PROACTIVEAUDIOBENCH: EVALUATING CONDITION-DRIVEN RESPONSES IN AUDIO STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous audio assistance requires models to execute a listening instruction as audio arrives, deciding when a target occurrence warrants a notification and when to remain silent. We introduce **ProactiveAudioBench**, covering sound events, speech, and music through Detection, Tracking, Counting, and Matching. These tasks organize first, repeated, and Nth-occurrence notification rules alongside targets specified by acoustic references. The benchmark contains 1,999 listening requests and 1,851 reference response opportunities, with negative requests requiring silence. Our evaluation framework supports simulated and native streaming, representing each request as a sequence of expected and emitted responses. Joint recall, joint precision, and negative-request false alarms assess the timing and content of notifications together with omissions and unnecessary replies. Across 23 system configurations and four tasks, we find protocol-dependent differences in error timing: simulated-streaming errors cluster before or near reference checkpoints, whereas native-streaming errors show a longer tail of delayed responses.

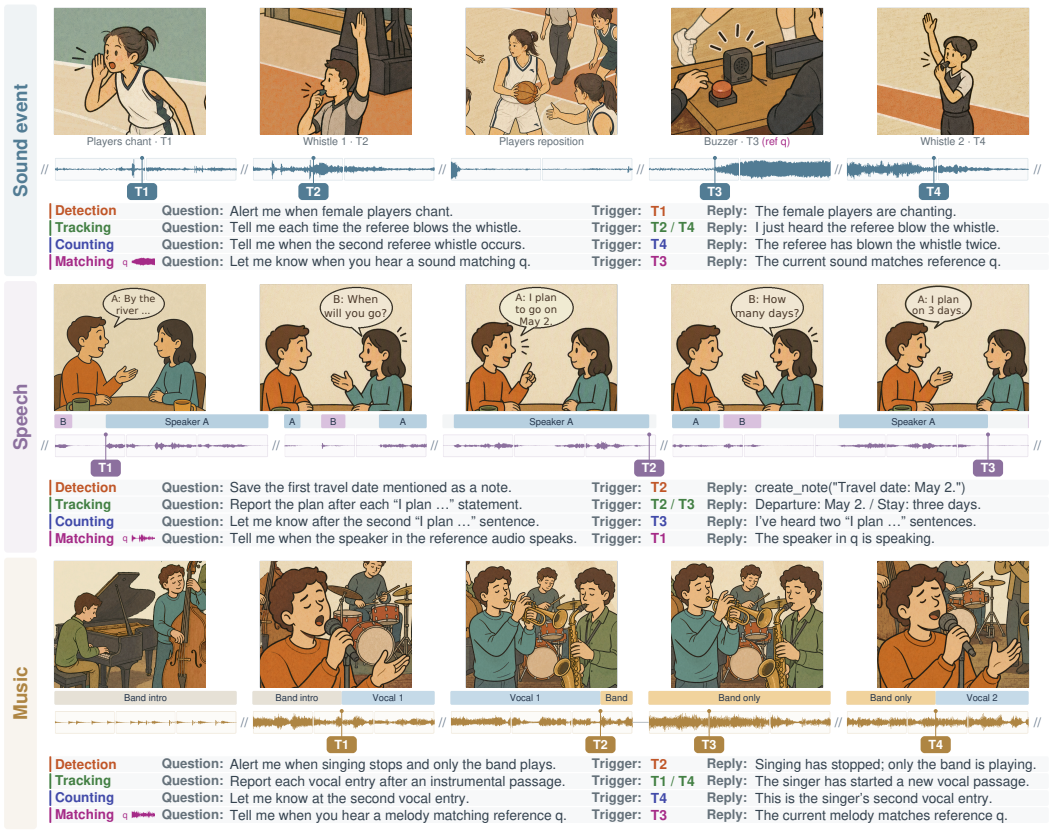


Figure 1: Overview of ProactiveAudioBench.

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.

1 INTRODUCTION

Multimodal large language models are expanding from understanding fixed inputs to continuous perception and real-time interaction. Existing audio benchmarks such as MMAU (Sakshi et al., 2024) and MMAR (Ma et al., 2025b) evaluate understanding and reasoning across speech, sound, and music, while recent benchmarks have begun to study proactive interaction. ProactiveVideoQA (Wang et al., 2025b), OmniPro (Zhao et al., 2026), OmniInteract (Lu et al., 2026), and StreamArena (Zhang et al., 2026) evaluate different forms of proactive behavior in video or audiovisual streams. In audio, ProVoice-Bench (Xu et al., 2026) evaluates proactive intervention across speech and environmental sound, while Proactive-Sound-Bench (Xie et al., 2026) evaluates responses and silence under prescribed intervention policies. These benchmarks establish the importance of deciding not only *what* is occurring in a stream, but also *whether and when* a model should respond.

However, existing audio benchmarks provide limited evaluation of user-specified listening rules: the same target may require notification at its first occurrence, every occurrence, or only a particular occurrence. They also largely specify targets through text, although a particular sound, speaker, or melody may be difficult to describe precisely. We introduce **ProactiveAudioBench**, which evaluates proactive listening across **sound events, speech, and music** along two dimensions: what to listen for and which occurrences should trigger a notification. **Detection**, **Tracking**, and **Counting** require notifications at the first, every, and specified Nth qualifying occurrence, respectively. **Matching** specifies the listening target through reference audio and follows the notification rule in the request. Figure 1 illustrates these tasks.

We construct the benchmark using audio annotation and controlled signal insertion, mapping target occurrences, count conditions, and reference-matching criteria to expected notification sequences. Automated checks and human verification are used to filter and revise the resulting samples. ProactiveAudioBench contains 1,999 listening requests and 1,851 reference response opportunities. Our evaluation supports both simulated and native streaming and jointly measures notification timing, response content, omissions, and false responses. Our main contributions can be summarized as:

- **A benchmark for proactive listening.** We introduce ProactiveAudioBench, spanning sound events, speech, and music through four tasks (Detection, Tracking, Counting, and Matching) that evaluate occurrence-based notification rules and acoustic-reference targets.
- **A construction and evaluation framework for condition-driven responses.** We build 1,999 listening requests with annotated response opportunities and negative requests requiring silence, and evaluate them in simulated and native streaming with recall, precision, and false alarms.
- **Systematic evidence of failure modes in continuous listening.** Across 23 system configurations, we find that Counting and Matching remain particularly difficult, and that simulated and native streaming exhibit distinct timing error patterns.

2 RELATED WORK

2.1 AUDIO UNDERSTANDING

Advances in large language models have broadened audio understanding from task-specific recognition to general understanding and reasoning. Recent audio-language models, such as Pengi (Deshmukh et al., 2023), SALMONN (Tang et al., 2024), and Qwen-Audio (Chu et al., 2023), among others, support diverse audio understanding tasks through joint audio-language modeling, while multimodal systems (Cui et al., 2026; Li et al., 2025; Abouelenin et al., 2025) further integrate audio, vision, and text within unified frameworks.

Audio benchmarks evaluate increasingly diverse content and reasoning requirements. MMAU and MMAR assess understanding and reasoning across sound, speech, and music (Sakshi et al., 2024; Ma et al., 2025b); CMI-Bench and BASS examine musical attributes, structure, and semantics in greater detail (Ma et al., 2025a; Jang et al., 2026); and STAR-Bench assesses spatio-temporal perception and reasoning (Liu et al., 2026).

Understanding a specified target also requires localizing its occurrences. TAG-Bench evaluates whether models can return all temporal intervals matching a natural-language query (Dai et al., 2026), while TSDNet and DASM study target sound detection conditioned on reference audio (Yang

Table 1: Comparison of ProactiveAudioBench with existing proactive interaction benchmarks.

