

VOICE AGENTS UNDER ACOUSTIC STRESS: FROM SIGNAL DEGRADATION TO INTERACTION AND ACTION

Amir Ivry¹ Kai-Wei Chang² Lin Zhang³ Sharon Gannot⁴ Carlos Busso⁵

¹Technion – Israel Institute of Technology ²Massachusetts Institute of Technology
³Independent Researcher ⁴Bar-Ilan University ⁵Carnegie Mellon University

ABSTRACT

Voice agents must complete users’ tasks despite noise, reverberation, and competing speech. Evaluating agents’ robustness therefore requires following how acoustic conditions affect the conversation and the actions taken on the user’s behalf. This overview examines what existing benchmarks reveal about agents’ ability to complete tasks under acoustic stress and where further task-based evaluation is required. We then introduce TRACE, a practical workflow for designing, running, and interpreting evaluations of acoustic robustness in task-oriented human-agent interactions: the same agent attempts a specified task with an original recording and an acoustically stressed copy, and the resulting conversations are scored for task completion, wrong actions, recovery, and user effort. Finally, we explain how results from these evaluations can guide changes to an agent to prevent wrong actions and improve recovery.¹

Index Terms— voice agents, acoustic robustness, evaluation.

1. INTRODUCTION

Suppose a voice assistant is asked to *send a package to fifteen Oak Street*. Noise obscures the number, and the assistant hears *send a package to fifty Oak Street*. If it places the delivery order immediately, it sends the package to the wrong address. If it first asks the user to repeat the number, it may obtain the correct address before placing the order, but the user must answer another question (Fig. 2). The outcome therefore depends on how the assistant responds to the noisy request. Assessing acoustic robustness requires following that response through to the task’s outcome: does the assistant complete the task correctly, and how much effort does it require from the user? This overview explains how practitioners can design tests to answer that question for their own voice agents.

Recent benchmarks have begun to address this broader view of robustness. VoiceBench examines how spoken requests and listening conditions affect answer quality [1], while τ -Voice and EVA-Bench follow task completion and conversational behavior in simulated interactions [2, 3]. These broader tests matter because answering accurately does not guarantee that an agent knows when it needs more information. E.g., *Pardon?* finds that models can answer questions successfully yet fail to request clarification when essential information is missing [4] and LALM-as-a-Judge shows that automated detection of harmful spoken content depends heavily on the model used for judging and what its input modalities are [5].

These studies provide complementary robustness tests, each with its own tasks, conditions, and scoring rules. For practitioners, the remaining challenge is to combine their lessons into a reproducible

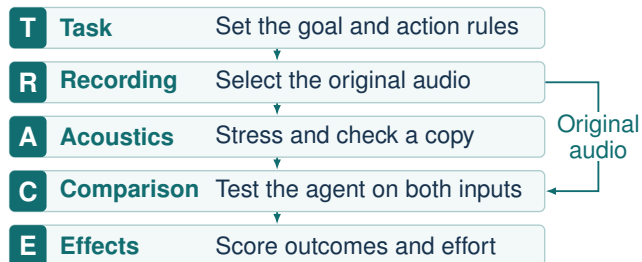


Fig. 1. The TRACE workflow. The original audio is retained while a copy is stressed acoustically. Both versions are tested with the same agent, and are compared by task outcomes, recovery, and user effort.

test of a particular application, with clear expectations for the agent’s behavior and criteria for judging the outcome. We address this need by introducing TRACE (Fig. 1), one possible practical workflow for evaluating acoustic robustness in task-oriented human-agent interactions. Our contributions are:

- **A guide to what needs testing.** We explain how acoustic stress can affect an agent’s decisions and compare which behaviors existing benchmarks test. This framework helps practitioners identify what remains to be evaluated for their application (Sec. 3).
- **TRACE: a workflow for carrying out those tests.** We provide steps for designing, running, and interpreting acoustic robustness tests, and show how to assess task completion, wrong actions, recovery, and user effort (Sec. 4).
- **Guidance for using the results.** We explain how failures found during TRACE can guide changes to the agent and how to check whether those changes help (Sec. 5).

2. HOW ACOUSTIC STRESS AFFECTS AGENT BEHAVIOR

The request to send a package to fifteen Oak Street in Fig. 2 illustrates how a misheard number can lead to a wrong action. Acoustic stress can disrupt three related decisions: what does the agent understand? whom does it follow? and when does it act?

