.93% on NL but only 32.22% on DB and 25.38% on Action, illustrating that superficial verbal agreement often masks underlying execution failures.

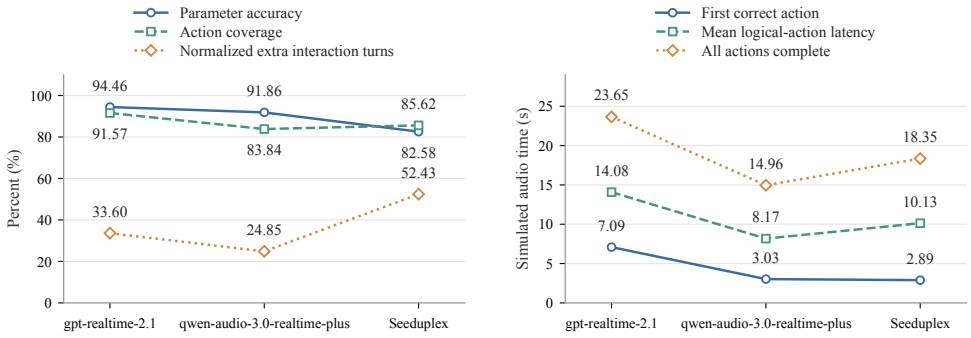
**Turn Management.** In the turn-taking management task, `gpt-realtime-2.1`, `qwen-audio-3.0-realtime-plus`, and `Seeduplex` yield rewards of 64.67%, 40.35%, and 15.33%, respectively. Notably, while `Seeduplex` achieves a DB score of 92.67%, its Action score drops significantly to 15.33%. This disparity demonstrates that reaching the end-state of a replacement intent does not guarantee a compliant full action trajectory.

**Unavailable Tools.** In tasks involving unavailable tools, agents are expected to explicitly provide justifications and reject operations immediately upon detecting that required tools are absent. Across all three models, this category yields universally low rewards (26.39%, 10.05%, and 10.56%, re-

Table 4: Results across six task categories, averaged equally over the two speech conditions.

| Task category | Model | Tasks standard / dialectal | Reward(%) | DB(%) | Action(%) | NL(%) |
|---|---|---|---|---|---|---|
| Driving safety | gpt-realtime-2.1 | 59 / 59 | **38.42** | **61.58** | **52.82** | 61.58 |
| | qwen-audio-3.0-realtime-plus | 59 / 58 | 20.26 | 32.22 | 25.38 | **82.93** |
| | Seeduplex | 59 / 59 | 24.86 | 60.45 | 28.81 | 76.84 |
| Turn management | gpt-realtime-2.1 | 25 / 25 | **64.67** | 91.33 | **76.67** | 16.67 |
| | qwen-audio-3.0-realtime-plus | 23 / 25 | 40.35 | 73.51 | 43.07 | 23.33 |
| | Seeduplex | 25 / 25 | 15.33 | **92.67** | 15.33 | **66.67** |
| Unavailable tools | gpt-realtime-2.1 | 60 / 60 | **26.39** | **72.50** | **50.83** | **43.61** |
| | qwen-audio-3.0-realtime-plus | 31 / 32 | 10.05 | 64.25 | 20.51 | 34.44 |
| | Seeduplex | 60 / 60 | 10.56 | 62.22 | 26.11 | 41.39 |
| Disambiguation | gpt-realtime-2.1 | 49 / 49 | **43.54** | 80.27 | **55.44** | 54.76 |
| | qwen-audio-3.0-realtime-plus | 46 / 47 | 32.26 | **84.20** | 44.43 | **59.52** |
| | Seeduplex | 49 / 49 | 21.09 | 79.25 | 28.23 | 51.36 |
| Simple | gpt-realtime-2.1 | 22 / 22 | 74.24 | 84.85 | 74.24 | — |
| | qwen-audio-3.0-realtime-plus | 22 / 21 | **74.35** | 86.83 | **74.35** | — |
| | Seeduplex | 22 / 22 | 20.45 | **90.91** | 20.45 | — |
| Complex | gpt-realtime-2.1 | 85 / 85 | **73.53** | **89.80** | **73.53** | — |
| | qwen-audio-3.0-realtime-plus | 78 / 83 | 46.05 | 82.31 | 46.53 | — |
| | Seeduplex | 85 / 85 | 15.29 | 69.22 | 15.29 | — |

Note: task counts show the complete three-run task sets for the standard / dialectal conditions; each count is the denominator for Reward in the corresponding run. Metrics are first averaged across three runs within each speech condition, then equally across the two conditions. A dash in NL indicates that NL scoring is disabled.