Benchmark	Domains			Proactive Listening Tasks			
	Sound	Speech	Music	Detection First	Tracking Every	Counting Nth-only	Matching Acoustic ref.
ProVoice-Bench (Xu et al., 2026)	✓	✓	✗	◐	✗	✗	✗
Proactive-Sound-Bench (AIM) (Xie et al., 2026)	✓	✗	◐	◐	◐	✗	✗
OmniPro (Zhao et al., 2026)	✓	✓	◐	✓	✓	◐	✗
OmniInteract (Lu et al., 2026)	◐	✓	✗	✓	◐	✗	✗
ProactiveAudioBench (ours)	✓	✓	✓	✓	✓	✓	✓

✓ explicit coverage; ◐ related or partial coverage; ✗ not evaluated. OmniPro and OmniInteract are audiovisual benchmarks; OmniPro is compared in its online setting.

et al., 2021; Cai et al., 2025). These studies connect target identification with temporal localization, providing a foundation for target-specific listening.

2.2 Streaming and Proactive Interaction

Streaming interaction studies how to organize responses as input continues to arrive. VideoLLM-online and MMDuet2 combine continuous observation, interaction history, and autonomous responses (Chen et al., 2024; Wang et al., 2025a). Full-duplex spoken-interaction benchmarks examine pauses, backchannels, turn taking, and interruptions, as well as instruction and persona constraints on speaking and listening behavior (Lin et al., 2025; Mathur & Manocha, 2026).

Proactive benchmarks evaluate dynamic answer updates, event and state monitoring, counting, and procedural guidance in video and audiovisual settings (Wang et al., 2025b; Zhao et al., 2026; Lu et al., 2026), with StreamArena extending evaluation to monitoring over long streams (Zhang et al., 2026). They associate response content with timing through answer intervals, annotated trigger points, or interaction windows. ProactiveVideoQA also evaluates non-streaming models by progressively supplying input chunks together with their previous replies (Wang et al., 2025b).

Audio-focused benchmarks more directly address listening and intervention conditions. ProVoice-Bench evaluates proactive intervention through intent capture, topic monitoring, fact checking, and environmental sound sensing, including user-defined triggers (Xu et al., 2026). Proactive-Sound-Bench, introduced in Audio Interaction Model (Xie et al., 2026), evaluates single and repeated interventions under prescribed acoustic-event policies. These benchmarks primarily organize tasks around intervention scenarios or event policies, with limited systematic coverage of user-specified first-occurrence, every-occurrence, and Nth-only notification rules and targets specified by acoustic references. To address this gap, ProactiveAudioBench organizes four listening tasks across sound, speech, and music, evaluating sustained execution of listening instructions under different targets and requirements.

3 ProactiveAudioBench

3.1 Problem Formulation

Given an audio stream x of duration T, a listening condition c, and an optional acoustic reference q, the model must decide whether to respond, when to respond, and what to say as audio arrives. The condition specifies the target and notification rule; $q = \varnothing$ when no reference is needed. At audio time t, decisions may use only c, q, the received audio prefix $x_{\le t}$, and available context from prior interactions, without access to future audio. The output is an ordered sequence of response events,

$$\widehat{\mathcal{Y}} = \big((\hat{t}_k, \hat{r}_k)\big)_{k=1}^{\hat{M}},$$

where $\hat{t}_k$ is the response position on the main audio timeline, $\hat{r}_k$ is its content, and $\hat{M}$ is the number of emitted responses. Each request is annotated with expected response times $G = (\tau_m)_{m=1}^{M}$ and corresponding reference replies. G specifies the opportunities requiring notification under the full listening condition, rather than the positions of all target events in the audio. These annotations are

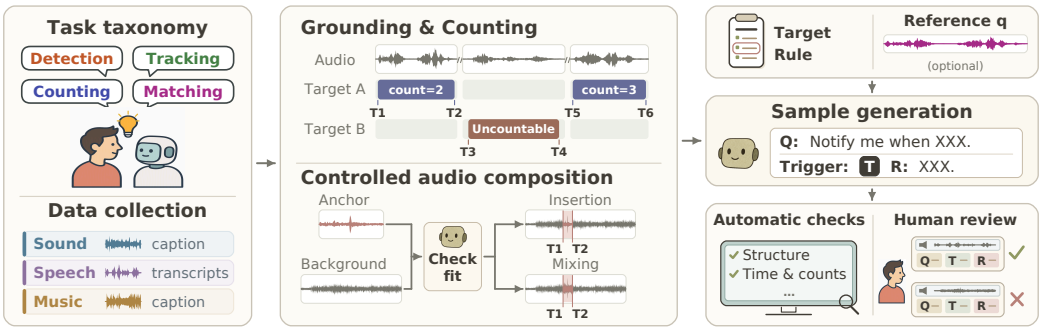


Figure 2: Data construction pipeline of ProactiveAudioBench, covering audio collection, annotation and composition, sample generation, and quality control.

used only for evaluation and withheld from the model; $G = \varnothing$ requires silence. The model must follow the notification rule, responding when required and remaining silent otherwise.

3.2 Task Design

ProactiveAudioBench covers sound events, speech, and music. Sound-event targets include alarms, impacts, doors, devices, traffic, natural sounds, and human and animal activity such as footsteps. Speech targets include words and phrases, topics, intents, tool-use requests, speaker identity, age, and pronunciation. Music targets include instruments, vocals, performance techniques, key changes, structure, and reference segments. Appendix A reports request counts by task and content group. Building on the examples in Figure 1, we define the notification rules and reference-based target specification of the four tasks below.

Detection. Notify once when the target first satisfies the listening condition, then remain silent for the rest of the request. The model must identify the first response opportunity and avoid repeated notifications after fulfilling the request.

Tracking. Monitor the target continuously and issue further notifications as qualifying occurrences recur. The model must identify subsequent response opportunities and distinguish a new occurrence from the continuation of an existing event, avoiding omissions and duplicate replies.

Counting. Accumulate distinct target occurrences and notify once at the Nth occurrence, remaining silent before and after that notification. This task assesses occurrence counting and notification at the specified occurrence index.

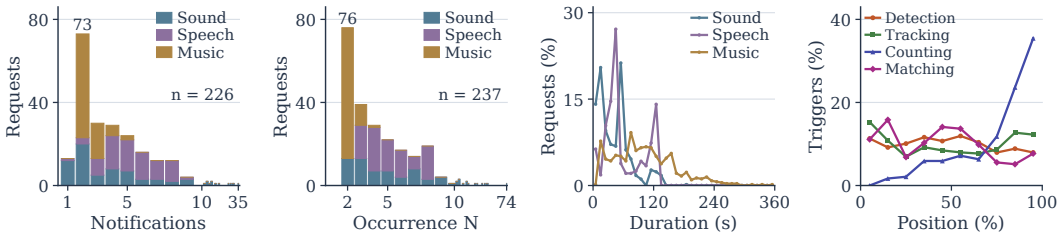
Matching. Use reference audio to identify the requested sound, speaker, or music in the main stream. Notify once when the target satisfies the listening condition, then remain silent. The matching criterion depends on the target, such as the same sound, speaker identity, or music segment.

Response and silence conditions. Each task includes cases requiring a response and cases requiring silence. Positive requests have one or more opportunities satisfying the full listening condition. Negative requests cover target absence, confusable distractors, insufficient occurrences, and reference mismatch. These cases assess both responding when the condition is met and suppressing unnecessary notifications when it is not.

3.3 Data Construction

To construct evaluation data for these continuous listening tasks, we develop the four-stage pipeline in Figure 2, transforming audio and supporting materials from the three domains into samples with listening conditions, reference replies, and annotated response times.

Task taxonomy and data collection. Guided by listening targets and notification rules across sound, speech, and music, we collect audio with detailed descriptions, transcripts, and existing annotations, and select materials suitable for the corresponding tasks.



(a) Tracking (b) Counting (c) Audio duration (d) Trigger positions

Figure 3: Dataset statistics of ProactiveAudioBench. Notification counts and reference response positions use checkpoint-level annotations.

Audio construction and temporal annotation. We use two parallel paths.

