SONATINE: SPATIAL AUDIO UNDERSTANDING VIA TOOL-INTEGRATED REASONING IN AUDIO LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most general-purpose Audio Large Language Models (Audio LLMs) accept only single-channel audio and therefore cannot use inter-channel spatial cues. Existing approaches to spatial audio understanding *internalize* these cues by feeding dense features extracted from multichannel First-Order Ambisonics (FOA) recordings into the model. However, such adaptation improves spatial understanding at a potential cost to general audio understanding. Spatial measurement can instead be *externalized*: signal-processing estimators compute quantities such as sound direction directly from FOA recordings, and their outputs are provided to an unchanged Audio LLM. But to reliably measure the queried target, such an estimator must operate on a *target-informative window* in which the queried sound is clear and overlap is limited. Because the estimator measures all active sounds in its input window, spatial information from other sources can otherwise be mixed into the measurement. To address this problem, we introduce **SONATINE**, a tool-integrated reasoning framework in which spatial estimators act as tools and the Audio LLM specifies a time window as an argument to each tool call. The Audio LLM, with no added spatial module, receives the question and only the mono channel, while the selected tool accesses all FOA channels within the specified window and returns a spatial observation. Within SONATINE, obtaining reliable spatial measurements requires *adaptive window selection*, in which the Audio LLM uses the question and mono audio to choose a target-informative window for each tool call. To teach this behavior and general spatial tool use, we construct **SONATINE-15k**, a dataset of 14,990 tool-use trajectories, and use it for Supervised Fine-Tuning (SFT), followed by Group Relative Policy Optimization (GRPO). On SO-Bench, our 7B model surpasses all evaluated internalized spatial baselines while using far fewer spatial QA examples, with its largest advantage under heavy source overlap. By externalizing spatial measurement, SONATINE also enables spatial audio understanding in closed-source models through prompting alone, helps retain general audio understanding, and supports transfer from FOA to a six-channel format without further training by changing only the tool implementations.

1 INTRODUCTION

Recent advances in Audio Large Language Models (Audio LLMs) have enabled them to recognize and reason about speech, music, and environmental sounds (Luo et al., 2026). Understanding a real acoustic scene, however, also requires locating sounds, estimating their distance, and tracking their motion. Many cues needed for these capabilities are encoded across the channels of a multichannel recording. Yet most general-purpose Audio LLMs (Google DeepMind, 2026; Xu et al., 2025b) accept only single-channel audio, thus causing their native interfaces to fail to access such inter-channel information. As a result, spatial audio understanding remains less explored than other forms of audio understanding.

Prior work has addressed this gap using First-Order Ambisonics (FOA), a four-channel format in which W is a mono recording of the full scene and X, Y, and Z capture directional variation along the front–back, left–right, and up–down axes, respectively (Tang et al., 2024; Sakshi et al., 2025a; Jiang et al., 2026; Zhu et al., 2026). Most FOA-based Audio LLMs *internalize* spatial information by feeding dense feature sequences derived from FOA into the model's input pathway and training

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

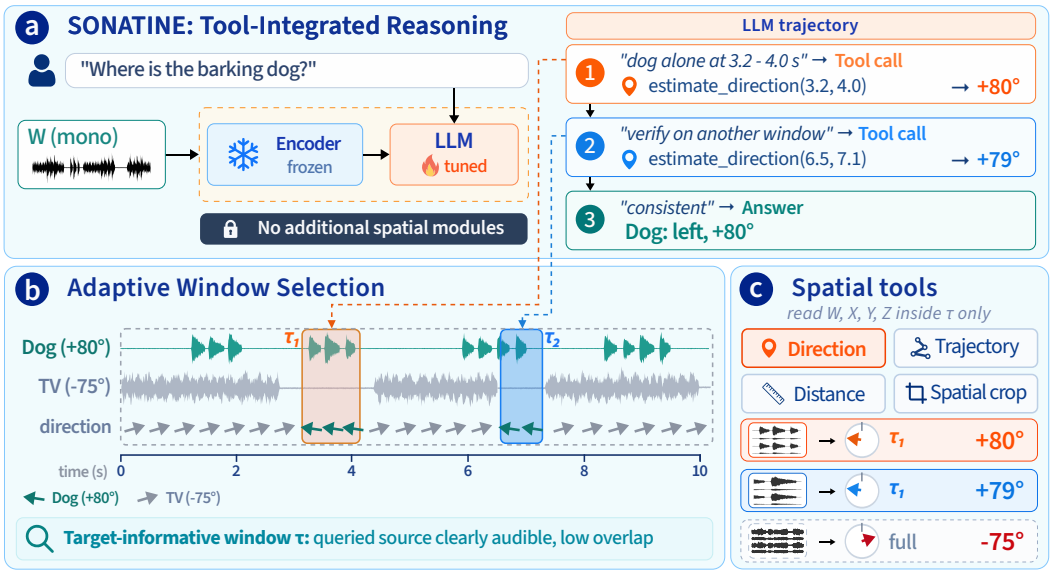


Figure 1: Overview of the SONATINE framework. (a) **Tool-integrated reasoning**: given a query and mono audio, the Audio LLM calls spatial tools until it has sufficient evidence to answer. (b) **Adaptive window selection**: to obtain reliable measurements, the model must select a window where the queried sound is audible and source overlap is limited. (c) **Spatial tools**: four tools process all recording channels within the model-specified window and return spatial evidence.

spatial modules, the downstream LLM, or both. Such adaptation, however, can alter the model's audio-language behavior and potentially trade general audio understanding for spatial capability.

Yet spatial information need not be internalized. The known physical relationships among FOA channels allow signal-processing estimators to compute quantities such as sound direction (Pulkki et al., 2017). Therefore, spatial measurement can be *externalized* without learning a dense multichannel representation. For example, Sridhar et al. (2025) use a fixed pipeline to extract spatial attributes from the recording and provide them as text to an unmodified LLM. However, applying a signal-processing estimator in the same fixed manner for every query can be unreliable because the estimator operates on all sounds active in its input window. Suppose a question asks where a dog barks while a television plays throughout the recording. Applied to the full recording, the estimator may report the persistent television's direction instead of the dog's (Fig. 1(c)). In contrast, it can correctly estimate the dog's direction from a short window where the bark is clear and the television is less prominent (Fig. 1(b)). We call such a window a *target-informative window*: a time window in which the queried sound is clearly audible and overlap from other sources is limited.

Tool-integrated reasoning (Gou et al., 2024) offers a way to retain externalized spatial measurement while making it possible for spatial tools to operate on target-informative windows. Once external spatial estimators are exposed as tools, the input window becomes a tool argument that the Audio LLM can specify during reasoning based on the question and audible content. Building on this observation, we introduce SONATINE (**S**patial audi**O** u**N**derst**A**nding via **T**ool-**IN**tegrated r**E**asoning), a novel framework that integrates an Audio LLM with external spatial estimators through tool-integrated reasoning. Through its unchanged input interface, the Audio LLM receives the question and only the W channel, then alternates between reasoning and tool calls until it answers. Each call specifies a spatial tool and a window over which the tool processes all four FOA channels. The tool then returns a spatial measurement or audio focused on a chosen direction, which is appended to the model context. Obtaining reliable measurements from these calls, as illustrated by the example in Fig. 1, requires *adaptive window selection*: selecting a target-informative window and invoking a spatial tool on it. To train the model to perform adaptive window selection, we construct **SONATINE-15k**, comprising 14,990 complete trajectories with reasoning, tool calls, and final answers. We train the Audio LLM to use spatial tools through Supervised Fine-Tuning (SFT) on these trajectories, followed by Group Relative Policy Optimization (GRPO) (Shao et al., 2024).

We evaluate SONATINE on SO-Bench (Zhu et al., 2026), a question-answering benchmark over FOA recordings, and find distinct benefits from externalized spatial processing and adaptive window selection. Our 7B SFT model surpasses the released encoder-based Spatial-Omni-7B (Zhu et al., 2026) checkpoint while using approximately $52\times$ fewer spatial QA training examples derived from the same source dataset, suggesting that not having to learn multichannel spatial features may contribute to this efficiency. Beyond this data efficiency, the externalized design provides three further benefits. First, it enables spatial audio understanding in closed-source Audio LLMs through prompting. Second, preserving the backbone's native input pathway helps retain general audio understanding. Third, the learned policy transfers across recording formats without further training. To evaluate this transfer, we construct an evaluation set from the six-channel microphone-array recordings in EasyCom (Donley et al., 2021), whose conversion to FOA is approximate and may discard some spatial information. In this evaluation, we retain the FOA-trained policy and keep the input and output formats of all four tools unchanged, replacing only their format-specific signal-processing implementations. With this six-channel backend, the FOA-trained SONATINE model outperforms all compared internalized spatial baselines on this format. Complementing these benefits, adaptive window selection shows its clearest advantage under heavy source overlap.

Our contributions are threefold:

- We introduce **SONATINE**, a novel tool-integrated framework in which an Audio LLM retains its native input interface while selecting spatial tools and time windows to obtain spatial information.
- We construct **SONATINE-15k**, a dataset of 14,990 tool-use trajectories designed to teach adaptive window selection, and show that the resulting model outperforms the released Spatial-Omni-7B checkpoint, with its largest advantage under heavy source overlap.
- We show that SONATINE's externalized design enables prompting-based spatial understanding in closed-source models, data-efficient adaptation, retention of general audio understanding, and transfer to a new recording format without further training.

2 Related Work

2.1 Spatial Audio Understanding in Audio LLMs

Most general-purpose Audio LLMs process a single audio channel, limiting their access to the inter-channel cues required for spatial perception. To overcome this limitation, prior work has developed Audio LLMs that process binaural or multichannel recordings to reason about source direction, distance, motion, and spatial relations. Examples include binaural models such as BAT (Zheng et al., 2024) and OWL (Biswas et al., 2025), which operate on left- and right-ear signals and encode channel-wise spectra and interaural phase differences, with OWL further introducing geometry-aware supervision and explicit spatial reasoning. Sridhar et al. (2025) introduce a stereo-based framework that applies a fixed preprocessing pipeline to estimate event-wise frame-level direction and distance over the full recording and provide these structured attributes to a reasoning LLM.