Content: what does the agent understand? Noise and reverberation can obscure the sounds that distinguish one word from another. The agent may then interpret “fifteen” as “fifty” and use the wrong address when placing the order. The problem can extend beyond substituting similar words: WildASR documents transcriptions containing words that were never spoken when recordings are clipped or cut short [6]. An agent relying on such a transcription may therefore act on information the user never supplied.

¹Audio examples and code: <https://amir-ivry.github.io/trace/>

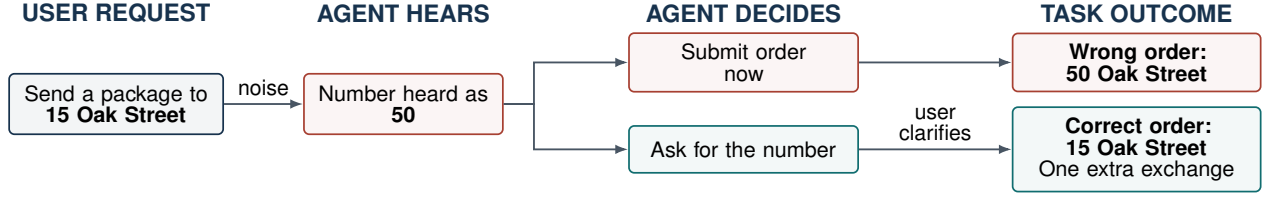


Fig. 2. How one misheard number due to acoustic stress affects the user-agent conversation. In this hypothetical example, the assistant hears *fifty* instead of *fifteen* in the delivery address. If it places the order immediately, the package goes to the wrong address. If it first asks the user to repeat the number and understands the reply, it places the order correctly, but the user must answer an extra question.

Source: whom does the agent follow? Competing speech can cause an agent to attribute a bystander’s words to the user. Lee et al. [7] show that voice assistants can mistake another speaker’s interjection for a correction by the original user. In the package example, a bystander’s “fifty” could then replace the user’s “fifteen,” even if the agent hears both numbers correctly. The resulting order would follow the wrong person’s instruction.

Timing: when does the agent act? Noise or overlapping speech can obscure the start of a user correction, making an unfinished request appear complete. If the user says “fifty—sorry, fifteen,” the agent may place the order before hearing the corrected number. Full-Duplex-Bench-v3 illustrates the consequence of premature action: an early tool call can retain an outdated destination despite a subsequent correction [8]. Understanding the correction later does not prevent the initial wrong action. The key challenge, therefore, is deciding when to act and when to wait for possible corrections before committing to an irreversible tool call.

3. WHAT EXISTING BENCHMARKS TELL US

Acoustic stress can make an agent misunderstand a request, follow the wrong speaker, or act before a correction ends. Tab.1 includes both speech-processing benchmarks, which help diagnose problems in audio quality or recognition, and voice-agent benchmarks, which assess responses and actions. Both are relevant because identifying a speech-processing problem does not establish its consequence for the task. We draw three conclusions from these tests and benchmarks: errors observed under noisy conditions are not necessarily caused by noise, task completion alone does not characterize the entire interaction, and requesting clarification does not necessarily indicate successful task recovery. These conclusions are detailed below.

Task accuracy and sensitivity to noise are different. To estimate whether acoustic stress adds errors, an evaluation must compare matched versions of the same task with and without the stressor. Laskar et al. make this comparison for tool calling by converting written tool-use requests into speech and adding noise [19]. Their results illustrate that tool-calling performance changes only modestly across the tested noise levels. Thus, performance with noisy audio measures task accuracy under that condition, whereas the difference from the matched clean condition estimates sensitivity to noise.

Task completion does not describe the whole interaction. τ -Voice measures task completion alongside how agents handle simulated conversations [2]. Under its realistic test conditions, the agent with the highest overall task completion rate also interrupts the user most often. Completing more tasks therefore does not establish that an agent handles the conversation better: interruptions must be measured separately from the final task outcome.

Clarification and task recovery are different outcomes. Recovery involves both obtaining a correction and using it to complete the

task, but evaluations may examine only part of this process. *Pardon?* tests whether missing essential information prompts a clarification request [4]. Beyond requesting help, Interactive ASR tests whether spoken feedback corrects a transcript [23], while τ -Elicit follows value collection through conversation to the final database submission [24]. For the example in Fig. 2, neither a clarification request nor a corrected transcript establishes that the right order was placed or that no wrong order preceded it.