(1) Grounding & Counting. This path identifies target events and their timing from audio and supporting materials. Speech is temporally aligned using Qwen3-ForcedAligner-0.6B (Shi et al., 2026). Sound and music events are localized using multimodal models and existing annotations, with frame-level similarities from PE-A-Frame (Vyas et al., 2025) refining boundaries in applicable subsets. We then select countable targets and extract their occurrence sequences: sound-event counting combines candidate detection with multimodal verification, while speech and music counting use transcripts and detailed annotations, respectively.

(2) Controlled audio composition. When audio satisfying a listening condition is difficult to obtain directly, we first use a model to assess background–target compatibility, then insert, splice, or mix selected or generated segments into the audio. The process records the actual positions, spans, and occurrence counts. Both paths produce audio and event records for subsequent sample generation.

Proactive sample generation. For each target and notification rule, we turn event records into a listening condition, reference reply, and expected response opportunities. Matching references are selected from the main audio or other audio sources. Cropping preserves necessary preceding context and keeps occurrence indices consistent with the notification rule.

Automatic checks and human review. Candidates undergo structural, temporal, and count checks, with task-specific acoustic or semantic screening. Human reviewers assess audio quality, the validity of listening requests and reference replies, and the accuracy of trigger annotations, accepting, revising, or rejecting samples. Batch-level spot-checks provide additional quality control (Appendix A).

3.4 Dataset Statistics

ProactiveAudioBench contains 1,999 requests (1,024 positive and 975 negative) across sound, speech, and music, using 1,583 distinct main-audio files totaling approximately 26.1 hours. Figure 3 summarizes the distributions. Request-level audio duration ranges from 2.1 to 357 seconds (median: 52.5). Positive Tracking requests require a median of 3 notifications, and positive Counting requests specify a median occurrence index of 4. Counting opportunities tend to occur late, with 59.1% in the final 20% of the audio. Selection and further statistical details are provided in Appendix A.

4 Evaluation Framework

4.1 Evaluation Modes

Both evaluation modes follow the causal constraint in Section 3.1: models can access only audio received so far. For a recording of duration T, evaluation uses a predefined source-audio observation endpoint H, where $0 < H \leq T$. Monitoring continues to this endpoint after the first reply, recording responses to subsequent opportunities and any additional outputs.

Simulated Streaming. Audio is provided in one-second increments at checkpoints $0 = t_0 < t_1 < \cdots < t_L = H$. At checkpoint t_i, earlier audio and the model's actual replies form the conversation history; the newly arrived segment $x_{(t_{i-1},t_i]}$ is then appended, and the model decides whether to

respond. The kth reply is recorded as

$$\hat{y}_k = (t_{i_k}, \hat{r}_k), \qquad k = 1, \ldots, \hat{M},$$

where t_{i_k} is the audio input endpoint for the corresponding decision, $\hat{r}_k$ is the reply content, and $\hat{M}$ is the number of emitted replies.

Native Streaming. Audio is delivered chronologically within a persistent real-time session, and model replies are collected throughout the evaluation. Independent replies are assembled using response identifiers or end-of-response signals when available. For interfaces with continuous outputs, response events are reconstructed from saved output audio and transcripts using fixed rules (Appendix A). The kth reply is recorded as

$$\hat{y}_k = (\hat{t}_k, \hat{r}_k), \qquad k = 1, \ldots, \hat{M},$$

where $\hat{t}_k$ is the main-stream audio progress when the first fragment of the reply body is received and $\hat{r}_k$ is the received reply content. Response completion status is also retained.

Both modes order replies by time to form the response event sequence $\widehat{\mathcal{Y}}$. Evaluation is defined over the respective observation intervals; $H = T$ denotes observation of the complete recording. Post-input output collection and timestamp handling are described in Appendix A.

4.2 Evaluation Metrics

For Tracking, the simulated annotations merge source occurrences assigned to the same one-second notification interval; native evaluation retains individually recoverable occurrences within these intervals. In each mode, G_j denotes the resulting reference opportunity set.

For request j, let $M_j = |G_j|$ and $\hat{M}_j$ denote the expected notification count and the emitted reply count, respectively. We retain every nonempty independent reply emitted during monitoring, including acknowledgments, unrelated answers, transcriptions, duplicate replies, and incomplete replies. Signals used solely to indicate continued listening do not count as replies.

Temporal and Content Matching. Both evaluation modes use the same content criteria and one-to-one matching procedure, with temporal eligibility determined by how responses are recorded. In simulated streaming, a predicted reply is temporally eligible if it falls within the permitted checkpoint range around the checkpoint associated with the reference opportunity. In native streaming, a reply is temporally eligible if its position on the main audio timeline falls within the permitted early and late offsets from the reference response time. An LLM judge evaluates action fulfillment using the listening condition, reference reply, and grounded event context. A content match requires correctness and relevance scores of at least 4/5 and no material factual contradiction supported by the annotations. Reference replies specify the intended meaning rather than mandatory wording. For Counting, reported counts or ordinals must be correct, with semantically equivalent expressions treated alike. Tool-selection tasks require the correct tool name and an intent consistent with the request and its preceding context; no tool is executed.

Among candidates satisfying both temporal and content requirements, we select an order-preserving one-to-one matching with the maximum number of matches, breaking ties by minimizing the sum of absolute temporal offsets. Each predicted reply can match at most one reference opportunity, and each opportunity can match at most one reply. Let TP_j be the number of valid matches. Then

$$\mathrm{FN}_j = M_j - \mathrm{TP}_j, \qquad \mathrm{FP}_j = \hat{M}_j - \mathrm{TP}_j.$$

This rule prevents duplicate replies from receiving multiple credits for the same opportunity; predictions without a valid match remain in the total reply count.

The following metrics are first computed for each combination of listening task and audio domain. Let $\mathcal{P}$, $\mathcal{N}$, and $\mathcal{D} = \mathcal{P} \cup \mathcal{N}$ denote the sets of positive requests, negative requests, and all requests in that combination, respectively.

Joint Recall. We compute the fraction of reference opportunities validly covered within each positive request, then average across requests:

$$R = \frac{1}{|\mathcal{P}|} \sum_{j \in \mathcal{P}} \frac{\mathrm{TP}_j}{M_j}.$$

This metric measures timely and correct coverage for an average listening request. Positive requests that complete the prescribed observation interval without a reply contribute zero. Each positive request receives equal weight, preventing requests with many events from dominating the result.

Joint Precision. We aggregate valid matches and emitted replies across all positive and negative requests:

$$P = \frac{\sum_{j \in \mathcal{D}} \mathrm{TP}_j}{\sum_{j \in \mathcal{D}} \hat{M}_j}.$$

This metric measures the fraction of emitted replies that correctly correspond to a reference opportunity within the temporal tolerance. Content errors, unnecessary speech, mistimed responses, and duplicates reduce precision. When a task–domain combination contains valid evaluation requests but no emitted replies, precision is defined as zero by convention.

False Alarm Rate (FAR). For negative requests, which require silence throughout observation, we report the fraction with at least one reply:

$$\mathrm{FAR} = \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} \mathbf{1}[\hat{M}_j > 0].$$

FAR measures failure to remain silent on negative requests, including incorrect alerts and other unnecessary speech. One or multiple replies on the same negative request count as one false-alarm request; their number also affects joint precision. Negative requests have no valid response opportunities, so FAR is independent of the temporal matching tolerance.

Aggregation. Each metric is computed for every task–domain combination, averaged equally over the three audio domains within each task, and then averaged equally over the four listening tasks:

$$S_a = \frac{1}{3} \sum_{b=1}^{3} S_{a,b}, \qquad S_{\mathrm{overall}} = \frac{1}{4} \sum_{a=1}^{4} S_a,$$

where a indexes the four listening tasks, b indexes the three audio domains, and $S_{a,b}$ is the metric value for the corresponding combination.

5 Experiments

We evaluate four tasks using joint recall, joint precision, and false alarm rates, then analyze timing distributions of correct and incorrect outputs in both evaluation modes.