(a) Tool invocation metrics

(b) Action latency

Figure 3: Tool Invocation Quality and Action Latency. Parameter accuracy, action coverage, and normalized extra interaction turns, averaged equally over the two speech conditions. First-correct-action, mean logical-action, and all-action completion latency, averaged equally over the two speech conditions. Latency is aggregated only over applicable task-run instances with successful action checks and valid timing evidence.

spectively). Notably, DB scores remain relatively high because, in tasks designed to retain the initial environment state, database checks can still pass even if the agent erroneously triggers unavailable tools. The primary bottlenecks lie in the Action and NL metrics, where agents frequently suffer from tool hallucinations—attempting to execute unavailable tools before ultimately informing the user of their incapability.

**Disambiguation.** Across disambiguation tasks, Joint Rewards drop to 43.54%, 32.26%, and 21.09%, despite high DB scores (∼80–84%). This disparity underscores the necessity of enforcing required queries and user confirmations rather than assessing end-states alone.

**Simple vs Complex Tasks.** On basic tasks, `qwen-audio-3.0-realtime-plus` and `gpt-realtime-2.1` tie at ∼74% Reward. However, on complex multi-step coordination, `gpt-realtime-2.1` maintains 73.53%, while `qwen-audio-3.0-realtime-plus` and `Seeduplex` deteriorate sharply to 46.05% and 15.29%, exposing scalability limits in compositional reasoning.

**Tool Invocation Granularity.** In addition to evaluating the models' ability to complete tasks, we examine specific aspects of their tool invocation behavior. Tool metrics are macro-averaged over applicable tasks, speech conditions (50% weight each), and runs. As shown in figure 3(a), `gpt-realtime-2.1` leads in parameter accuracy and action coverage despite slightly higher extra turns, indicating turn brevity does not dictate task success. `Seeduplex` underperforms markedly in parameter accuracy. We observe a persistent tendency where the model serializes non-string tool arguments as raw strings. Figure 3(b) delineates model responsiveness across three action-latency metrics: time-to-first-correct-action, mean logical-action latency, and total action-completion latency. The corresponding latency profiles are $(7.09, 14.08, 23.65)$ s for `gpt-realtime-2.1`, $(3.03, 8.17, 14.96)$ s for `qwen-audio-3.0-realtime-plus`, and $(2.89, 10.13, 18.35)$ s for `Seeduplex`. Notably, while `qwen-audio-3.0-realtime-plus` and `Seeduplex` demonstrate substantially lower latencies in both initiating and dispatching actions, their final Action success rates remain markedly inferior to `gpt-realtime-2.1`.

This divergence reveals a fundamental latency–accuracy trade-off in streaming conversational agents. We hypothesize that the prolonged latency of `gpt-realtime-2.1` does not reflect execution inefficiency, but rather indicates a more deliberate intent-grounding strategy: the model defers tool execution until it establishes sufficient conversational context and resolves potential ambiguities, thereby averting premature, erroneous invocations.

## 5 Conclusion

In this work, we present CAR-Voice-Bench, a holistic benchmark designed to rigorously assess task completion, tool invocation fidelity, and dynamic floor control in full-duplex in-cabin voice interactions. Spanning 300 diverse automotive tasks, our framework tightly couples terminal task outcomes with intermediate interaction trajectories through joint Natural Language (NL), Database (DB), and Action assertions. On each model's retained task sets, `gpt-realtime-2.1` achieves the highest overall pass@1, pass@3, and $\text{pass}^3$ scores. Specifically, it achieves a macro-averaged pass@1 of 51.61%, compared with 35.31% for `qwen-audio-3.0-realtime-plus` and 17.56% for Seeduplex; the corresponding $\text{pass}^3$ scores are 30.17%, 18.70%, and 7.50%, respectively. Notably, while `gpt-realtime-2.1` maintains robust performance scaling from simple to complex multi-step coordination, `qwen-audio-3.0-realtime-plus` suffers substantial performance degradation under high task complexity. Finally, our joint analysis of tool-execution latency and action success rates shows that `gpt-realtime-2.1` achieves a higher overall action success rate while exhibiting longer execution latencies on its eligible task-run instances with successful action checks. We hypothesize that more thorough integration of conversational context and intent confirmation may partly explain this observation.