A robustness test should show whether acoustic stress leads to additional wrong actions, fewer completed tasks, or excess work for the user. This evaluation requires comparing the same agent on the same tasks with and without added stress and checking what it does throughout each conversation, including mistakes it later corrects. TRACE organizes these checks into a practical workflow for testing a voice agent on the tasks it is intended to perform.

4. TRACE: A WORKFLOW FOR TESTING VOICE AGENTS

TRACE tests how acoustic stress affects task completion through five steps: *Task, Recording, Acoustics, Comparison, and Effects*. Fig. 1 summarizes the workflow and Tab.2 develops the delivery example in Fig. 2 into a test with explicit scoring rules.

The example compares the original request for 15 Oak Street with a copy in which “fifteen” is replaced by silence. It tests whether the agent asks for the missing number and uses the reply before placing the order. Unlike adding noise, which may leave the number understandable, this deliberate removal creates a need for clarification.

4.1. T—Task: The goal and rules for completing the goal

A test first defines correct behavior and permitted actions to achieve the user’s goal [25]. In Tab.2, the goal is exactly one active order for 15 Oak Street, using a number provided by the user. Accordingly, the agent may neither guess the removed number, even if correct, nor place a wrong order, even if it is later canceled. Time and user-turn constraints determine when an unfinished attempt ends. Jointly, these rules define success before testing begins.

4.2. R—Recording: The speech that the agent receives

The recording step establishes the original speech input before additional stress is introduced. Recordings are selected to represent the application’s spoken instructions and users, with existing room and microphone effects documented as the starting acoustic conditions [11]. In our example, the recording contains the user’s spoken instruction, “Please send a package to fifteen Oak Street.” The start and end of the word “fifteen” are marked so that it can later be removed

Table 1. What existing benchmarks test, from speech quality to task completion. The final column identifies what their scores do not verify, such as whether a correct tool call actually completes the task.

What is tested	Benchmarks that test it	What remains untested
Audio quality, intelligibility, recognition, or speaker attribution	REVERB; DNS; CHiME-7; NOTSOFAR; AEC; TTS with noisy references [9, 10, 11, 12, 13, 14]	Whether these abilities lead to successful agent tasks.
Answer quality, including use of earlier dialogue information	VoiceBench; VoxEval; Audio MultiChallenge [1, 15, 16]	Whether the agent executes the correct action and reaches the intended task state.
Correct tools and arguments; rejection of side conversations	WearVox; Audio2Tool; Text to Voice [17, 18, 19]	Whether the proposed calls execute successfully and complete the task.
Turn-taking and response timing; timed tool calls	Full-Duplex-Bench series [20, 21, 22, 8]	Whether action timing remains reliable with real service delays and failures.
Clarification requests or transcript correction	<i>Pardon?</i> ; Interactive ASR [4, 23]	Whether the correction leads to successful task recovery.
Task completion or correct final database values	τ -Voice; EVA-Bench; τ -Elicit [2, 3, 24]	Whether results generalize beyond the tested tasks, acoustic conditions, and interactions.

without changing the rest of the recording. The correct address is retained for scoring but is not provided to the agent.

Preserving the original audio and marking the relevant word provides a basis for applying a controlled change to a localized audio segment and testing how it affects the agent’s responses and actions.

4.3. A—Acoustics: The acoustic modification to the original speech and its effect on the available information

The recording in 4.2 is the *reference input*, whereas a stressed copy contains the chosen acoustic modification. In Tab.2, the original samples for “fifteen” are replaced with zeros to simulate a brief, packet-loss-like audio dropout, preserving the request’s duration and the remaining context.

The expected agent behavior depends on what the modified input supports [26]. The target stays unchanged when sufficient information remains (*answer-preserving stress*) [27], whereas missing information requires a permitted way to recover it, such as clarification (*missing evidence*) [4]. A change that alters the answer itself (*changed scene*) requires a revised target; e.g., adding a speaker to a speaker-counting task is scored against the new count, separately from tests requiring an unchanged answer.

The changed audio and all other information available to the agent are examined to determine what information remains available. In the delivery example, clarification is required only if the address’ number cannot be determined from either the remaining recording or the available context.