5.1 Experimental Setup

Models. We evaluate 23 system configurations (17 simulated and 6 native). Table 2 shows 13, including MiniCPM-o 4.5 in both modes; all 23 appear in Appendix B.

Observation and scoring. Simulated streaming uses one-second increments with prior audio and actual replies. Positive requests generally end three steps after the last annotated trigger, capped at recording end; negatives use the configured endpoint. Native traces cover full recordings. Matching uses ± 1 checkpoint (simulated) or $[-1, +3]$ seconds of audio progress (native). The content judge, `gpt-5.6-luna` at temperature zero, uses listening conditions, reference replies, and event evidence. Both modes use order-preserving one-to-one matching; coverage details appear in Appendix B.

5.2 Main Results

Task rankings differ by mode. Tracking leads recall for eight of nine simulated configurations; all four native configurations peak on Detection, with only 0.8%–5.2% on Counting and 3.6%–8.4% on Matching (Table 2).

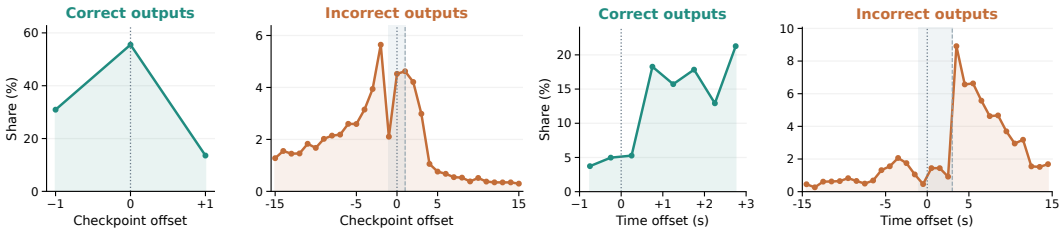
Recall alone obscures notification reliability. Baichuan-Omni-1.5 achieves 59.3% recall but 1.9% precision and 99.4% FAR. Qwen3.5-Omni-Plus and Gemini-3.1-Pro have similar recall (45.5% versus 46.8%), yet markedly different precision (45.9% versus 26.4%) and FAR (24.6% versus 66.4%), motivating joint assessment of all three metrics.

Table 2: Performance on proactive tasks (%). R and P require both temporal and content correctness; FAR is the response rate on negative requests. Temporal matching follows the panel-specific rule.

		Detection			Tracking			Counting			Matching			Avg.		
Model	n	R↑	FAR↓	P↑	R↑	FAR↓	P↑	R↑	FAR↓	P↑	R↑	FAR↓	P↑	R↑	FAR↓	P↑
A. Simulated Streaming *(± 1 checkpoint)*																
Open-source models																
Baichuan-Omni-1.5	1999	**58.3**	98.0	1.4	71.2	99.5	4.5	**59.4**	100.0	0.9	**48.3**	100.0	0.7	**59.3**	99.4	1.9
Qwen3-Omni-30B-A3B	1393	24.6	87.5	1.5	41.7	94.5	3.2	18.7	94.7	0.8	34.0	97.9	1.2	29.7	93.6	1.7
Kimi-Audio-7B	1999	25.9	52.9	1.9	42.9	65.8	8.3	31.0	62.0	0.8	18.3	57.8	0.8	29.5	59.6	2.9
MiniCPM-o 4.5	1958	50.3	93.4	9.1	67.5	95.5	14.1	29.7	96.4	2.2	32.5	94.4	5.9	45.0	94.9	7.8
Closed-source models																
Gemini-3.1-Pro	1996	48.0	53.1	25.3	**75.8**	59.9	43.9	29.4	68.5	20.0	34.1	84.1	16.5	46.8	66.4	26.4
Gemini-3.7-Flash	1995	26.4	84.3	11.0	57.8	78.4	35.4	19.5	68.0	10.1	27.2	58.5	17.1	32.8	72.3	18.4
Qwen3.5-Omni-Plus	1988	52.1	**14.4**	**57.7**	68.8	**27.9**	**65.2**	15.9	**26.2**	**23.2**	45.3	**30.0**	**37.5**	45.5	**24.6**	**45.9**
Muse-Spark-1.2	1996	26.0	27.3	30.1	40.1	50.5	32.1	7.5	82.4	5.9	18.6	77.1	13.6	23.0	59.3	20.4
Doubao-Seed-2.0-Lite	1997	27.9	71.8	14.1	25.8	77.0	31.7	5.8	82.2	3.7	9.1	98.6	4.8	17.1	82.4	13.6
B. Native Streaming *($[-1, +3]$ s audio progress)*																
MiniCPM-o 4.5	1999	9.0	15.4	13.0	4.0	39.6	22.6	**5.2**	53.6	7.4	5.2	51.1	5.4	5.9	39.9	12.1
Qwen3.5-Omni-Plus Realtime	1999	**21.5**	**4.0**	**35.3**	**15.8**	9.5	**45.6**	1.7	13.7	4.0	**8.4**	14.1	**44.1**	**11.9**	10.3	**32.2**
GPT Realtime 2	1533	12.3	12.6	10.1	10.9	14.1	28.2	2.9	34.4	5.5	3.6	26.3	8.8	7.4	21.8	13.1
StepAudio 3 Realtime	1999	8.3	5.3	12.5	3.9	**0.9**	21.4	0.8	**8.5**	**17.1**	5.1	**9.8**	12.5	4.5	**6.1**	15.9

n: scoreable requests; averages: domains, then tasks. **Bold**/underlined: best/second-best within each panel; ties share ranks. Protocol and full results: Appendix B.

5.3 Temporal Distribution of Responses



(a) Simulated streaming (b) Native streaming

Figure 4: Response timing on positive requests, normalized within models and averaged equally across models. Shading marks matching windows; further details appear in Appendix B.

Native notifications arrive later. In Figure 4, 55.5% of correct simulated outputs occur at the reference checkpoint (30.9% one step before; 13.6% one step after). Native median offsets are 1.1–1.7 seconds later, consistent with turn detection and generation scheduling. These offsets measure audio progress; simulated timestamps exclude inference time.

Errors differ in timing. Simulated errors concentrate before or near the reference checkpoint; native errors cluster beyond the matching window with a longer late tail.

5.4 Request Completion and Response Behavior

Figure 5 decomposes positive requests within their observed horizon. Complete execution requires all notifications to match in timing and content, with no extra replies.

Higher recall does not imply complete execution. Baichuan completes only 0.5% of positive requests; 55.5% cover all opportunities with extras. On Tracking, Gemini-3.1-Pro has higher recall than Qwen3.5-Omni-Plus but lower complete execution (17.6% versus 34.0%).

Native failures differ: Qwen Realtime and StepAudio have no reply on 59.3% and 71.9% of positive requests; MiniCPM replies without a valid match on 49.8%.

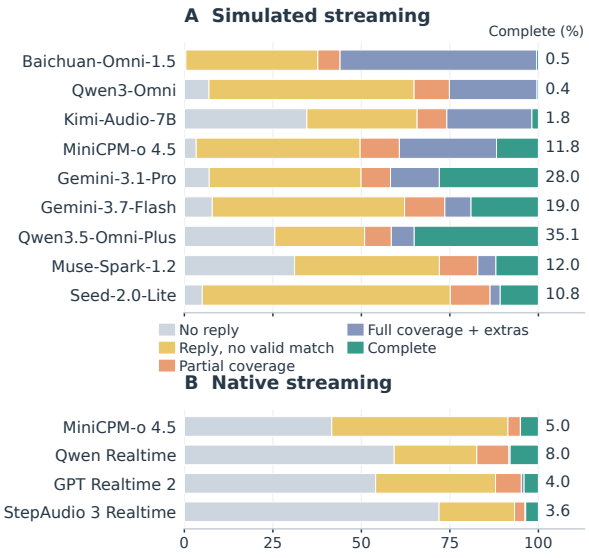


Figure 5: Positive-request outcomes, macro-averaged across domains and tasks. Right: complete execution rates. Partial coverage may include extra replies.