FOA-based work, which instead operates on a four-channel sound-field representation, has explored both analytically derived cues and learned spatial representations. Tang et al. (2024) inject intensity vectors alongside semantic features from the omnidirectional W channel. Subsequent systems use trainable spatial branches to transform multichannel spectral features, intensity vectors, or spatial covariance statistics into LLM-compatible embeddings (Jiang et al., 2026; Sakshi et al., 2025a; You et al., 2026). Spatial-Omni (Zhu et al., 2026), our primary baseline, adds a trainable SO-Encoder that injects FOA spatial tokens as an independent modality and introduces SO-Dataset and SO-Bench, which we use to construct SONATINE-15k and evaluate spatial understanding, respectively. JAEGER (Liu et al., 2026) further learns a Neural Intensity Vector directly from raw FOA waveforms and combines it with RGB-D observations for 3D audio-visual grounding and reasoning.

While most of the above systems assume fixed input geometries, PhaseCoder (Dementyev et al., 2026) conditions its spatial encoder on microphone coordinates to process raw audio from different array configurations. Unlike these methods, SONATINE exposes spatial estimators as tools and lets the Audio LLM select both a spatial operation and a time window during reasoning, enabling spatial audio understanding through tool use rather than learned multichannel representations.

2.2 TOOL-AUGMENTED MULTIMODAL REASONING

Tool-augmented reasoning extends language models with information and operations that are difficult to represent reliably in model parameters alone. Early approaches include ReAct (Yao et al., 2023), which interleaves reasoning with actions that query external environments, and Toolformer (Schick et al., 2023), which learns in a self-supervised manner when and how to invoke external APIs. Building on these approaches, ToRA (Gou et al., 2024) formalizes tool-integrated reasoning by interleaving natural-language rationales with executable programs for mathematical problem solving. The same paradigm has since been extended to multimodal perception. In the vision domain, training-free systems such as VisProg (Gupta & Kembhavi, 2023) and ViperGPT (Surís et al., 2023) synthesize executable programs that compose pretrained visual modules, whereas DeepEyes (Zheng et al., 2025) uses end-to-end reinforcement learning to decide when and where to crop an image for closer inspection.

Recent work in the audio domain has used external tools to recover acoustic evidence that may be lost or underused during reasoning. Audio-Maestro (Lee et al., 2025) integrates timestamped outputs from specialized tools, while AuTAgent (Tong et al., 2026) learns through reinforcement learning when and which tools to invoke; Audio-Mind (Wang et al., 2026a) further makes evidence acquisition conditional on whether the model's initial judgment is sufficient. Concurrent work introduces SpeechAgent-R (Wang et al., 2026b), which coordinates diverse acoustic tools across speech, sound-event, and audio-processing tasks, including speech localization and spatial target-speech extraction. In contrast to these approaches, SONATINE makes the time window an explicit argument of each spatial tool call and trains the Audio LLM's tool-use policy on SFT trajectories demonstrating target-informative window selection.

3 THE SONATINE FRAMEWORK

3.1 PROBLEM FORMULATION

We formulate spatial audio understanding over a First-Order Ambisonics (FOA) recording $\mathbf{A} = [W, X, Y, Z] \in \mathbb{R}^{4\times L}$ and a natural-language question q, where L is the number of audio samples. The W channel is omnidirectional, while X, Y, and Z encode directional variation along the front–back, left–right, and up–down axes, respectively. Let π_θ denote the Audio LLM policy and $\mathcal{T}$ the set of spatial tools available to it. Our framework separates monaural semantic listening from multichannel spatial measurement: π_θ receives only W through its native audio interface, whereas the tool executor can invoke tools in $\mathcal{T}$ over the full recording $\mathbf{A}$. Thus, the policy has no direct access to X, Y, and Z and can obtain their spatial evidence only through returned tool observations.

Fig. 1 summarizes this separation and the resulting tool-use loop, which interleaves model generation with tool execution. At interaction step i, the policy produces an intermediate textual rationale r_i and an action a_i. If a_i is non-terminal, it specifies exactly one tool call by jointly selecting a spatial tool k_i, a time window τ_i, and any optional tool-specific arguments $\boldsymbol{\eta}_i$ according to

$$(r_i, a_i) \sim \pi_\theta(\cdot \mid r_{<i}, a_{<i}, o_{<i}, q, W, \mathcal{T}), \qquad a_i = (k_i, \tau_i, \boldsymbol{\eta}_i), \quad \tau_i = [t_i^{\text{start}}, t_i^{\text{end}}]. \tag{1}$$

Here, k_i identifies a selected tool $T_{k_i} \in \mathcal{T}$, and $\boldsymbol{\eta}_i$ contains any optional arguments other than the time window. Alternatively, a_i may be a terminal action that emits the final answer. For a tool call, the selected tool operates on the portion of the full multichannel recording within τ_i and returns a spatial observation

$$o_i = T_{k_i}(\mathbf{A}|_{\tau_i}; \boldsymbol{\eta}_i), \tag{2}$$

where $\mathbf{A}|_{\tau_i}$ denotes the recording restricted to the selected window. Depending on the tool, the observation contains either a structured spatial measurement or spatially cropped audio that is returned as a mono signal emphasizing sounds from a selected direction. The observation is inserted into the model context before the next interaction step. Within a fixed call budget, the policy may repeat this process and accumulate observations from multiple sequential tool calls before emitting the final answer.

3.2 SPATIAL TOOL INTERFACE

Under the WXYZ convention used in this work, let $W_{f,t}$, $X_{f,t}$, $Y_{f,t}$, and $Z_{f,t}$ denote the complex short-time Fourier transform (STFT) coefficients of these components at frequency bin f and time

frame t. Their cross-spectral relationship yields the pseudo-intensity vector (PIV)

$$\mathbf{I}_{f,t} = \mathrm{Re}\{W^*_{f,t}\mathbf{V}_{f,t}\}, \qquad \mathbf{V}_{f,t} = [X_{f,t}, Y_{f,t}, Z_{f,t}]^\top. \tag{3}$$

The direction of $\mathbf{I}_{f,t}$ estimates where the dominant sound energy in time–frequency bin (f, t) comes from. We represent direction using azimuth (horizontal angle) and elevation (vertical angle). The FOA backend uses this time–frequency evidence to implement the direction, trajectory, and spatial-cropping tools below; together with distance estimation, SONATINE exposes four spatial tools to the model: Using the notation above, every tool receives the selected time window τ_i and returns an observation o_i; $\boldsymbol{\eta}_i$ is empty for all tools except `spatial_crop`.

- `estimate_direction`$(\tau_i) \rightarrow o_i$ uses the direction of $\mathbf{I}_{f,t}$ at each time–frequency bin within τ_i as a vote for a source direction. It returns up to three top-voted direction candidates, their vote shares, and a window-level diffuseness score.
- `get_trajectory`$(\tau_i) \rightarrow o_i$ applies the same direction-estimation procedure as `estimate_direction` at successive points within τ_i and returns a sequence of dominant source directions and their support scores over time.
- `spatial_crop`$(\tau_i, \boldsymbol{\eta}_i) \rightarrow o_i$ takes the target azimuth and elevation in $\boldsymbol{\eta}_i$ and returns spatially cropped audio.
- `get_distance`$(\tau_i) \rightarrow o_i$ returns a single distance estimate in meters. It uses a regressor architecture adapted from Berghi & Jackson (2026); the regressor is trained on approximately 200K labeled audio windows from the SO-Dataset (Zhu et al., 2026) training split. Appendix C.2 provides the implementation and training details.

The Audio LLM interacts with these four tools only through their arguments and returned observations. Keeping this interface consistent across recording formats allows a policy trained on one format to be applied to another by replacing only the format-specific tool implementations.

3.3 Adaptive Window Selection and SONATINE-15k

Each observation o_i summarizes the spatial evidence within the window τ_i selected by the policy. As motivated in Sec. 1, when the queried target overlaps with another source within τ_i, the source-agnostic measurement tools can return an observation o_i that mixes their spatial cues. In `estimate_direction`, for example, a window with substantial source overlap can produce bin-level estimates from multiple sources, splitting the votes across their directions and potentially making a non-target source the leading candidate. To prevent such distortion, the policy must use the question and the W channel to select a window with sufficient evidence from the target and limited interference from other sources. We call such a window *target-informative*, and its selection *adaptive window selection*.

To train adaptive window selection together with general spatial tool use via SFT, we construct **SONATINE-15k**, a curated dataset of 14,990 trajectories spanning 16 task types and drawn from 14,394 unique FOA recordings in the SO-Dataset (Zhu et al., 2026) training split. Of these, 11,990 contain one or more spatial tool calls, while the remaining 3,000 are no-tool trajectories for three task types that do not require spatial information. The tool-use trajectories supervise both the choice of spatial tool and its time window. For target-conditioned measurement calls, we construct target-informative windows by applying the adaptive window selection principle using source-activity annotations: we first choose a window duration from a predefined range and then, within the queried source's active spans, select the window placement that minimizes overlap from non-target sources. We then execute the appropriate tool on the selected window and retain the result only if its observation passes task-specific quality and consistency checks. For instance, in an azimuth-estimation trajectory, we select a low-overlap window containing the queried source, invoke `estimate_direction` on that window, and verify that the returned direction supports the ground-truth azimuth. Finally, we assemble the verified tool calls and observations into complete trajectories, using an LLM to synthesize the rationales. Appendix B.2 details the window selection and task-specific trajectory-construction procedures.

4 EXPERIMENTAL SETUP

4.1 DATASETS AND EVALUATION

Spatial evaluation. We evaluate spatial audio understanding on the 7,877 questions in SO-Bench (Zhu et al., 2026), an FOA-based open-ended QA benchmark covering the same 16 task types as the SO-Dataset training split. For aggregate reporting, we group the 16 tasks into four capability categories: non-spatial audio perception, direct spatial estimation, spatial grounding and relation understanding, and complex spatial-semantic reasoning. We orient all task-specific metrics so that higher is better and report unweighted means within each category and across all 16 tasks. Further details on the task taxonomy, scoring, and per-task results are provided in Appendices B.1, E.3, and E.6, respectively.

General audio understanding. To assess how well spatially adapted Audio LLMs retain general audio understanding, we evaluate them on the sound, music, and speech domains of MMAU (Sakshi et al., 2025b) and on single-audio, closed-choice questions from the same three domains of MMAU-Pro (Kumar et al., 2026). Both benchmarks are evaluated using multiple-choice accuracy without spatial tools.