4.4. C—Comparison: Testing original and stressed recordings

The comparison holds the agent and task fixed while changing the audio [19]. Each delivery run starts with a fresh conversation, using identical model settings. The original recording of the user’s instruction and its stressed copy form a matched input pair.

For each recording of the delivery instruction, the same agent performs the task in two separate runs. Both start with a fresh conversation and no delivery order. One run receives the original audio, including “fifteen” and the other receives a copy with that word replaced by silence. The task and agent settings are otherwise identical.

If the agent asks for clarification, the conversation continues through the user’s reply [28]. The simulated user’s response rule

stays fixed within each pair: cooperative users supply “fifteen” when asked, whereas in abandonment cases the interaction ends without a reply. These replies receive no added stress. The comparison therefore assesses how removing the number changes the agent’s actions, including whether it obtains the missing information through clarification.

4.5. E—Effects: Task completion, recovery, and user effort

TRACE measures how a chosen acoustic condition affects task completion and user effort across multiple test cases. Tab.2 first uses five hypothetical delivery scenarios to explain how individual conversations are scored. It then illustrates how these scores become overall performance measures. For this purpose, the table assumes ten recordings of the delivery instruction, each tested with original audio and with the street number removed. Counting the successful conversations and averaging the extra user turns gives the success rate and user effort in each condition. Comparing these measures illustrates the effect of removing the number.

Task completion and violations. TRACE’s scoring distinguishes reaching the requested result from completing the task without prohibited actions. We use *goal attainment* for reaching that result within the task’s time and turn limits, and *violation* for any prohibited action. *Task success* requires goal attainment without violations.

In scenario 3 in Tab.2, the agent cancels an incorrect delivery order and replaces it with one for the correct address. This scenario attains the goal, but the earlier wrong-address submission prevents task success. Scenario 5 in Tab.2 also fails because guessing the missing street number violates the requirement to obtain it from the user, even when the guess is correct.

Scoring checks both the final task result and the actions taken throughout the conversation. The percentages of conversations that attain the goal, contain a violation, and achieve task success are calculated separately for the original recordings and their stressed copies.

Recovery. A *recovery attempt* begins with clarification by the agent or correction by the user. It completes when the supplied information is used to continue the task. Scenarios 2 and 3 in Tab.2 complete recovery, but scenario 3 retains its violation. Scenario 4 ends without the missing information. The *recovery-completion rate* counts completed attempts among all attempts: six of eight, or 75%, in the

Table 2. A worked TRACE example for delivery to 15 Oak Street. The first part defines the task, speech inputs, and comparison setup. The second part shows how five possible human-agent conversations are scored. The third part illustrates how these scores are summarized across ten independent recordings, each tested with original audio and with the street number removed. All outcomes and counts are hypothetical.

T: Task	One active order for 15 Oak Street within time and turn limits. Prohibited: wrong orders, ordering before obtaining address.
R: Recording	“Please send a package to fifteen Oak Street .” The address’ start and end positions are marked. No other context reveals it.
A: Acoustics	The original audio is retained; “fifteen” is replaced by silence in a copy. Remaining audio and context are checked for clues.
C: Comparison	Both runs start with a fresh conversation using the same agent settings and user-response rules. A user’s willingness to clarify stays fixed within each pair. Replies from both ends receive no added acoustic stress.

Scenario	Input audio	Agent behavior and user response	Goal attained	Violation	Recovery	Task success	Extra user turns
1	Original	Places the correct order without clarification.	Yes	No	No attempt	Yes	0
2	Number removed	Asks for the number; the user provides it; agent places the correct order.	Yes	No	Completed	Yes	1
3	Original	Places a wrong order; the user corrects it; agent cancels and replaces the order correctly.	Yes	Yes	Completed	No	1
4	Number removed	Asks for the number; the user refuses to continue; no order is placed.	No	No	Incomplete	No	1
5	Number removed	Guesses the number correctly and orders without clarification.	Yes	Yes	No attempt	No	0

Metrics across ten paired recordings: ten conversations per condition

	Original	Scenario 1: 9 conversations; scenario 3: 1.	10/10 (100%)	1/10 (10%)	1/1 (100%)	9/10 (90%)	0.1
	Number removed	Scenario 2: 6 conversations; scenario 4: 2; scenario 5: 2.	8/10 (80%)	2/10 (20%)	6/8 (75%)	6/10 (60%)	0.8

table’s stressed condition. Trials without an attempt are excluded. Successful trials are also distinguished by whether recovery occurred. Because agents may encounter different errors, this rate alone cannot rank recovery strategies.