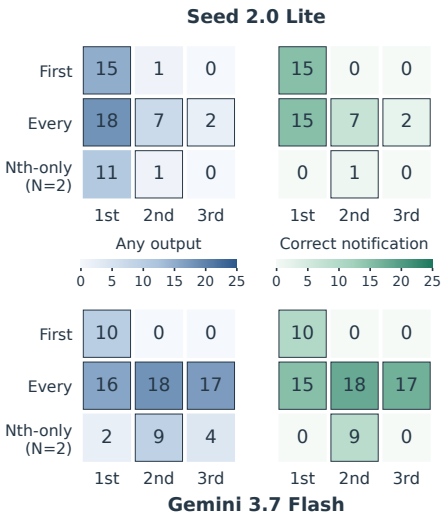


Figure 6: Rule contrasts over 25 groups. Blue: any output; green: correct notifications (included in blue). Outlines mark required positions.

Full coverage with extras means that every required notification is matched but additional replies are emitted. These replies leave recall unchanged while reducing precision and preventing complete execution. On Tracking, this outcome accounts for 38.0% of Gemini-3.1-Pro requests versus 18.9% for Qwen3.5-Omni-Plus. Since this decomposition covers positive requests, FAR provides the complementary view of silence on negative requests.

5.5 Controlled Notification-Rule Analysis

To distinguish notification-rule effects from acoustic differences, we apply First, Every, and Nth-only ($N = 2$) to the same 25 verified audio–target groups, observing full recordings. Seven configurations each complete 75 requests under the main matching criteria (Figure 6; details in Appendix C).

Occurrence claims are often premature. Seed correctly notifies the first occurrence in 15/25 groups under First, but under Nth-only it still outputs near the first in 11/25. All 25 Nth-only replies claim a second occurrence, yet 23 precede its eligible window; only one correctly notifies the second occurrence. For example, on the recording containing the Chinese phrase for "county town," both rules elicit a reply at 4 seconds, although the second occurrence begins at 12 seconds. The claimed ordinal changes while the response time remains fixed.

First-notification success does not ensure sustained coverage. On the same 25 Every groups, Seed's correct-notification counts fall from 15 to 7 to 2 across the first three occurrences; Gemini-3.7-Flash reaches 15, 18, and 17. All three positions use the same audio groups, so the decline does not result from fewer recordings contributing to later columns. Together, these fixed-audio results reveal failures in waiting and sustained notification that first-occurrence accuracy alone does not capture.

6 Conclusion

ProactiveAudioBench studies the sustained execution of listening instructions across sound events, speech, and music. Its four tasks organize first, repeated, and Nth-occurrence notifications alongside acoustic-reference matching. The benchmark pairs these listening requirements with annotated response opportunities and an evaluation framework for simulated and native streaming, providing a basis for studying notification timing, reply correctness, and silence under user-specified conditions.

AI USE STATEMENT

Large language models assist event annotation and listening request–response construction, with human verification of grounding and retained samples as described in Section 3.3. The evaluation framework uses a separate language-model judge for the content assessment described in Section 4.2. AI tools also assisted manuscript drafting, editing, and figure preparation. The authors are responsible for the final manuscript, data, and interpretation.

ETHICS STATEMENT

ProactiveAudioBench is intended for research on instruction-following audio assistance. It uses prerecorded sound, speech, and music, including samples constructed through signal insertion, splicing, or mixing. Model-assisted annotations and generated listening requests undergo human review. Use and redistribution of source recordings are subject to their applicable licenses and permissions.

Speech recordings and acoustic references may contain identifiable voices. Applications of continuous listening should obtain appropriate consent, protect personal information, and avoid covert monitoring or unauthorized voice imitation. Our evaluation uses prerecorded inputs, and tool selections are scored as text without executing external actions. Deployment decisions should consider application-specific privacy and safety requirements.

REPRODUCIBILITY STATEMENT

Section 3 describes the task definitions, data construction, and dataset composition. Section 4 specifies causal input, observation horizons, response representation, matching, and metric aggregation. Appendix A provides review and selection details, request counts, and native response reconstruction and timestamp handling.

REFERENCES

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, et al. Phi-4-Mini Technical Report: Compact yet Powerful Multimodal Language Models via Mixture-of-LoRAs, 2025. URL https://arxiv.org/abs/2503.01743.

Pengfei Cai, Yan Song, Qing Gu, Nan Jiang, Haoyu Song, and Ian McLoughlin. Detect Any Sound: Open-Vocabulary Sound Event Detection with Multi-Modal Queries, 2025. URL https://arxiv.org/abs/2507.16343.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online Video Large Language Model for Streaming Video, 2024. URL https://arxiv.org/abs/2406.11816.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-Audio: Advancing Universal Audio Understanding via Unified Large-Scale Audio-Language Models, 2023. URL https://arxiv.org/abs/2311.07919.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, et al. MiniCPM-o 4.5: Towards Real-Time Full-Duplex Omni-Modal Interaction, 2026. URL https://arxiv.org/abs/2604.27393.

Yuhang Dai, Xin Shu, Zengxi Li, Lei Xie, Xiangang Li, and Jianwei Yu. TAG-Bench: Benchmarking Temporal Audio Grounding in Large Audio Language Models, 2026. URL https://arxiv.org/abs/2609.01542.

Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. Pengi: An Audio Language Model for Audio Tasks. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.11834.

Min Jang, Orevaoghene Ahia, Nazif Tamer, Sachin Kumar, Yulia Tsvetkov, and Noah A. Smith. BASS: Benchmarking Audio LMs for Musical Structure and Semantic Reasoning, 2026. URL https://arxiv.org/abs/2602.04085.

Yadong Li, Jun Liu, Tao Zhang, Tao Zhang, Song Chen, et al. Baichuan-Omni-1.5 Technical Report, 2025. URL https://arxiv.org/abs/2501.15368.

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, 2025. URL https://arxiv.org/abs/2503.04721.

Zihan Liu, Zhikang Niu, Qiuyang Xiao, Zhisheng Zheng, Ruoqi Yuan, Yuhang Zang, Yuhang Cao, Xiaoyi Dong, Jianze Liang, Xie Chen, Leilei Sun, Dahua Lin, and Jiaqi Wang. STAR-Bench: Probing Deep Spatio-Temporal Reasoning as Audio 4D Intelligence. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.24693.

Xudong Lu, Xueying Li, Annan Wang, Yang Bo, Jinpeng Chen, Zengliang Li, Nianzu Yang, Rui Liu, Xue Yang, Jingwen Hou, and Hongsheng Li. OmniInteract: Benchmarking Real-World Streaming Interaction for Real-Time Omnimodal Assistants, 2026. URL https://arxiv.org/abs/2605.26485.

Yinghao Ma, Siyou Li, Juntao Yu, Emmanouil Benetos, and Akira Maezawa. CMI-Bench: A Comprehensive Benchmark for Evaluating Music Instruction Following, 2025a. URL https://arxiv.org/abs/2506.12285.

Ziyang Ma, Yinghao Ma, Yanqiao Zhu, Chen Yang, Yi-Wen Chao, Ruiyang Xu, Wenxi Chen, Yuanzhe Chen, Zhuo Chen, Jian Cong, Kai Li, Keliang Li, Siyou Li, Xinfeng Li, Xiquan Li, Zheng Lian, Yuzhe Liang, Minghao Liu, Zhikang Niu, Tianrui Wang, Yuping Wang, Yuxuan Wang, Yihao Wu, Guanrou Yang, Jianwei Yu, Ruibin Yuan, Zhisheng Zheng, Ziya Zhou, Haina Zhu, Wei Xue, Emmanouil Benetos, Kai Yu, Eng-Siong Chng, and Xie Chen. MMAR: A Challenging Benchmark for Deep Reasoning in Speech, Audio, Music, and Their Mix, 2025b. URL https://arxiv.org/abs/2505.13032.

Puneet Mathur and Dinesh Manocha. DuplexSpeechBench-IFEval: Evaluating Implicit Instruction Following in Full-Duplex Voice Agents, 2026. URL https://arxiv.org/abs/2609.03423.

S Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. MMAU: A Massive Multi-Task Audio Understanding and Reasoning Benchmark, 2024. URL https://arxiv.org/abs/2410.19168.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-ASR Technical Report, 2026. URL https://arxiv.org/abs/2601.21337.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. SALMONN: Towards Generic Hearing Abilities for Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.13289.

Apoorv Vyas, Heng-Jui Chang, Cheng-Fu Yang, Po-Yao Huang, Luya Gao, Julius Richter, Sanyuan Chen, Matt Le, Piotr Dollár, Christoph Feichtenhofer, Ann Lee, and Wei-Ning Hsu. Pushing the Frontier of Audiovisual Perception with Large-Scale Multimodal Correspondence Learning, 2025. URL https://arxiv.org/abs/2512.19687.

Yueqian Wang, Songxiang Liu, Disong Wang, Nuo Xu, Guanglu Wan, Huishuai Zhang, and Dongyan Zhao. MMDuet2: Enhancing Proactive Interaction of Video MLLMs with Multi-Turn Reinforcement Learning, 2025a. URL https://arxiv.org/abs/2512.06810.

Yueqian Wang, Xiaojun Meng, Yifan Wang, Huishuai Zhang, and Dongyan Zhao. ProactiveVideoQA: A Comprehensive Benchmark Evaluating Proactive Interactions in Video Large Language Models, 2025b. URL https://arxiv.org/abs/2507.09313.

Zhifei Xie, Zihang Liu, Ze An, Xiaobin Hu, Yue Liao, Ziyang Ma, Dongchao Yang, Mingbao Lin, Deheng Ye, Shuicheng Yan, and Chunyan Miao. Audio Interaction Model, 2026. URL https://arxiv.org/abs/2606.05121.

Ke Xu, Yuhao Wang, and Yu Wang. From Reactive to Proactive: Assessing the Proactivity of Voice Agents via ProVoice-Bench, 2026. URL https://arxiv.org/abs/2604.15037.

Dongchao Yang, Helin Wang, Yuexian Zou, Fan Cui, and Yujun Wang. Detect what you want: Target Sound Detection, 2021. URL https://arxiv.org/abs/2112.10153.

Xichen Zhang, Guankai Li, Yinghao Zhu, Shijian Wang, Sitong Wu, Shaozuo Yu, Meng Chu, Yuan Lu, and Jiaya Jia. StreamArena: Toward Continuous, Interactive, and Long-Horizon Agentic Streaming Video Understanding, 2026. URL https://arxiv.org/abs/2608.05703.

Ruixiang Zhao, Jie Yang, Zijie Xin, Tianyi Wang, Fengyun Rao, Jing LYU, and Xirong Li. OmniPro: A Comprehensive Benchmark for Omni-Proactive Streaming Video Understanding, 2026. URL https://arxiv.org/abs/2605.18577.

A Dataset and Evaluation Details

Review and selection. Annotators work by subtask and check audio quality, request–reply validity, and trigger accuracy. Each subtask or delivery batch is audited on 30% of its items; acceptance requires over 98% of the checked items to satisfy all three criteria, otherwise the batch is returned for revision.

Of 4,701 candidates, 2,932 were accepted, 840 marked for revision, and 929 excluded. From the 3,772 accepted/revised candidates, quotas by subtask, request language, and positive/negative label yielded 1,999 requests (1,640 accepted and 359 revised). Selection prioritizes accepted items and coverage of positive trigger positions within the available candidates; revisions are applied before inclusion. Table 3 gives the final composition.

Table 3: Final request counts by audio domain, listening task, and content group.

		English		Chinese		
Task	**Content group**	**Pos.**	**Neg.**	**Pos.**	**Neg.**	**Total**
Sound events		**160**	**156**	**159**	**155**	**630**
Detection (180)	Event categories	38	37	38	37	150
	Noise conditions	8	7	8	7	30
Tracking (150)	Event occurrences	38	37	38	37	150
Counting (150)	Event occurrences	38	37	38	37	150
Matching (150)	Reference sounds	38	38	37	37	150
Speech		**187**	**175**	**213**	**184**	**759**
Detection (310)	Keywords	10	9	10	9	38
	Phrases	9	9	10	9	37
	Topics	9	9	10	9	37
	Intents	10	9	10	9	38
	Tool-use requests	25	25	25	25	100
	Age groups	0	0	10	10	20
	Pronunciation	10	10	10	10	40
Tracking (150)	Word/phrase occurrences	38	37	38	37	150
Counting (150)	Word/phrase occurrences	36	31	51	32	150
Matching (149)	Speaker identity	40	36	39	34	149
Music		**154**	**154**	**151**	**151**	**610**
Detection (154)	Instruments	15	15	12	12	54
	Vocals	10	10	10	10	40
	Performance techniques	10	10	10	10	40
	Key changes	5	5	5	5	20
Tracking (148)	Musical sections	37	37	37	37	148
Counting (148)	Musical sections	37	37	37	37	148
Matching (160)	Reference segments	40	40	40	40	160
All requests		**501**	**485**	**523**	**490**	**1,999**

Shaded rows give domain totals; parentheses give task totals. Pos./Neg. denote positive/negative requests.

Additional dataset statistics. Positive requests yield 1,851 checkpoint-grouped reference opportunities for simulated evaluation, or 1,941 when recoverable same-second occurrences are retained separately for native evaluation. The audio-file total and aggregate duration are deduplicated by file; duration distributions include all requests, count reused audio once per request, and are normalized within each domain. Music requests generally use longer recordings.

Distribution conventions. Figure 3 uses checkpoint-level annotations for notification counts and reference positions. Among positive requests, 94.2% of Tracking requests require at most 10 notifications and 93.2% of Counting requests specify an index of at most 10. Each count bar represents one integer and is stacked by domain; the axes are linear up to 10 and logarithmic thereafter. Reference positions are relative to main-audio duration and normalized within each task, weighting opportunities equally. Detection, Tracking, and Matching opportunities span multiple recording regions. Lines in the duration and position plots connect bin centers.

Native response reconstruction. Replies are assembled by response identifiers or completion signals, or from continuous outputs using fixed boundary rules. Extraction is independent of reference answers. All nonempty, non-control text emitted during monitoring is retained, including canceled, failed, or unfinished replies; completion status and reference-setup outputs remain separate. Coincident replies stay distinct and retain trace order; empty traces do not establish successful silence.

Timestamps. Onsets use input-audio progress at the first body-text fragment after control removal; later completion does not shift them. Post-stream replies retain their phase and wall-clock offset from the recording's end. Audio progress is capped at the observed source endpoint, excluding appended silence, and is distinct from wall-clock latency.

B Additional Experimental Details

Coverage and records. Table 4 retains all 23 configurations; the main table highlights 13. Qwen Realtime denotes Qwen3.5-Omni-Plus Realtime. Known incomplete observations are excluded. Some legacy runners stored failed calls as waiting outputs, so observed silence cannot always be separated from call failures.

Table 4: Complete results for all 23 evaluated configurations (%). R and P require both temporal and content correctness; FAR is the response rate on negative requests. Temporal matching follows the panel-specific rule.