Cross-format transfer. We construct a cross-format evaluation set from the EasyCom (Donley et al., 2021) dataset, which contains six-channel head-mounted microphone-array recordings of natural conversations captured with an AR-glasses headset. The set contains 350 questions drawn from all 12 sessions, with 70 questions for each of five tasks: single-speaker azimuth estimation, relative left–right localization of two speakers ordered by when they speak, azimuth estimation of a speaker specified by speaking order, transcription of a speaker specified by azimuth, and single-speaker distance estimation. We report wrap-aware angular mean absolute error (MAE) in degrees for both azimuth tasks, exact accuracy for relative left–right localization, word error rate (WER) for location-conditioned speech transcription, and distance MAE in meters for distance estimation. See Appendix E.5 for the benchmark construction and cross-format evaluation protocol.

4.2 MODELS AND IMPLEMENTATION

General-purpose Audio LLMs. We evaluate the closed-source GPT-Audio 1.5 (OpenAI, 2026) and Gemini 3.8 Flash (Google DeepMind, 2026) models and the open-weight Qwen2.5-Omni-7B (Xu et al., 2025a), Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025b), Gemma 4 12B IT (Gemma Team et al., 2026), Kimi-Audio-7B-Instruct (KimiTeam et al., 2025), and Audio Flamingo Next (Ghosh et al., 2026) models. In the no-tool setting, each model receives the question and the monaural W channel of the FOA recording. Appendix D.5 gives the exact decoding and interaction protocol.

Baselines with internalized spatial processing. We compare SONATINE with three baselines that internalize spatial processing in learned representations: BAT (Zheng et al., 2024), JAEGER (Liu et al., 2026), and Spatial-Omni-7B (Zhu et al., 2026). For JAEGER, we use its audio-only localization variant. This variant and Spatial-Omni-7B operate directly on FOA recordings using their learned spatial representations. Because BAT expects binaural input, we render each FOA evaluation recording using a first-order magnitude least-squares (MagLS) binaural decoder (Schörkhuber et al., 2018) and provide the resulting left and right channels to BAT.

SONATINE variants. We apply SONATINE to the closed-source Audio LLMs by exposing the spatial tools through their native function-calling APIs and prompting them to perform adaptive window selection. For our trainable SONATINE variants, we use Qwen2.5-Omni-7B and Qwen3-Omni-30B-A3B-Instruct as backbones, referring to the resulting models as **SONATINE 7B** and **SONATINE 30B**, respectively. We compare each backbone before tool-use training, after SFT on SONATINE-15k, and after further refinement with GRPO using questions from the training split of SO-Dataset (Zhu et al., 2026). Appendix D documents the full prompting and training setup, including SFT and GRPO optimization.

Tool backends. For FOA recordings, we implement the four tools introduced in Sec. 3.2: `estimate_direction`, `get_trajectory`, `spatial_crop`, and `get_distance`. For six-channel recordings, we expose the same four tools to the Audio LLM with the same arguments and output fields, changing only their implementations to support the six-channel recording format. Appendix C details both backends.

Table 1: Overall comparison on SO-Bench across four capability categories. General-purpose Audio LLMs are evaluated without spatial tools, whereas SONATINE variants use the shared spatial tool interface; closed-source models access the tools through native function calling (Appendix D.5). All scores are percentages; higher is better, and the best result in each column is shown in bold.

Model	Non-Spatial Perception	Direct Spatial Estimation	Grounding & Relations	Complex Spatial-Semantic	Overall
General-purpose Audio LLMs					
GPT-Audio 1.5	13.14	27.53	27.75	11.23	20.84
Gemini 3.8 Flash	21.24	27.02	31.83	11.57	23.88
Qwen2.5-Omni 7B	20.47	33.73	36.64	13.20	27.20
Qwen3-Omni 30B	20.75	32.84	36.49	14.95	27.47
Gemma 4 12B	18.35	34.86	31.93	5.26	23.27
Kimi-Audio 7B	15.60	17.82	34.76	12.22	22.35
Audio Flamingo Next	13.76	31.71	34.86	6.97	23.34
Internalized spatial baselines					
BAT	2.74	18.52	10.51	4.28	9.00
JAEGER (Audio)	10.16	33.89	26.58	9.87	20.69
Spatial-Omni-7B	57.83	**78.01**	63.60	32.58	57.47
SONATINE (closed-source prompting)					
GPT-Audio 1.5	19.23	66.72	26.45	18.93	30.77
Gemini 3.8 Flash	25.48	73.86	40.69	26.67	40.55
SONATINE (tool-use training)					
SONATINE 7B (SFT)	**62.62**	74.14	**64.40**	42.26	60.36
SONATINE 30B (SFT)	59.47	75.19	62.33	43.16	59.41
SONATINE 7B (SFT + GRPO)	62.17	75.36	64.36	42.90	60.65
SONATINE 30B (SFT + GRPO)	62.51	75.66	63.07	**44.89**	**60.78**

5 Experimental Results

Table 1 compares general-purpose Audio LLMs, internalized spatial baselines, and SONATINE on the four SO-Bench capability categories. The evaluated open-weight general-purpose Audio LLMs obtain only 22.35–27.47 overall because their single-channel inputs lack multichannel spatial cues. We treat Spatial-Omni-7B as our primary baseline because it uses the same Qwen2.5-Omni-7B backbone as SONATINE 7B and is trained on spatial QA examples from the SO-Dataset.

Spatial understanding through prompting. With GPT-Audio 1.5, SONATINE improves direct spatial estimation by 39.19 points and the overall score by 9.93 points. With Gemini 3.8 Flash, it improves all spatial capability categories, including a 46.84-point gain in direct spatial estimation, and raises the overall score by 16.67 points. These gains show that the SONATINE framework can enable spatial audio understanding in closed-source models by externalizing spatial measurement.

Gains from spatial tool-use learning. SFT on SONATINE-15k improves the overall score by 33.16 points for Qwen2.5-Omni-7B and 31.94 points for Qwen3-Omni 30B. The gains span all four capability categories, showing that the supervised trajectories teach both direct use of spatial measurements and their integration into spatial-semantic reasoning. These SFT policies also exceed Spatial-Omni-7B in overall score by 2.89 points for SONATINE 7B and 1.94 points for SONATINE 30B. GRPO further improves the overall score by 0.29 points for SONATINE 7B and 1.37 points for SONATINE 30B.

Cost-efficient spatial adaptation. SONATINE SFT uses 14,990 tool-use trajectories, approximately $52\times$ fewer than the 787K Spatial-Omni-7B training examples estimated from the released artifacts (Appendix E.1). We hypothesize that this data efficiency comes from externalizing spatial measurement: fixed tools supply multichannel spatial cues, so the Audio LLM does not need to learn a spatial encoder from spatial QA supervision. Although our framework additionally trains the distance tool on approximately 200K $(\text{audio}, \text{distance})$ pairs, their scalar labels are substantially cheaper to obtain than spatial QA annotations, and the lightweight distance regressor is far smaller than the Audio LLM, keeping both data construction and model training costs modest.

Table 2: Direct spatial-estimation results by source overlap. (a) Ablation of window-selection supervision. (b) Comparison with an internalized spatial baseline; Spatial-Omni-7B has no containment score because it does not select time windows.

		Direct estimation task				Source overlap		
Setting	Containment (%)	Azimuth	Elevation	Distance	Overall	None	Partial	Heavy
(a) Window-supervision ablation								
Full recording	43.07	64.32	70.51	79.22	71.35	82.02	73.22	50.97
High-energy window	54.54	62.67	**70.77**	81.69	71.71	**82.30**	70.81	52.85
Target-informative window	**72.64**	**64.84**	69.99	**82.10**	**72.31**	79.65	**73.63**	**58.32**
(b) Model comparison								
Spatial-Omni-7B	–	**71.44**	**77.19**	**85.39**	**78.01**	**92.16**	77.35	52.53
SONATINE 7B (SFT)	79.98	68.66	71.47	82.30	74.14	80.98	**78.13**	59.84
SONATINE 7B (SFT+GRPO)	**80.62**	70.23	73.55	82.30	75.36	82.27	77.89	**61.60**

6 Analysis

6.1 Validating Adaptive Window Selection

SONATINE-15k follows the principle of adaptive window selection. We test whether this principle provides useful supervision by training three policies on size-matched direct-estimation trajectories that differ in window supervision: full-recording, high-energy, or target-informative windows. We focus on azimuth, elevation, and distance because they directly show the effect of window selection on spatial measurements and underpin more complex reasoning tasks. Because adaptive window selection aims to reduce interference from concurrent sources, we also test its effect across source-overlap levels. We define a question's source-overlap ratio as the fraction of the queried source's active time that overlaps with at least one other source and group questions into *None* (< 0.05), *Partial* (0.05–0.5), and *Heavy* (≥ 0.5).

Table 2(a) shows that target-informative supervision achieves the highest overall accuracy. Its gains are largest under heavy overlap: 7.35 and 5.47 points over full-recording and high-energy supervision, respectively. This pattern matches the intended role of adaptive window selection: window choice matters most when competing sources would otherwise strongly affect the tool measurement. To test whether SFT teaches the intended behavior, we measure first-call containment: the percentage of the first tool-call window within a target-informative reference window for the queried source (Appendix F). We use the initial call because it precedes any spatial observation, whereas later calls may gather complementary evidence for verification or consensus. Target-informative supervision improves containment by 29.57 points over full-recording supervision and by 18.10 points over high-energy supervision, confirming that it induces adaptive window-selection behavior in the learned policy.

We next test whether the main model trained on the full SONATINE-15k dataset gains this benefit over the released Spatial-Omni-7B checkpoint. Table 2(b) shows first-call containment of 79.98% for SONATINE 7B SFT and 80.62% for SONATINE 7B SFT+GRPO, indicating that both select target-informative windows consistently. Although SONATINE 7B SFT+GRPO trails Spatial-Omni-7B by 2.65 points overall on direct spatial estimation, it leads by 9.07 points under heavy overlap, showing greater robustness to concurrent sources. Together, these results support adaptive window selection as a principle for reliable spatial-tool measurements and show that SONATINE-15k successfully teaches the model to apply it.

6.2 Retention of General Audio Understanding

We next test whether acquiring spatial tool-use behavior causes the Audio LLM to forget its pretrained, non-spatial audio capabilities. In this comparison, we evaluate Spatial-Omni-7B, JAEGER, SONATINE 7B SFT, and SONATINE 7B SFT+GRPO. Because all four models use Qwen2.5-Omni-7B as their backbone, we compare them against the unadapted Qwen2.5-Omni-7B model under the same MMAU and MMAU-Pro evaluation protocol (Appendix E.4).