User effort and delay. Effort includes extra user turns and repeated information, including failed recovery [25]. The table counts turns after the initial instruction: scenarios 2-4 each add one, including the refusal to continue. Delay is measured between stated events, such as the end of user speech and the start of the agent’s reply. Premature responses are reported separately so interruptions cannot appear as faster service.

The effect of acoustic stress. Each recording is tested under both conditions with the same task and user-response rules. In Tab.2, success falls from nine of ten conversations (90%) to six (60%): a 30-percentage-point decline. Violations rise from 10% to 20%, and mean extra turns from 0.1 to 0.8. These hypothetical counts illustrate the calculation, not measured performance. Actual comparisons report uncertainty and include timed-out and abandoned trials in the scores.

5. USING TRACE RESULTS TO IMPROVE THE AGENT

TRACE can help improve an agent by showing what goes wrong and whether a change helps. The workflow is applied before and after the change using the same test cases. Two examples illustrate this process for the delivery task in Fig. 2.

Improving how the agent hears the street number. Suppose TRACE reveals that the agent uses the wrong street number even though the number remains audible. One possible change is to add speech enhancement and target speaker extraction (TSE) before the audio reaches the agent. TRACE is then repeated with enhancement and TSE both enabled and disabled, using the same recordings under both original and stressed audio conditions. The resulting task-success rates and response delays show whether this front-end

processing helps the agent complete the task and the delay it brings.

Improving how the agent obtains a missing number. Suppose TRACE instead reveals that asking for the entire instruction again often leaves the number unresolved. The agent is revised to ask directly, “What is the street number?” [28]. Applying TRACE to both clarification policies, with the same tasks, acoustic conditions, and rules for user replies, shows whether the targeted question increases task success or reduces additional user turns. Recovery is assessed through the agent’s use of the answer to complete the delivery task.

For either change, TRACE is also applied to cases that were not used to develop it. This tests whether the improvement extends to new inputs and whether tasks completed correctly before still succeed.

6. LIMITATIONS

TRACE’s guidance depends on how well the selected robustness tests represent the intended application. For example, testing noise only in the initial instruction may miss failures that occur when clarification replies are also noisy. Improvements identified through TRACE therefore support conclusions about the tested tasks and conditions. Their usefulness in other settings requires further assessment.

7. CONCLUSION

TRACE provides one practical workflow for evaluating acoustic robustness in task-oriented human-agent interactions through what an agent accomplishes, the errors it makes, and the effort required from its user. The next step is to test TRACE empirically across different voice agents, tasks, and acoustic conditions. These studies should compare TRACE with existing evaluation practices to determine whether it reveals additional failures and helps practitioners improve task success, while measuring any change in user effort.

8. ACKNOWLEDGMENTS

Funding: No funding was received for conducting this study. The authors have no relevant financial or nonfinancial interests to disclose.
Ethics: No human-subject data were collected as part of this study.
Use of AI: ChatGPT (OpenAI) was used for language editing. All AI-assisted outputs were reviewed and validated by the authors, who take full responsibility for the paper’s content.