		Detection			Tracking			Counting			Matching			Avg.		
Model	n	**R↑**	**FAR↓**	**P↑**	**R↑**	**FAR↓**	**P↑**	**R↑**	**FAR↓**	**P↑**	**R↑**	**FAR↓**	**P↑**	**R↑**	**FAR↓**	**P↑**
A. Simulated Streaming *(± 1 checkpoint)*																
Open-source models																
Phi-4-Multimodal	1994	1.7	3.4	1.7	3.1	4.5	1.5	0.0	0.5	0.0	0.0	0.4	0.0	1.2	2.2	0.8
Audio Flamingo Next	1994	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0
Voxtral-Mini-3B	1994	22.9	99.6	1.0	37.4	99.5	4.4	13.1	99.1	0.5	18.2	99.6	0.3	22.9	99.4	1.5
Voxtral-Small-24B	1772	19.8	61.9	1.3	32.9	76.6	4.8	14.4	82.0	0.7	12.7	72.9	0.4	20.0	73.4	1.8
MOSS-Audio-8B	1994	12.4	63.8	0.7	18.6	86.0	3.0	22.5	88.6	0.5	5.5	89.9	0.1	14.7	82.1	1.1
Baichuan-Omni-1.5	1999	58.3	98.0	1.4	71.2	99.5	4.5	59.4	100.0	0.9	48.3	100.0	0.7	59.3	99.4	1.9
Qwen2-Audio-7B	1994	15.9	84.3	0.4	13.5	80.6	1.6	22.0	84.0	0.3	5.6	23.3	0.7	14.3	68.0	0.8
Qwen2.5-Omni-7B	1999	20.8	29.9	13.0	18.5	39.6	18.2	4.4	14.6	6.0	7.0	47.0	1.5	12.7	32.8	9.7
Qwen3-Omni-30B-A3B	1393	24.6	87.5	1.5	41.7	94.5	3.2	18.7	94.7	0.8	34.0	97.9	1.2	29.7	93.6	1.7
Kimi-Audio-7B	1999	25.9	52.9	1.9	42.9	65.8	8.3	31.0	62.0	0.8	18.3	57.8	0.8	29.5	59.6	2.9
MiniCPM-o 4.5	1958	50.3	93.4	9.1	67.5	95.5	14.1	29.7	96.4	2.2	32.5	94.4	5.9	45.0	94.9	7.8
Closed-source models																
Gemini-3.1-Pro	1996	48.0	53.1	25.3	75.8	59.9	43.9	29.4	68.5	20.0	34.1	84.1	16.5	46.8	66.4	26.4
Gemini-3.7-Flash	1995	26.4	84.3	11.0	57.8	78.4	35.4	19.5	68.0	10.1	27.2	58.5	17.1	32.8	72.3	18.4
Gemini-3.8-Flash	1993	28.3	72.8	13.5	53.3	76.6	38.6	20.2	58.9	11.4	35.8	46.7	25.3	34.4	63.8	22.2
Qwen3.5-Omni-Plus	1988	52.1	14.4	57.7	68.8	27.9	65.2	15.9	26.2	23.2	45.3	30.0	37.5	45.5	24.6	45.9
Muse-Spark-1.2	1996	26.0	27.3	30.1	40.1	50.5	32.1	7.5	82.4	5.9	18.6	77.1	13.6	23.0	59.3	20.4
Doubao-Seed-2.0-Lite	1997	27.9	71.8	14.1	25.8	77.0	31.7	5.8	82.2	3.7	9.1	98.6	4.8	17.1	82.4	13.6
B. Native Streaming *($[-1, +3]$ s audio progress)*																
MiniCPM-o 4.5	1999	9.0	15.4	13.0	4.0	39.6	22.6	5.2	53.6	7.4	5.2	51.1	5.4	5.9	39.9	12.1
Qwen3.5-Omni-Plus Realtime	1999	21.5	4.0	35.3	15.8	9.5	45.6	1.7	13.7	4.0	8.4	14.1	44.1	11.9	10.3	32.2
GPT Realtime 2	1533	12.3	12.6	10.1	10.9	14.1	28.2	2.9	34.4	5.5	3.6	26.3	8.8	7.4	21.8	13.1
StepAudio 3 Realtime	1999	8.3	5.3	12.5	3.9	0.9	21.4	0.8	8.5	17.1	5.1	9.8	12.5	4.5	6.1	15.9
Seeduplex	1482	6.5	3.8	8.9	6.5	3.2	29.4	1.9	2.4	5.6	—	—	—	—	—	—
GPT Live 1[‡]	1477	13.6	23.9	7.2	18.2	40.1	20.9	1.8	45.8	0.6	—	—	—	—	—	—

n denotes the number of scoreable requests; coverage differs across models. Scores are averaged equally over the three audio domains and then over the four tasks. — indicates that the required strata are unavailable. Matching is not evaluated for Seeduplex or GPT Live 1. [‡]GPT Live 1 uses reconstructed output episodes (Appendix A). GPT Realtime 2 excludes 466 unresolved long-audio requests. Simulated streaming uses predefined observation windows, whereas native streaming uses full-source observations. Tracking uses checkpoint-level opportunities in A and individually recoverable source occurrences in B. Comparisons should be made within each panel.

Timing distributions. Figure 4 uses positive requests. Correct offsets use the jointly matched reference; unmatched outputs use the nearest reference solely for visualization. Each output class is normalized within each model, then averaged equally across models in its mode.

Bins are one checkpoint for simulated outputs, 0.5 s for native correct outputs within $[-1, +3]$ s, and 1 s for native errors. Out-of-view events remain in the denominators: simulated, 38.3% early and 3.6% late; native, 8.7% and 22.5%.

Native correctness is conditioned on $[-1, +3]$ s, so the error increase beyond $+3$ s partly reflects the matching boundary. Simulated positive requests typically end three steps after the last annotated trigger, leaving later outputs unobserved. Audio-progress offsets are distinct from wall-clock inference latency.

C CONTROLLED NOTIFICATION RULES: FULL RESULTS

Data and protocol. The set contains 25 verified audio–target groups (eight Sound, 13 Speech, four Music), each at most 90 seconds long. First, Every, and Nth-only ($N = 2$) use identical audio, targets, and full-recording observation: 75 requests per configuration and 525 traces overall. Simulated streaming uses one-second increments, actual reply history, and ± 1 checkpoint matching; native streaming uses $[-1, +3]$ seconds of audio progress. Results are separated by mode. MiniCPM receives the condition through system context. All requests are positive, so negative-request FAR is undefined.

Window coverage and joint correctness. Columns index verified target occurrences, regardless of the ordinal claimed in a reply. Blue counts occurrences with non-control output in their timing windows; repeated output within a window counts once. Green preserves the order-preserving one-to-one matches satisfying the rule, timing, and content criteria. Blue includes green. Outlines mark required positions. Green zeros elsewhere are imposed by the rule, so blue must be consulted to assess whether the model spoke.

Denominators and omitted outputs. The first three columns use all 25 groups; 4th+ pools 41 later occurrences from 16 groups. Labels show numerator/denominator and colors show the ratio, not domain-macro recall: Seed First is 15/25 but 50.64% after domain averaging; Nth-only is 1/25 but 2.56%. Off-window counts requests with output outside every target window; Silent counts requests with no output. Both denominators are 25. In-window and off-window replies can coexist in a request, so rows need not sum to 25.

Two of 2,167 outputs overlap adjacent windows, both from Qwen Realtime on one group under First and Every. Blue counts both covered windows; green retains the unique joint assignment. The main Seed/Gemini figure has no overlapping-window outputs.

Behavior across configurations. Qwen3-Omni responds around multiple occurrences even under First and Nth-only, and every rule has 24/25 requests with off-window output. Muse covers the first three Every occurrences in 11, 8, and 11 groups, but has joint hits in only 6, 3, and 3. Seed already has low later coverage before content filtering: its second and third Every occurrences receive output in only seven and two groups. Thus, the plots distinguish absent timely output from emitted output that fails joint correctness.

Qwen Realtime has no Nth-only output near the first occurrence, yet 12/25 requests are silent, 9/25 contain off-window output, and only 3/25 correctly notify the second occurrence. StepAudio has 19/25 silent Nth-only requests; all seven outputs in the remaining six requests are outside every target window. MiniCPM has 17/25 Nth-only requests with off-window output and 3/25 silent requests. Low early-event output alone therefore does not establish correct waiting.

Checking the occurrence claims. All 25 Seed Nth-only requests produce one reply, each explicitly claiming a second occurrence. Nine precede the first eligible window, 11 fall near the first occurrence, and three lie between the first and second windows. One correctly matches the second occurrence; one follows its window and falls outside every target window. Thus, 23/25 replies precede the second eligible window. This content-specific finding is limited to the inspected Seed replies; other models' outputs can include acknowledgments.