Table 3: Cross-format evaluation on six-channel EasyCom recordings.

Method	Single-speaker azimuth MAE (°) ↓	Left–right accuracy (%) ↑	Order-conditioned azimuth MAE (°) ↓	Location-conditioned WER (%) ↓	Single-speaker distance MAE (m) ↓
Qwen2.5-Omni-7B (base)	21.7	50.0	35.7	63.7	1.177
JAEGER	22.8	50.0	32.7	99.9	1.106
Spatial-Omni-7B	85.9	48.6	83.7	78.2	0.387
SONATINE 7B (SFT + GRPO)	**10.1**	**60.0**	**28.5**	**61.6**	**0.329**

Fig. 2 shows that SONATINE retains the strongest general audio understanding among the spatially adapted models. The full results in Appendix E.4 show that both the SFT and SFT+GRPO policies outperform the internalized spatial baselines across every reported sound, music, and speech domain on MMAU and MMAU-Pro. Their similar results further indicate that GRPO introduces little additional forgetting beyond SFT. We hypothesize that this retention follows from preserving the backbone's native input pathway without injecting multichannel spatial features into it and using a compact adaptation set.

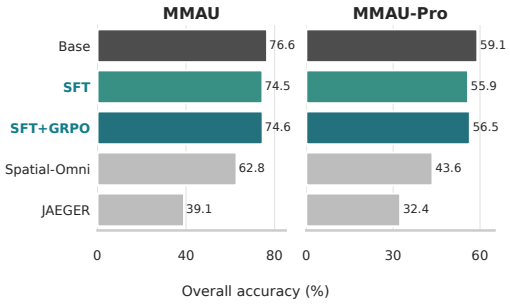


Figure 2: Overall retention on MMAU and MMAU-Pro. Teal denotes SONATINE and gray internalized spatial baselines; higher is better.

6.3 Cross-Format Transfer without Further Training

We then evaluate the transfer of an FOA-trained policy to six-channel microphone-array recordings from EasyCom. The base model receives the front-center channel without spatial tools. Because Spatial-Omni-7B and JAEGER accept only FOA, we provide them with an approximate FOA conversion of the six-channel recordings. For SONATINE, we transfer the FOA-trained SFT+GRPO policy without further training, keep the tool arguments, returned observations, and coordinate convention consistent, and replace only the format-specific tool implementations with a backend that operates directly on the six-channel recording.

Table 3 shows that SONATINE achieves the best result on all five tasks, whereas Spatial-Omni-7B and JAEGER fail to outperform the base model on most tasks. This contrast reflects where spatial measurement resides. Spatial-Omni-7B and JAEGER internalize it in FOA-specific model architectures and learned spatial representations. Adapting them natively to a new recording format therefore requires modifying their input architectures and training new spatial representations; otherwise, the input must be approximately converted to FOA, which could cause the model to undesirably discard some spatial information. SONATINE instead places spatial measurement outside the model weights: fixed signal-processing tools extract spatial cues, while its policy learns when to invoke a tool, which tool to use, and which time window to analyze. This separation allows the learned policy to transfer across recording formats without learning new spatial representations.

7 Conclusion

We introduced SONATINE, a tool-integrated spatial reasoning framework that separates semantic listening from spatial measurement. The Audio LLM receives mono audio through its native interface, while external tools operate on the full multichannel recording and return spatial evidence. To teach adaptive window selection, we constructed SONATINE-15k, a dataset of 14,990 trajectories, and used it for SFT followed by GRPO. However, iterative tool use and reasoning increase inference latency and context length; reducing this overhead remains a direction for future work. Despite this limitation, our experiments demonstrate that SONATINE is both effective and data-efficient, achieving strong spatial performance with substantially fewer spatial QA examples while retaining general audio understanding. Beyond these gains, its externalized design enables prompting-based spatial understanding in closed-source models and transfer from FOA to six-channel recordings without further training. Together, these findings establish external spatial tool use as an effective, data-efficient, and transferable alternative to internalizing spatial measurement within Audio LLMs.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets, implementing methods, assisting with translation, and cleaning and reformatting data sets. We did not use generative AI tools to develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative or thematic data analysis, or interpret results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs were not applicable to this work. Additionally, we used generative AI tools to create or edit software code, summarize or analyze existing literature, search for information, edit the paper for readability, and identify relevant literature. All AI-assisted outputs were reviewed using task-appropriate procedures before inclusion. Synthetic and reformatted data were subjected to programmatic validation and manual inspection; AI-assisted code was reviewed and tested for correctness; suggested literature and factual information were checked against the original sources; and translations and editorial revisions were reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Davide Berghi and Philip JB Jackson. Reverberation-based features for sound event localization and detection with distance estimation. *IEEE Signal Processing Letters*, 2026.

Subrata Biswas, Mohammad Nur Hossain Khan, and Bashima Islam. OWL: geometry-aware spatial reasoning for audio large language models. *arXiv preprint arXiv:2509.26140*, 2025.

Changan Chen, Carl Schissler, Sanchit Garg, Philip Kobernik, Alexander Clegg, Paul Calamia, Dhruv Batra, Philip W. Robinson, and Kristen Grauman. SoundSpaces 2.0: A simulation platform for visual-acoustic learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Symeon Delikaris-Manias and Ville Pulkki. Cross pattern coherence algorithm for spatial filtering applications utilizing microphone arrays. *IEEE Transactions on Audio, Speech, and Language Processing*, 21:2356–2367, 11 2013.

Artem Dementyev, Wazeer Zulfikar, Sinan Hersek, Pascal Getreuer, Anurag Kumar, and Vivek Kumar. PhaseCoder: microphone geometry-agnostic spatial audio understanding for multimodal llms. *arXiv preprint arXiv:2601.21124*, 2026.

Jacob Donley, Vladimir Tourbabin, Jung-Suk Lee, Mark Broyles, Hao Jiang, Jie Shen, Maja Pantic, Vamsi Krishna Ithapu, and Ravish Mehra. EasyCom: an augmented reality dataset to support algorithms for easy communication in noisy environments. *arXiv preprint arXiv:2107.04174*, 2021.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann,

Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Sreyan Ghosh, Arushi Goel, Kaousheik Jayakumar, Lasha Koroshinadze, Nishit Anand, Zhifeng Kong, Siddharth Gururani, Sang gil Lee, Jaehyeon Kim, Aya Aljafari, Chao-Han Huck Yang, Sungwon Kim, Ramani Duraiswami, Dinesh Manocha, Mohammad Shoeybi, Bryan Catanzaro, Ming-Yu Liu, and Wei Ping. Audio Flamingo Next: next-generation open audio-language models for speech, sound, and music. *arXiv preprint arXiv:2604.10905*, 2026.

Google DeepMind. Gemini 3.8 Flash, 2026. URL `https://deepmind.google/models/model-cards/gemini-3-8-flash/`. Accessed: 2026-09-17.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 14953–14962, 2023.

Xilin Jiang, Hannes Gamper, and Sebastian Braun. Sci-phi: a large language model spatial audio descriptor. *IEEE Open Journal of Signal Processing*, 2026.

KimiTeam, Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, Zhengtao Wang, Chu Wei, Yifei Xin, Xinran Xu, Jianwei Yu, Yutao Zhang, Xinyu Zhou, Y. Charles, Jun Chen, Yanru Chen, Yulun Du, Weiran He, Zhenxing Hu, Guokun Lai, Qingcheng Li, Yangyang Liu, Weidong Sun, Jianzhou Wang, Yuzhi Wang, Yuefeng Wu, Yuxin Wu, Dongchao Yang, Hao Yang, Ying Yang, Zhilin Yang, Aoxiong Yin, Ruibin Yuan, Yutong Zhang, and Zaida Zhou. Kimi-Audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

Srđan Kitić and Alexandre Guérin. TRAMP: tracking by a real-time ambisonic-based particle filter. *arXiv preprint arXiv:1810.04080*, 2018.

Sonal Kumar, Simon Sedlácek, Vaibhavi Lokegaonkar, Fernando López, Wenyi Yu, Nishit Anand, Hyeonggon Ryu, Lichang Chen, Maxim Plicka, Miroslav Hlavácek, William Fineas Ellingwood, Sathvik Udupa, Siyuan Hou, Allison Ferner, Sara Barahona, Cecilia Bolaños, Satish Rahi, Laura Herrera-Alarcón, Satvik Dixit, Siddhi Patil, Soham Deshmukh, Lasha Koroshinadze, Yao Liu, Leibny Paola García-Perera, Eleni Zanou, Themos Stafylakis, Joon Son Chung, David Harwath, Chao Zhang, Dinesh Manocha, Alicia Lozano-Diez, Santosh Kesiraju, Sreyan Ghosh, and Ramani Duraiswami. MMAU-Pro: A challenging and comprehensive benchmark for holistic evaluation of audio general intelligence. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 22688–22697, 2026.

H. Kuttruff. *Room Acoustics, Fifth Edition*. Taylor & Francis, 2009. ISBN 9780203876374.

Kuan-Yi Lee, Tsung-En Lin, and Hung-Yi Lee. Audio-Maestro: enhancing large audio-language models with tool-augmented reasoning. *arXiv preprint arXiv:2510.11454*, 2025.

Zhan Liu, Changli Tang, Yuxin Wang, Zhiyuan Zhu, Youjun Chen, Yiwen Shao, Tianzi Wang, Lei Ke, Zengrui Jin, and Chao Zhang. JAEGER: joint 3D audio-visual grounding and reasoning in simulated physical environments. *arXiv preprint arXiv:2602.18527*, 2026.

Yan-Chen Lu and Martin Cooke. Binaural estimation of sound source distance via the direct-to-reverberant energy ratio for static and moving sources. *IEEE Trans. Speech Audio Process.*, 18 (7):1793–1805, 2010.