9. REFERENCES

- [1] Y. Chen, X. Yue, C. Zhang, X. Gao, R. T. Tan, and H. Li, “VoiceBench: Benchmarking LLM-based voice assistants,” *Trans. Assoc. Comput. Linguistics*, vol. 14, pp. 378–398, 2026.
- [2] S. Ray, K. Dhandhanian, V. Barres, and K. Narasimhan, “ τ -Voice: Benchmarking full-duplex voice agents on real-world domains,” in *Proc. ICML*, 2026.
- [3] T. Bogavelli et al., “EVA-Bench: A new end-to-end framework for evaluating voice agents,” arXiv:2605.13841v3 [preprint], 2026, Accepted to Findings of EMNLP 2026.
- [4] S. Huang et al., “Pardon? evaluating conversational repair in large audio-language models,” in *Findings of ACL*, 2026, pp. 19528–19541.
- [5] A. Ivry and S. Watanabe, “LALM-as-a-judge: Benchmarking large audio-language models for safety evaluation in multi-turn spoken dialogues,” in *Proc. ICML*, 2026.
- [6] G. Tay et al., “Back to basics: Revisiting ASR in the age of voice agents,” in *Proc. Conference on Language Modeling (COLM)*, 2026.
- [7] D. Lee, E. Song, C. H. Lee, H. Kim, and S. Yoon, “Still between us? evaluating and improving voice assistant robustness to third-party interruptions,” in *Proc. 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026, pp. 40975–41004.
- [8] G.-T. Lin, C. Chen, Z. Chen, and H. y. Lee, “Full-Duplex-Bench-v3: Benchmarking tool use for full-duplex voice agents under real-world disfluency,” arXiv:2604.04847v1 [preprint], 2026.
- [9] K. Kinoshita et al., “The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech,” in *Proc. IEEE WASPAA*, 2013, pp. 1–4.
- [10] C. K. A. Reddy et al., “The INTERSPEECH 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results,” in *Proc. Interspeech*, 2020, pp. 2492–2496.
- [11] S. Cornell et al., “The CHiME-7 DASR challenge: Distant meeting transcription with multiple devices in diverse scenarios,” in *Proc. 7th International Workshop on Speech Processing in Everyday Environments (CHiME 2023)*, 2023, pp. 1–6.
- [12] A. Vinnikov et al., “NOTSOFAR-1 challenge: New datasets, baseline, and tasks for distant meeting transcription,” in *Proc. Interspeech*, 2024, pp. 5003–5007.
- [13] R. Cutler et al., “ICASSP 2023 acoustic echo cancellation challenge,” *IEEE Open J. Signal Process.*, vol. 5, pp. 675–685, 2024.
- [14] X. Wang et al., “An investigation of noise robustness for flow-matching-based zero-shot TTS,” in *Proc. Interspeech*, 2024, pp. 687–691.
- [15] W. Cui, X. Jiao, Z. Meng, and I. King, “VoxEval: Benchmarking the knowledge understanding capabilities of end-to-end spoken language models,” in *Proc. ACL*, 2025, pp. 16735–16753.
- [16] A. Gosai et al., “Audio MultiChallenge: A multi-turn evaluation of spoken dialogue systems on natural human interaction,” in *Proc. ACL*, 2026, pp. 35740–35770.
- [17] Z. Lin et al., “WearVox: An egocentric multichannel voice assistant benchmark for wearables,” in *Proc. ICLR*, 2026.
- [18] R. Pahwa et al., “Audio2Tool: Speak, call, act—a dataset for benchmarking speech tool use,” arXiv:2604.22821v2 [preprint], 2026.
- [19] M. T. R. Laskar, X.-Y. Fu, S. S. Sarfjoo, Q. McNamara, J. Robertson, and Shashi Bhushan TN, “From text to voice: A reproducible and verifiable framework for evaluating tool calling LLM agents,” arXiv:2605.15104v2 [preprint], 2026.
- [20] G.-T. Lin et al., “Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities,” in *Proc. IEEE ASRU*. IEEE, 2025, pp. 1–8.
- [21] G.-T. Lin et al., “Full-duplex-bench v1.5: Evaluating overlap handling for full-duplex speech models,” in *Proc. ICASSP*. IEEE, 2026, pp. 19447–19451.
- [22] G.-T. Lin et al., “Full-duplex-bench v2: A multi-turn evaluation framework for duplex dialogue systems with an automated examiner,” in *Proc. 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2026, pp. 27–36.
- [23] P. Wang et al., “Interactive ASR: Towards human-like interaction and semantic coherence evaluation for agentic speech recognition,” arXiv:2604.09121v3 [preprint], 2026.
- [24] Soham Ray and Victor Barres, “ τ -elicitation: Benchmarking multi-turn entity extraction in voice agents,” arXiv:2609.13602 [preprint], 2026.
- [25] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, “PARADISE: A framework for evaluating spoken dialogue agents,” in *Proc. ACL/EACL*, 1997, pp. 271–280.
- [26] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, “Beyond accuracy: Behavioral testing of NLP models with CheckList,” in *Proc. ACL*, 2020, pp. 4902–4912.
- [27] A. Ivry, “Task-aware answer preservation under audio compression for large audio language models,” arXiv:2605.06631 [preprint], 2026.
- [28] D. Bohus and A. I. Rudnicky, “Sorry and I didn’t catch that!—an investigation of non-understanding errors and recovery strategies,” in *Proc. SIGDIAL*, 2005, pp. 128–143.