For the "county town" example, the first two source occurrences begin at 3.12 and 12.00 seconds. Seed responds at 4 seconds under both First and Nth-only, changing only its occurrence claim. Identical timing is not universal: among 24 groups with exactly one Seed reply under both rules, only five have the same reply time.

Paired checks and interpretation. Among the 15 groups with a valid Seed First notification, Nth-only responds near the first occurrence in nine and correctly notifies the second in none. Gemini and Seed both correctly report the first Every occurrence on 11 groups (two Sound and nine Speech); they jointly match 30 and six of the 33 subsequent opportunities, respectively. This conditional subset supplements the full results and does not extend to Music. The study controls audio and observation horizon, but does not isolate perception, event segmentation, memory, or counting mechanisms.

Correct-notification counts measure opportunity-level joint hits; complete-request success also excludes extra replies, explaining Seed's 15/25 versus 14/25 First results. Ordinal 1–11 denominators are 25, 25, 25, 16, 8, 5, 5, 3, 2, 1, and 1; the sparse tail does not support a stable count trend.

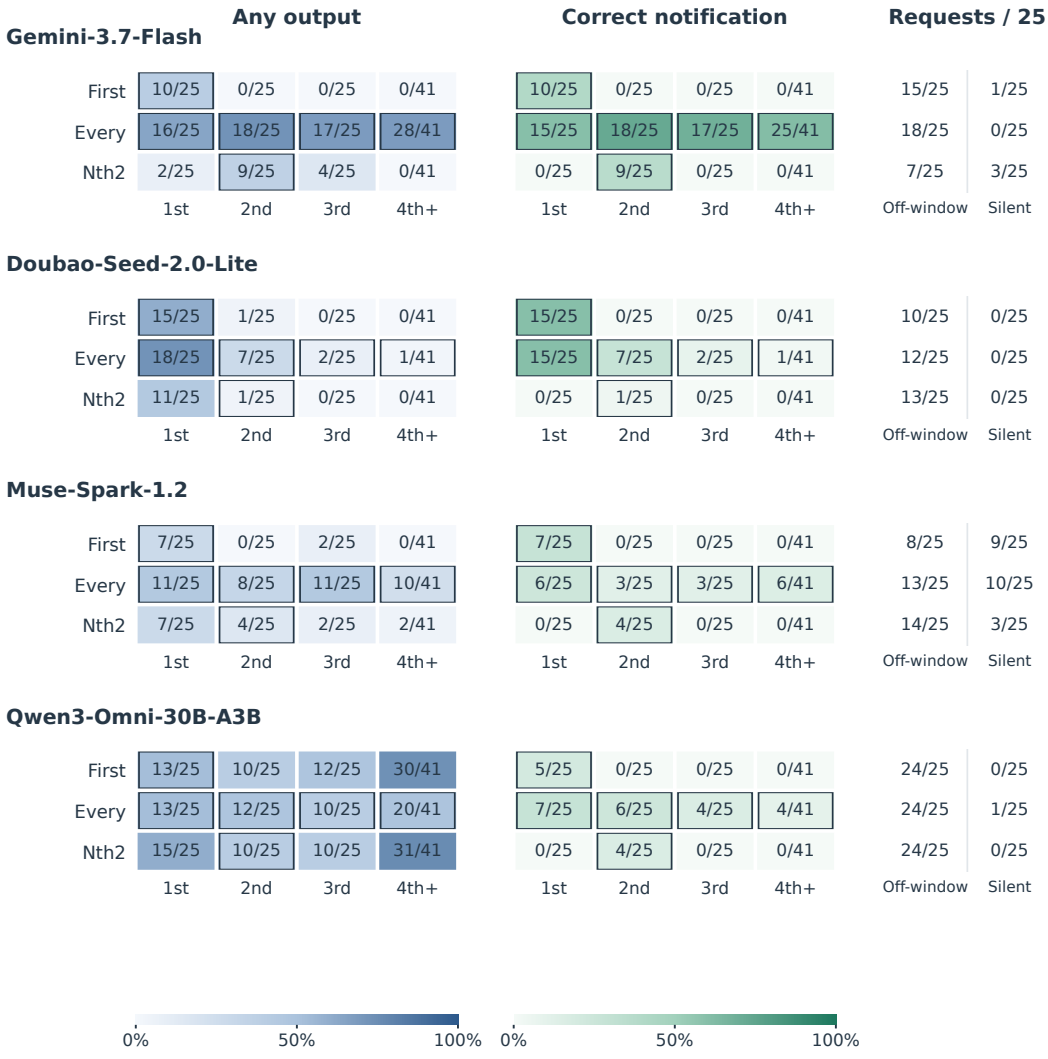


Figure 7: Full rule contrasts for four simulated configurations. Blue shows any-output coverage; green shows joint correctness. Outlines mark required positions. The first three columns each use 25 groups; 4th+ pools 41 later occurrences. Off-window and Silent count requests out of 25. Colors encode coverage fractions.

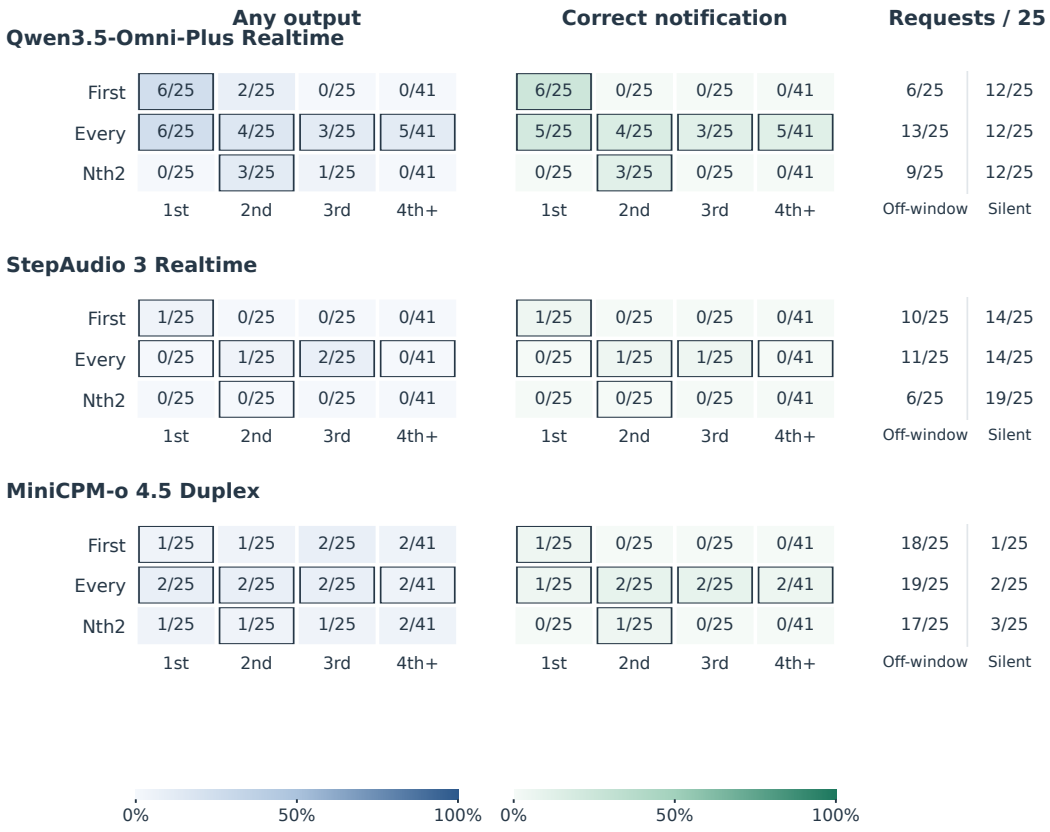


Figure 8: Full rule contrasts for three native configurations, with the same counting conventions as Figure 7. The timing window is $[-1, +3]$ seconds of audio progress. Early-event coverage should be read together with off-window outputs and silent requests.