Kaiwen Luo, Zhenhong Zhou, Leyan Wang, Liang Lin, Tianyu Shao, Yuanhe Zhang, Yang Xiao, Yuxuan Li, Miao Yu, Kailin Lyu, Jiaming Zhang, Li Sun, Songze Li, Yueming Wu, Ting Dang, Xiaojun Jia, Dongrui Liu, Kai Li, Rohan Kumar Das, Siyuan Liang, Xinfeng Li, Qiankun Li, Jing Chen, Xingjun Ma, Kun Wang, Junhao Dong, Deqing Zou, Yu Cheng, Xia Hu, Zhigang Zeng, Sen Su, Yang Liu, Yu-Gang Jiang, Philip S. Yu, and Yew-Soon Ong. A survey of large audio language models: Generalization, trustworthiness, and outlook. *arXiv preprint arXiv:2605.20266*, 2026.

OpenAI. GPT-Audio-1.5, 2026. URL `https://developers.openai.com/api/docs/models/gpt-audio-1.5`. Accessed: 2026-09-23.

Archontis Politis, Tapani Pihlajamäki, and Ville Pulkki. Parametric spatial audio effects. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12)*, 2012.

Ville Pulkki, Archontis Politis, Mikko-Ville Laitinen, Juha Vilkamo, and Jukka Ahonen. First-order directional audio coding (DirAC). *Parametric Time-Frequency Domain Spatial Audio*, pp. 89–140, 2017.

Qwen. Qwen3.6-27B: flagship-level coding in a 27B dense model, 2026a. URL `https://qwen.ai/blog?id=qwen3.6-27b`. Accessed: 2026-09-19.

Qwen. Qwen3.8-27B, 2026b. URL `https://huggingface.co/Qwen/Qwen3.8-27B`. Accessed: 2026-09-25.

S Sakshi, Vaibhavi Lokegaonkar, Neil Zhang, Ramani Duraiswami, Sreyan Ghosh, Dinesh Manocha, and Lie Lu. SPUR: a plug-and-play framework for integrating spatial audio understanding and reasoning into large audio-language models. *arXiv preprint arXiv:2511.06606*, 2025a.

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. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025b.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Christian Schörkhuber, Markus Zaunschirm, and Robert Höldrich. Binaural rendering of ambisonic signals via magnitude least squares. In *Proceedings of the DAGA*, volume 44, pp. 339–342, 2018.

Andreas Schwarz and Walter Kellermann. Coherent-to-diffuse power ratio estimation for dereverberation. *IEEE ACM Trans. Audio Speech Lang. Process.*, 23(6):1006–1018, 2015.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Arvind Krishna Sridhar, Yinyi Guo, and Erik Visser. Spatial audio motion understanding and reasoning. *arXiv preprint arXiv:2509.14666*, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: visual inference via Python execution for reasoning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11854–11864, 2023.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Jun Zhang, Lu Lu, Zejun Ma, Yuxuan Wang, and Chao Zhang. Can large language models understand spatial audio? In *25th Annual Conference of the International Speech Communication Association, Interspeech 2024, Kos, Greece, September 1-5, 2024*, 2024.

Oliver Thiergart, Giovanni Del Galdo, and Emanuël AP Habets. On the spatial coherence in mixed sound fields and its application to signal-to-diffuse ratio estimation. *The Journal of the Acoustical Society of America*, 132(4):2337–2346, 2012.

Siqian Tong, Xuan Li, Yiwei Wang, Baolong Bi, Yujun Cai, Shenghua Liu, Yuchen He, and Chengpeng Hao. AuTAgent: a reinforcement learning framework for tool-augmented audio reasoning. *arXiv preprint arXiv:2602.13685*, 2026.

Yucheng Wang, Jing Peng, Hanqi Li, Chenghao Wang, Wenming Tu, Yu Xi, Zhaokai Sun, Kai Yu, and Shuai Wang. Audio-Mind: an auditable agentic framework for audio understanding. *arXiv preprint arXiv:2605.28480*, 2026a.

Yuwen Wang, Tian-Hao Zhang, Minghao Cai, Yilin Ren, Ziyang Jiang, Xin Wang, Zhichao Wang, Zhou Pan, Kun Zhan, and Xinyuan Qian. Hear, invoke, and understand: A skill-calling multimodal agent for large audio language models. *arXiv preprint arXiv:2608.01881*, 2026b.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*, 2025a.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Yuhuan You, Lai Wei, Xihong Wu, and Tianshu Qu. The world is not mono: Enabling spatial understanding in large audio-language models. *arXiv preprint arXiv:2601.02954*, 2026.

Zhisheng Zheng, Puyuan Peng, Ziyang Ma, Xie Chen, Eunsol Choi, and David Harwath. BAT: learning to reason about spatial sounds with large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61454–61469, 2024.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: incentivizing "thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

Zhiyuan Zhu, Yixuan Chen, Yiwen Shao, Wenxiang Guo, Changhao Pan, Yu Zhang, Yuxiang Wang, Wei Liu, Houhua Zhang, Chengkuan Zeng, Wenbo Cheng, Yunxi Liu, Rui Yang, Steve Yves, Liefeng Bo, and Zhou Zhao. Spatial-omni: Spatial audio understanding integration in multimodal LLMs via FOA encoding. *arXiv preprint arXiv:2606.10738*, 2026.

A Case Study

Fig. 3 presents a complete evaluation trajectory from SONATINE-7B SFT+GRPO for azimuth estimation in a crowded acoustic scene. The queried clock is active from 0.004 to 19.05 seconds, while cooking begins at 2.415 seconds and two additional sounds occur later in the recording. Consequently, the clock overlaps with another source for 87.3% of its active duration, but has a short target-only prefix at the beginning of the recording. The model selects 0.0–2.0 seconds for its first `estimate_direction` call, obtaining an azimuth estimate of $56.2°$. Because the observation has a high diffuseness score, the model requests a second measurement over the nested 0.0–1.5 second window. The second estimate of $57.7°$ agrees with the first, and the model returns $56.2°$, close to the $58.4°$ ground truth.

B Datasets, Benchmark, and Trajectory Construction

B.1 SO-Dataset, SO-Bench, and Shared Task Taxonomy

SO-Dataset and SO-Bench share the same 16 task types. SO-Dataset provides the training-side questions and recordings from which we construct SONATINE-15k and sample the GRPO questions, whereas SO-Bench provides the held-out evaluation questions. Consequently, we define the shared task taxonomy and the target quantity or decision associated with each task once.

Table 4 lists the 16 task types and shows how they map to the four capability categories used in the main results. It also reports the number of trajectories in SONATINE-15k and the number of evaluation questions in SO-Bench for each task.

B.2 SFT Trajectory Construction

We use source-activity annotations only during data construction to form reference time windows for target-conditioned tool calls. For each such call, we first set a target window duration from a

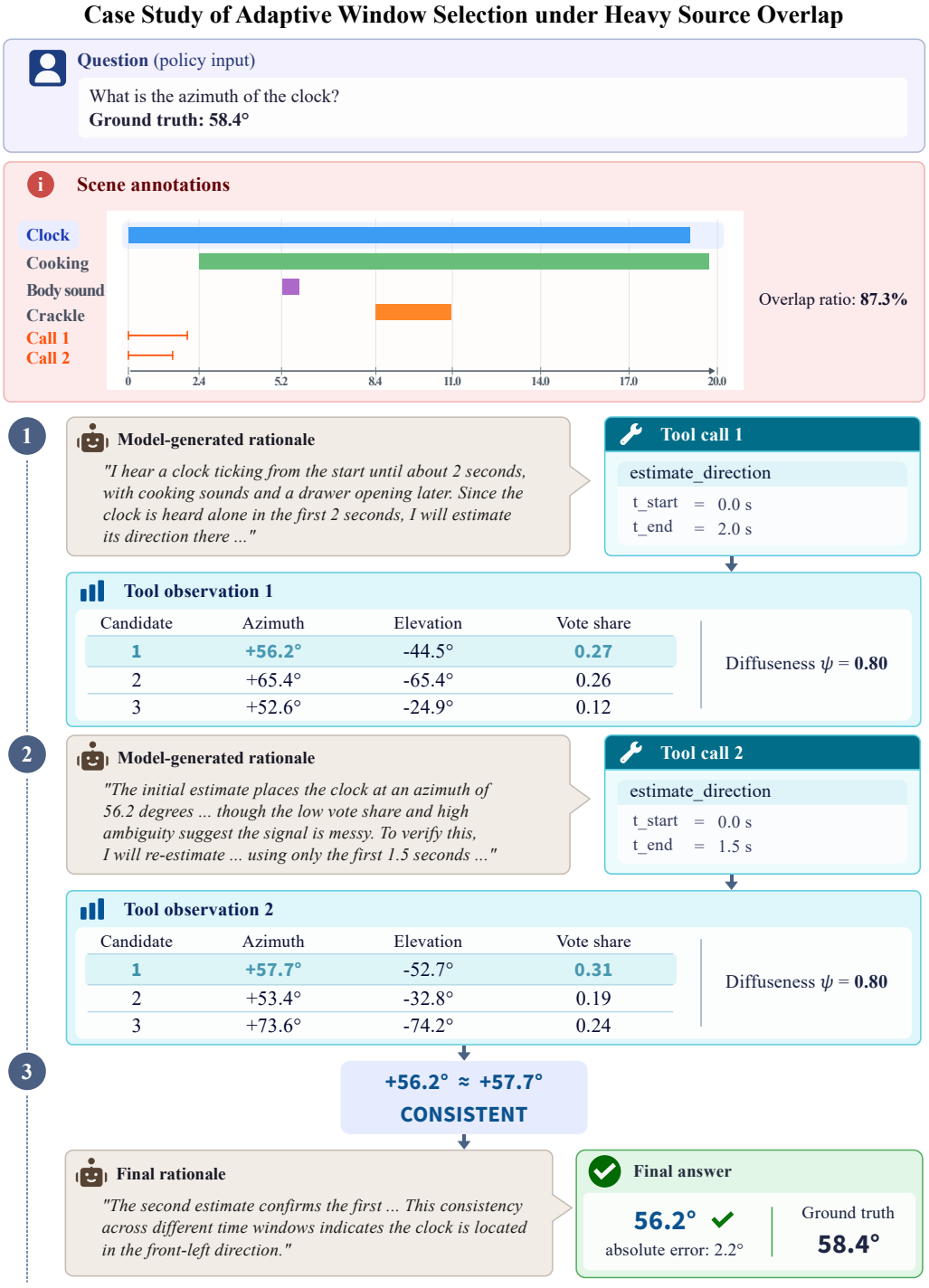


Figure 3: **Case study of adaptive window selection under heavy source overlap.** SONATINE-7B SFT+GRPO selects the queried clock's short target-only prefix and verifies the result with a second, nested window before answering.