### AI use statement

Generative AI tools assisted with benchmark task drafting and manuscript editing. The authors take full responsibility for all content, including the paper's text, claims, benchmark artifacts, and reported results.

## REFERENCES


Rosana Ardila, Megan Branson, Kelly Davis, et al. Common Voice: A Massively-Multilingual Speech Corpus, 2019. URL https://arxiv.org/abs/1912.06670. arXiv:1912.06670.

Victor Barres, Honghua Dong, Soham Ray, et al. $\tau$2-Bench: Evaluating Conversational Agents in a Dual-Control Environment, 2025. URL https://arxiv.org/abs/2506.07982. arXiv:2506.07982.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, et al. SLURP: A Spoken Language Understanding Resource Package, 2020. URL https://arxiv.org/abs/2011.13205. arXiv:2011.13205.

Hui Bu, Jiayu Du, Xingyu Na, et al. AISHELL-1: An Open-Source Mandarin Speech Corpus and A Speech Recognition Baseline, 2017. URL https://arxiv.org/abs/1709.05522. arXiv:1709.05522.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, et al. MultiWOZ – A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling, 2018. URL https://arxiv.org/abs/1810.00278. arXiv:1810.00278.

Yiming Chen, Xianghu Yue, Chen Zhang, et al. VoiceBench: Benchmarking LLM-Based Voice Assistants, 2024. URL https://arxiv.org/abs/2410.17196. arXiv:2410.17196.

Alexis Conneau, Min Ma, Simran Khanuja, et al. FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech, 2022. URL https://arxiv.org/abs/2205.12446. arXiv:2205.12446.

Chien-yu Huang, Ke-Han Lu, Shih-Heng Wang, et al. Dynamic-SUPERB: Towards A Dynamic, Collaborative, and Comprehensive Instruction-Tuning Benchmark for Speech, 2023. URL https://arxiv.org/abs/2309.09510. arXiv:2309.09510.

Johannes Kirmayr, Lukas Stappen, and Elisabeth André. CAR-bench: Evaluating the Consistency and Limit-Awareness of LLM Agents under Real-World Uncertainty, 2026. URL https://arxiv.org/abs/2601.22027. arXiv:2601.22027.

Minghao Li, Yingxiu Zhao, Bowen Yu, et al. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs, 2023. URL https://arxiv.org/abs/2304.08244. arXiv:2304.08244.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, et al. Speech Model Pre-training for End-to-End Spoken Language Understanding, 2019. URL https://arxiv.org/abs/1904.03670. arXiv:1904.03670.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, et al. The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Yujia Qin, Shihao Liang, Yining Ye, et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs, 2023. URL https://arxiv.org/abs/2307.16789. arXiv:2307.16789.

Soham Ray, Keshav Dhandhania, Victor Barres, and Karthik Narasimhan. $\tau$-Voice: Benchmarking Full-Duplex Voice Agents on Real-World Domains, 2026. URL https://arxiv.org/abs/2603.13686. arXiv:2603.13686.

S Sakshi, Utkarsh Tyagi, Sonal Kumar, et al. MMAU: A Massive Multi-Task Audio Understanding and Reasoning Benchmark, 2024. URL https://arxiv.org/abs/2410.19168. arXiv:2410.19168.

Suwon Shon, Ankita Pasad, Felix Wu, et al. SLUE: New Benchmark Tasks for Spoken Language Understanding Evaluation on Natural Speech, 2021. URL https://arxiv.org/abs/2111.10367. arXiv:2111.10367.

Shuzheng Si, Wentao Ma, Haoyu Gao, et al. SpokenWOZ: A Large-Scale Speech-Text Benchmark for Spoken Task-Oriented Dialogue Agents, 2023. URL https://arxiv.org/abs/2305.13040. arXiv:2305.13040.

Bin Wang, Xunlong Zou, Geyu Lin, et al. AudioBench: A Universal Benchmark for Audio Large Language Models, 2024. URL https://arxiv.org/abs/2406.16020. arXiv:2406.16020.

Qian Yang, Jin Xu, Wenrui Liu, et al. AIR-Bench: Benchmarking Large Audio-Language Models via Generative Comprehension, 2024. URL https://arxiv.org/abs/2402.07729. arXiv:2402.07729.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, et al. SUPERB: Speech processing Universal PERformance Benchmark, 2021. URL https://arxiv.org/abs/2105.01051. arXiv:2105.01051.