Table 4: Shared task taxonomy of SO-Dataset and SO-Bench (Zhu et al., 2026). SONATINE-15k is constructed from the SO-Dataset training split and covers the same 16 task types as the held-out SO-Bench evaluation. The section headers show the four capability categories used for aggregation in Table 1.

Task type	Prediction target	SONATINE-15k trajectories	SO-Bench questions
Non-Spatial Audio Perception			
`detect_source`	Sound events and their active intervals	1,000	667
`detect_time`	Active interval of a queried source	1,000	667
`count_sources`	Number of sound sources	1,000	34
Direct Spatial Estimation			
`estimate_azimuth`	Azimuth of a queried source	1,000	1,152
`estimate_elevation`	Elevation of a queried source	1,000	1,153
`estimate_distance`	Distance to a queried source	977	486
Spatial Grounding and Relation Understanding			
`identify_source_by_doa`	Source at a specified azimuth and elevation	1,000	667
`identify_source_by_location`	Source at a verbal spatial location	1,000	666
`relative_left_right`	Left–right relation between two sources	1,000	333
`compare_elevation`	Relative elevation of two sources	1,000	333
`compare_distance`	Relative distance of two sources	995	333
`onset_from_location`	Onset time of a source at a specified location	1,000	487
Complex Spatial-Semantic Reasoning			
`classify_motion`	Motion pattern of a queried source	1,000	62
`multi_hop`	Answer requiring multiple spatial relations	1,000	333
`spatial_temporal`	Sources and their spatial evolution over time	1,000	333
`speech_content`	Speech from a specified direction	18	171
Total		14,990	7,877

predefined range and then search for its placement within the target's active spans. If the target has an exclusive stretch, we prioritize placing the window within that stretch. Otherwise, among target-active placements of the target duration (or the longest duration supported by a shorter active span), we select the one with the lowest average number of simultaneously active non-target sources. Choosing the duration before placement prevents the construction from obtaining trivially low overlap through arbitrarily short fragments and gives the spatial tools a sufficiently long and broadly comparable amount of signal. The duration is varied across examples within the predefined range so that the policy is not trained to reproduce one fixed window length. We refer to this duration-constrained search as the adaptive placement rule. Because it fixes the duration before minimizing overlap, it is not identical to first finding the globally least-overlapped, uncut stretch of target activity. Nevertheless, the resulting windows closely approximate that independently defined reference: on the direct-estimation questions for which the construction returns a window, mean containment is 0.9768; Appendix F.1 provides the definition and analysis.

Window construction depends on how each task grounds its target. Target-conditioned direction, distance, and motion tasks (`estimate_azimuth`, `estimate_elevation`, `estimate_distance`, and `classify_motion`) use the adaptive placement rule above. Pairwise and multi-source tasks (`relative_left_right`, `compare_elevation`, `compare_distance`, and `multi_hop`) apply it independently to each measured source. The two source-identification tasks (`identify_source_by_location` and `identify_source_by_doa`) likewise include tool-use trajectories with adaptively placed windows, in which the rule is applied independently to each candidate source and the resulting direction estimates are compared. For `spatial_temporal`, we respect the interval stated in the question and select its least-crowded subinterval. In addition, direction-grounded tasks (`identify_source_by_location`, `identify_source_by_doa`, `onset_from_location`, and `speech_content`) include trajectories that crop the full recording toward the direction given in the question. Selecting a target-active window from annotations in these cases could reveal temporal information unavailable at inference time, such as the onset requested by the question, whereas the full-recording directional crop avoids this leakage.

Because the construction yields effectively one meaningful least-overlap window for each target, we assign that window to the first tool call. Multi-call trajectories use other target-active windows only in subsequent calls, providing complementary measurements for verification or consensus rather than optimizing a second independent selection target. We execute the corresponding tools and retain trajectories whose observations pass task-specific reliability and consistency checks and whose derived answers agree with the ground truth when directly verifiable. Qwen3.6-27B (Qwen, 2026a) then synthesizes rationales conditioned on the verified tool calls and observations. The resulting SONATINE-15k dataset contains 14,990 trajectories drawn from 14,394 unique FOA recordings across 16 task types: 11,990 contain one or more spatial tool calls, while the remaining 3,000 cover `detect_source`, `detect_time`, and `count_sources`, which do not require spatial capability and therefore contain no tool calls.

B.3 Example SFT Trajectory

The following example from the `relative_left_right` task shows how an SFT trajectory interleaves rationales, tool calls, observations, and a final answer. The policy selects a target-informative window for each source.

```
User:
Is crackle to the left or right of singing?

Assistant:
I hear a crackle early on, followed by singing, and the question asks to compare their
     positions. To get a clear estimate for the crackle, I will analyze the time window from
     0.8s to 3.38s where it plays alone, before the singing starts.
<tool_call>
{"name": "estimate_direction", "arguments": {"t_start": 0.8, "t_end": 3.38}}
</tool_call>

Tool:
{"candidates": [{"azimuth_deg": 116.4, "elevation_deg": -42.2, "vote_share": 0.82}, {"
     azimuth_deg": -82.7, "elevation_deg": 4.2, "vote_share": 0.09}, {"azimuth_deg": -165.5, "
     elevation_deg": -6.1, "vote_share": 0.06}], "psi": 0.01}

Assistant:
The crackle is clearly located back-left and below, so now I need to locate the singing. I
     will analyze the segment from roughly 4.45s to 6.22s, where the singing is active and the
      crackle has already stopped, ensuring a clean measurement.
<tool_call>
{"name": "estimate_direction", "arguments": {"t_start": 4.45, "t_end": 6.22}}
</tool_call>

Tool:
{"candidates": [{"azimuth_deg": -94.0, "elevation_deg": 21.2, "vote_share": 0.98}, {"
     azimuth_deg": -102.7, "elevation_deg": 19.1, "vote_share": 0.10}, {"azimuth_deg": 113.7,
     "elevation_deg": -38.8, "vote_share": 0.01}], "psi": 0.02}

Assistant:
The crackle is located at an azimuth of 116.4 degrees, which places it to the back-left, while
      the singing is at -94.0 degrees, placing it to the back-right. Since positive azimuth
     indicates left and negative indicates right, the crackle is clearly to the left of the
     singing.
<answer>The crackle is to the left of the singing.</answer>
```

B.4 GRPO Training Questions

We sample the GRPO training questions from the SO-Dataset training split using task-specific stratified sampling to reduce label imbalance during optimization. We train each policy for at most 200 optimization steps or 24 hours, with 32 questions sampled per step. Under this common stopping rule, the 7B run reaches 200 steps and uses 6,400 question instances, whereas the 30B run reaches the 24-hour limit after 150 steps and uses 4,800 question instances.

C SPATIAL TOOL INTERFACE AND BACKENDS

C.1 FULL TOOL SCHEMA

The release interface exposes four tools as JSON function schemas. The same schema is stored in every SFT example, imported when constructing the GRPO data, and checked for exact equality before training. It is also shared by the FOA and six-channel backends; changing the recording format replaces only the implementation behind the interface.

A tool call is a JSON object of the form `{"name": name, "arguments": args}` enclosed in `<tool_call>` tags. All arguments are JSON numbers, and the common required arguments `t_start` and `t_end` denote seconds from the beginning of the recording. The four function signatures and exact model-facing observations are as follows.

`estimate_direction`. This tool returns `{"candidates": [...], "psi": p}`. The candidate list contains at most three entries, ordered by peak strength; each entry contains `azimuth_deg`, `elevation_deg`, and `vote_share`. Azimuth is measured from the front in $[-180, 180]$ degrees, with positive values to the left; elevation is measured from the horizontal plane in $[-90, 90]$ degrees, with positive values upward; and `vote_share` measures the fraction of total directional evidence within the candidate's local refinement neighborhood. The diffuseness score `psi` lies in $[0, 1]$, with larger values indicating more diffuse evidence.

`spatial_crop`. This tool uses `azimuth_deg` and `elevation_deg` to specify the direction to be extracted and returns `{"energy_ratio_db": r, "cropped_audio": "<attached>"}` together with a mono audio attachment. Here `energy_ratio_db` is the cropped signal energy relative to the reference channel over the selected window.

`get_trajectory`. This tool returns `{"frames": [...]}`, where each frame contains an integer `frame` index and the top candidate's `azimuth_deg`, `elevation_deg`, and `vote_share`.

`get_distance`. This tool returns `{"distance_m": d}`, the estimated distance in meters of the dominant source in the selected window.

C.2 DETAILED TOOL IMPLEMENTATIONS

Direction estimation. Following the spherical evidence aggregation approach of Kitić & Guérin (2018), `estimate_direction` computes a Hann STFT with a 256-sample window and 128-sample hop and uses all frequency bins. For each time–frequency bin (f, t), it uses the PIV $\mathbf{I}_{f,t}$ defined in Sec. 3.2 and computes the bin energy

$$E_{f,t} = \frac{1}{2}\left(|W_{f,t}|^2 + \|\mathbf{V}_{f,t}\|^2\right). \tag{4}$$

The direction associated with the PIV is

$$\phi_{f,t} = \operatorname{atan2}(I^{(y)}_{f,t}, I^{(x)}_{f,t}), \qquad \theta_{f,t} = \operatorname{atan2}\left(I^{(z)}_{f,t}, \sqrt{(I^{(x)}_{f,t})^2 + (I^{(y)}_{f,t})^2}\right), \tag{5}$$

where `atan2` is the quadrant-aware arctangent. To emphasize direct-sound onsets, we first define the largest energy in the preceding four frames at the same frequency,

$$E^{\text{prev}}_{f,t} = \max_{j \in \{1,2,3,4\}} E_{f,t-j}. \tag{6}$$

Each bin then receives weight

$$w_{f,t} = \operatorname{clip}\left(\frac{1}{4}\log_2 \frac{E_{f,t} + \epsilon}{E^{\text{prev}}_{f,t} + \epsilon}, 0, 1\right), \qquad \epsilon = 10^{-20}, \tag{7}$$

where $E_{f,t}$ is the energy of the current (f, t) bin; if all weights vanish, the implementation falls back to uniform votes. The weighted directions are accumulated in a $5^\circ \times 5^\circ$ spherical histogram and

smoothed with a 3×3 uniform filter. The backend returns up to three peaks in descending order, suppressing a 25° neighborhood after each selection and refining each peak by the weighted mean direction within 20°. Candidate `vote_share` is the refined peak mass divided by the total vote weight. The reported diffuseness is $\psi = 1 - \|\sum_{f,t} \mathbf{I}_{f,t}\| / \sum_{f,t} E_{f,t}$.

Trajectory estimation. `get_trajectory` repeatedly applies the same direction estimator at successive points within the requested window and returns the strongest candidate at each step.

Spatial cropping. `spatial_crop` combines a steered hypercardioid beam with a CroPaC post-filter (Delikaris-Manias & Pulkki, 2013) and a direction-dependent time–frequency mask (Politis et al., 2012). It uses a reconstruction-consistent Hann STFT with 512-sample windows and a 256-sample hop. For requested azimuth a and elevation e, let $\mathbf{u} = [\cos a \cos e,\ \sin a \cos e,\ \sin e]^\top$ be the corresponding target-direction unit vector. The static beam at bin (f, t) is $B_{f,t} = 0.25 W_{f,t} + 0.75 \mathbf{u}^\top \mathbf{V}_{f,t}$. The CroPaC gain is

$$g_{f,t} = \mathrm{clip}\left(\frac{2\mathbf{u}^\top \mathbf{I}_{f,t}}{|W_{f,t}|^2 + \|\mathbf{V}_{f,t}\|^2}, 0, 1 \right), \tag{8}$$

where the numerator and denominator are separately smoothed with a 3×3 uniform filter. The PIV mask is $m_{f,t} = \max(\mathbf{u}^\top \mathbf{d}_{f,t}, 0)^2$, where $\mathbf{d}_{f,t}$ is the unit direction obtained from $\mathbf{I}_{f,t}$. The returned crop is the inverse STFT of $g_{f,t} m_{f,t} B_{f,t}$, peak-normalized to 0.9. Before peak normalization, the backend also computes `energy_ratio_db` as $10 \log_{10}$ of the cropped-signal energy divided by the W-channel energy over the same window.

Learned distance regressor. Unlike the three DSP tools above, the FOA distance backend is a 38.72M-parameter CNN–Conformer regressor adapted from Berghi & Jackson (2026). The regressor contains four residual CNN blocks with 64, 128, 256, and 512 channels, respectively; each block uses two 3×3 convolutions, batch normalization, ReLU activations, and average pooling. The CNN reduces time by a factor of four and frequency by a factor of 16. Its output is projected to 512 dimensions and passed through an eight-layer, eight-head Conformer with feed-forward dimension 1,024 and convolution kernel size 31. A $512 \to 128 \to 1$ head with an ELU activation produces frame-level distance estimates, whose mean over valid frames is returned as the window-level prediction.

The regressor is trained on 199,250 single-source windows from the SO-Dataset training split. Training and test rooms are disjoint, and a room-disjoint internal validation split is used for checkpoint selection. Training runs for 30 epochs with batch size 32 and Adam at an initial learning rate of 5×10^{-5}. The learning rate is held constant for the first 60% of epochs and then multiplied by 0.95 per epoch. The objective is the sum of frame-level and window-level mean squared error, and the checkpoint with the highest validation accuracy within 1 m is selected.

C.3 Six-Channel Array Backend for EasyCom Recordings

SONATINE applies six-channel implementations of its spatial tools directly to the original array recordings.

Array calibration and model input. EasyCom (Donley et al., 2021) provides synchronized six-channel, 48-kHz recordings from microphones mounted on the augmented-reality glasses, together with measured ATFs for the same device. We use the ATF release, which contains 768-sample impulse responses for 1,020 directions and six microphones. Because explicit microphone coordinates are not provided, the ATFs themselves encode the array geometry. All array tools use a Hann STFT with 768-sample windows and a 384-sample hop, giving the same 385 frequency bins as the ATFs without interpolation. The Audio LLM receives the front-center frame microphone (channel 2), resampled to 16 kHz, while the tools access all six channels.

Direction and trajectory. Let $\mathbf{x}_{f,t} \in \mathbb{C}^6$ be an observed time–frequency bin and $\mathbf{v}_{q,f}$ the unit-normalized ATF steering vector for grid direction q. The array backend computes the normalized matched-filter score

$$c_{q,f,t} = \frac{|\mathbf{v}_{q,f}^H \mathbf{x}_{f,t}|^2}{\|\mathbf{x}_{f,t}\|^2}, \tag{9}$$

and assigns the bin to $\arg\max_q c_{q,f,t}$. The resulting direction receives the same onset weight as in the FOA backend, multiplied by its maximum match score, and is passed to the same spherical histogram and peak-refinement procedure. Here ψ is one minus the weighted mean maximum match score, so a larger value indicates that no measured steering direction explains the observation well. Trajectory estimation simply invokes this direction estimator repeatedly over the requested window.

Spatial cropping. For the ATF direction nearest the requested azimuth and elevation, the backend forms a matched-filter beam $\mathbf{v}_{q,f}^H \mathbf{x}_{f,t}$ and scales it to the response level of the front-center microphone. Let $\mathbf{u}_q \in \mathbb{R}^3$ be the unit vector for this target grid direction and $\widehat{\mathbf{u}}_{f,t}$ the unit vector for the best-matching direction of bin (f, t). The target-direction score $c_{q,f,t}$ is smoothed with a 3×3 uniform filter to obtain a coherence gain, which is combined with the directional mask $\max(\widehat{\mathbf{u}}_{f,t}^\top \mathbf{u}_q, 0)^2$. The inverse-STFT signal is returned as mono 16-kHz audio, and its pre-normalization energy is compared with that of the front-center microphone for `energy_ratio_db`.

Training-free distance estimation. Unlike the learned FOA distance regressor, the EasyCom backend has no learned parameters. It first estimates the dominant direction and removes the response expected from an isotropic diffuse field from that direction's matched-filter score. This use of DRR as a distance cue follows Lu & Cooke (2010), while estimating it from a directional-plus-diffuse sound-field model follows Thiergart et al. (2012) and Schwarz & Kellermann (2015). Specifically, the diffuse floor at frequency f is the spherical-quadrature average of $|\mathbf{v}_{q,f}^H \mathbf{v}_{r,f}|^2$ over all measured directions r; after subtracting this floor, the clipped direct fraction is energy-averaged over bins that can distinguish directions. Following the critical-distance relation described by Kuttruff (2009), for mean direct fraction $\bar{r}$ we set $\mathrm{DRR} = \bar{r}/(1 - \bar{r})$ and estimate

$$d = \frac{d_c}{\sqrt{\mathrm{DRR}}}, \qquad d_c = 0.057\sqrt{\frac{V}{T_{60}}}. \tag{10}$$

We use the room dimensions and reverberation time reported with EasyCom, $V = 6 \times 7 \times 3 = 126$ m^3 and $T_{60} = 0.645$ s, giving $d_c = 0.797$ m (Donley et al., 2021). These are room properties rather than test answers, and no EasyCom labels are used to fit the estimator.

D SONATINE Training, Prompting, and Inference Details

D.1 Policy Instantiation and Prompts

The closed-source prompting policies use the pretrained Audio LLMs without parameter updates. Fig. 4 reproduces the complete system message passed to the model. The `Choosing the time window of a tool call` instruction induces adaptive window selection by directing the model toward windows in which the target is active with minimal overlap from other sources. The initial user message contains the monaural W-channel audio item followed by the fixed text below and the SO-Bench question inserted verbatim in place of the bracketed description:

```
Use tools only when they are necessary to answer the question reliably.
Write reasoning before tool calls and before the final answer. Put the final answer in <answer
    >...</answer>.
Question: [verbatim SO-Bench question]
```

Tool observations are returned through the providers' native function-calling interfaces.

```
You are a helpful assistant that answers questions about the given audio recording.

# Spatial conventions
The recording is a spatial (first-order Ambisonics) recording, but the audio you hear is a single-channel mix
    of it, so your own hearing carries no information about direction or distance. The spatial tools below
    read the original multichannel recording and are the only reliable source for where a sound is.
    Therefore, if a question requires any spatial information, such as direction, distance, or moving
    trajectory, you MUST USE a spatial tool. Azimuth is in [-180, 180] degrees: 0 = front, positive = left,
    negative = right, +-180 = behind. Elevation is in [-90, 90] degrees: positive = above the listener,
    negative = below. Distances are in meters. All times are in seconds from the start of the recording.

# Choosing the time window of a tool call
When you call a tool, prefer as t_start and t_end a part of the recording in which the target sound source is
    sounding and overlaps as little as possible with the other sound sources.

# Answer format
Think step by step in plain text. Every message must end either with exactly one tool call or with the final
    answer inside <answer>...</answer>, never both. Do not use markdown. The final answer is one short
    sentence that directly answers the question, with the number, label or name asked for.
```

Figure 4: Complete system prompt used for the closed-source prompting evaluation, reproduced verbatim from the saved evaluation prompt. Tool schemas are supplied separately through each provider's native function-calling API.

D.2 Supervised Fine-Tuning

We train both Qwen2.5-Omni-7B and Qwen3-Omni-30B-A3B-Instruct on all 14,990 SONATINE-15k trajectories for two epochs. Each trajectory contains the initial audio and question, the assistant rationales and tool calls, the resulting tool observations (including any audio returned by `spatial_crop`), and the final rationale and answer. We minimize next-token cross-entropy on every assistant turn: the supervised tokens therefore include the rationales, serialized tool calls, and final answer, whereas the system and user messages, tool observations, and input or returned-audio placeholders are masked from the loss.

For Qwen2.5-Omni-7B, we perform full-parameter fine-tuning of the thinker language model while freezing the audio encoder and its projector. We use AdamW with a learning rate of 1×10^{-5}, 5% linear warmup followed by cosine decay, and a global batch size of 64.

For Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025b), we instead use LoRA with rank 16, alpha 32, and dropout 0.05. Adapters are applied to the thinker self-attention query, key, value, and output projections, every expert's gate, up, and down projections, and the language-model head; the router and all other parameters remain frozen. We use AdamW with a learning rate of 1×10^{-4}, the same warmup and cosine schedule, and a global batch size of 64. The learned adapters are merged into the base weights for inference.

D.3 GRPO Optimization and Rewards

Starting from the two-epoch SFT checkpoints, we optimize both policies with GRPO (Shao et al., 2024), subject to a common stopping rule of at most 200 optimization steps or 24 hours of wall-clock training. Each step samples 32 questions and generates a group of $G = 8$ executable-tool rollouts per question, for an effective batch of 256 completions. We perform a single policy update on each freshly generated rollout batch and do not reuse rollouts across updates, yielding strict on-policy optimization. Each rollout contains at most seven assistant–tool turns, a 768-token completion, and a 10,240-token total context. We sample with temperature 1.0, top-$p = 0.9$, and top-$k = 50$.

For Qwen2.5-Omni-7B, training on four H100 GPUs reaches the 200-step cap within 24 hours; the reported checkpoint therefore uses 6,400 question instances. We update all thinker-language-model parameters while keeping the audio encoder and projector frozen. The learning rate is linearly warmed to 1×10^{-6} over the first 45 steps and then held constant. For Qwen3-Omni-30B-A3B-Instruct, training on eight H100 GPUs reaches the 24-hour limit after 150 steps; the reported checkpoint therefore uses 4,800 question instances. We continue optimizing the rank-16 LoRA adapters learned during SFT while keeping the base parameters, router, audio encoder, and projector frozen, using a learning rate of 1×10^{-5} with linear warmup over the first 45 steps followed by a constant schedule. For both models, the GRPO KL coefficient is zero.

The reward is computed from the final answer. The scorer reads the last `<answer>...</answer>` span and requires it to be nonempty. A missing answer tag, an unparseable answer, or a scoring exception receives zero, and every returned reward is clipped to $[0, 1]$. Gold source and time annotations are used only to compute the task outcome for tasks that require them; they are neither exposed to the policy nor used as action-level supervision. Table 5 separates these training rewards from the metrics used for final evaluation. For numeric tasks, let

$$\rho(e; f, \tau) = \frac{1}{1 + \max(e - f, 0)/\tau}, \tag{11}$$

where e is the task-specific absolute error, f is a zero-penalty floor, and τ controls reward decay beyond that floor.

Table 5: Task-specific GRPO rewards and SO-Bench evaluation metrics. All GRPO rewards lie in $[0, 1]$, and an unparseable answer receives zero. Relation and label tasks use an LLM only to extract the answer claim; correctness is computed against the gold label outside the judge.

Task type	GRPO reward	SO-Bench evaluation metric
Non-Spatial Audio Perception		
`detect_source`	$0.5F_{\text{class}} + 0.5F_{\text{soft}}$, where F_{soft} weights class-matched events by temporal IoU.	Event F1; a prediction is a true positive only when its class matches and temporal IoU is at least 0.5.
`detect_time`	Temporal IoU between the first predicted window and the reference window.	Temporal IoU under the same first-window convention.
`count_sources`	Exact-match reward in $\{0, 1\}$ for the parsed source count.	Exact source-count accuracy.
Direct Spatial Estimation		
`estimate_azimuth`	$\rho(e; 5^\circ, 7.5^\circ)$ with wrap-aware azimuth error e.	Wrap-aware accuracy within 20°.
`estimate_elevation`	$\rho(e; 5^\circ, 4.5^\circ)$ with absolute elevation error e after clipping the prediction to $[-90^\circ, 90^\circ]$.	Accuracy within 10°.
`estimate_distance`	$\rho(e; 0, 0.5\,\text{m})$ with absolute distance error e in meters.	Accuracy within $1\,\text{m}$.
Spatial Grounding and Relation Understanding		
`identify_source_by_doa`	Binary correctness of the extracted source label.	Source-label accuracy using LLM-based answer extraction.
`identify_source_by_location`	Binary correctness of the extracted source label.	Source-label accuracy using LLM-based answer extraction.
`relative_left_right`	Binary correctness of the extracted left–right relation.	Left–right relation accuracy.
`compare_elevation`	Binary correctness of the extracted higher-source claim.	Relative-elevation accuracy.
`compare_distance`	Binary correctness of the extracted closer-source claim.	Relative-distance accuracy.
`onset_from_location`	$\rho(e; 0.1\,\text{s}, 0.25\,\text{s})$ with absolute onset-time error e.	Accuracy within $0.2\,\text{s}$.
Complex Spatial-Semantic Reasoning		
`classify_motion`	Binary correctness of the extracted motion label.	Motion-label accuracy.
`multi_hop`	Binary correctness of the extracted answer label.	Semantic answer accuracy using LLM-based answer extraction.
`spatial_temporal`	$0.5 \times \mathbb{1}[\text{source match}] + 0.5F_{\text{dir}}$.	$\mathbb{1}[\text{source match}] \times F_{\text{dir}}$.
`speech_content`	$1 - \min(\text{WER}, 1)$ after transcript extraction.	$1 - \min(\text{WER}, 1)$ after transcript extraction.

For free-form relation and label tasks—`classify_motion`, `relative_left_right`, `compare_distance`, `compare_elevation`, `identify_source_by_doa`, `identify_source_by_location`, and `multi_hop`—we use Qwen3-4B-Instruct-2507 (Yang et al., 2025) only as an answer-claim extractor. Deterministic code compares the extracted claim with the gold label.

D.4 Inference and Tool-Execution Protocol

For SO-Bench evaluation, we decode greedily (temperature 0) with at most 768 newly generated tokens per model reply and a 16,384-token context. Each example initially contains the full mono W channel. Audio returned by `spatial_crop` is appended to subsequent turns.

For the open-weight models, each question is limited to eight model replies, with tool-call replies and the final-answer reply both counting toward the limit. Execution terminates when the model emits an `<answer>` span or reaches this limit. If the final allowed reply is a tool call, the prediction is empty and receives zero; if it is otherwise unparsable, its raw text is passed to the standard scorer.

D.5 Closed-Source Tool-Prompting Protocol

We evaluate GPT-Audio 1.5 and Gemini 3.8 Flash on September 23, 2026. The no-tool evaluation follows the same input protocol as the general-purpose open-weight models: a single request contains the mono W-channel audio and the SO-Bench question verbatim, with no system prompt or tools. We decode with temperature 0; for Gemini 3.8 Flash, we additionally set the thinking level to `low`.

For prompting-only SONATINE, we provide the spatial conventions, adaptive-window-selection instructions, answer rules, and four tool schemas, exposing the tools through each provider's native function-calling interface rather than serialized tool-call tags. The API loop permits at most four tool-execution rounds and then requests a final answer, for at most five model replies in total. Only the first function call in a reply is executed; additional calls in the same reply receive an error observation. If a reply is malformed, a repair message asks the model to use native function calling or provide its final answer.

E Evaluation and Scoring Protocol

E.1 Spatial-Omni-7B Checkpoint Provenance and Training-Data Estimate

Our experiments use the `SO-7B_finetuned.pt` checkpoint obtained from Hugging Face revision `6e1adab`, which was the only SO-7B checkpoint available at the time of our evaluation. The Spatial-Omni manuscript was subsequently updated and reports revised results. Our results therefore characterize the checkpoint available at evaluation time and may not match the values reported in the current manuscript.

The Spatial-Omni-7B training-set size is estimated from the released artifacts rather than reported by the authors. The released checkpoint configuration selects the full training split without a sample cap, the data loader reads every record under this setting, and the accompanying launch script describes the split as approximately 787K examples. The exact dataset revision used to train the released checkpoint is not identified.

E.2 BAT Input Preprocessing

BAT (Zheng et al., 2024) expects binaural rather than FOA input. We therefore normalize each SO-Bench FOA recording to SN3D, render it binaurally at 32 kHz using a first-order MagLS decoder (Schörkhuber et al., 2018) and the default RLRAudioPropagation HRTF used in BAT's SoundSpaces training data (Chen et al., 2022), and preserve a common gain across the two ears.

E.3 SO-Bench Scoring Protocol

The shared 16-task taxonomy and the relationship between SO-Dataset and SO-Bench are defined in Appendix B.1. The task-specific SO-Bench metrics are summarized in the evaluation column of Table 5. This section describes the implementation used to parse model outputs and compute the evaluation metrics.

We extend the released SO-Bench scorer to cover the complete 16-task taxonomy and implement the metrics in the evaluation column of Table 5. We do not rely solely on its rule-based parsing because it is sensitive to surface form and can therefore misinterpret free-form responses. Numeric tasks are parsed deterministically and scored with task-specific tolerances; azimuth error is computed modulo 360°, and numbers carrying non-angular units are not interpreted as angles.

For free-form relation, source-label, event, interval, direction, and transcript answers, we use Qwen3.8-27B (Qwen, 2026b) as an extractor. We disable thinking and decode greedily with temperature 0. Given a task-specific prompt containing the question and model answer, the extractor

Table 6: Domain-level retention of general audio understanding on MMAU and MMAU-Pro (accuracy, %; higher is better).

	MMAU				MMAU-Pro			
Method	Sound	Music	Speech	Overall	Sound	Music	Speech	Overall
Base Audio LLM (Qwen2.5-Omni-7B)	84.08	71.56	74.17	76.61	61.60	60.86	54.77	59.08
Spatial-Omni-7B	71.77	56.89	59.76	62.81	38.78	50.14	41.86	43.59
JAEGER	39.04	37.13	41.14	39.10	35.82	30.61	30.86	32.43
SONATINE 7B (SFT)	79.88	68.26	75.38	74.51	53.96	60.93	52.75	55.88
SONATINE 7B (SFT + GRPO)	79.88	68.56	75.38	74.61	54.73	60.79	53.98	56.50

returns a relation, motion, or source label; a list of sound events with source labels and start–end times; a list of time intervals; a source label and direction labels; or a transcript, depending on the task. Deterministic code then compares the extraction with the annotation and computes any F1, IoU, or WER-based score.

Every per-example score is bounded to $[0, 1]$. We first average scores within each task, including failed examples as zeros. Each category score is the unweighted mean of its constituent task scores, and the overall SO-Bench score is the unweighted mean over all 16 task scores.

E.4 General-Audio Retention Protocol

We evaluate every model with the same multiple-choice protocol on the 1,000-question MMAU test split (v05.15.25) (Sakshi et al., 2025b) and the 3,356-question single-audio, closed-choice subset of MMAU-Pro (Kumar et al., 2026). Each model receives the first 20 seconds of the clip as mono audio, without access to spatial tools, and decodes greedily with at most 256 new tokens. Spatial-Omni-7B and JAEGER use their respective mono-compatibility inference paths.

MMAU contains 333 sound, 334 music, and 333 speech questions. For MMAU-Pro, the reported scores use its 1,047 sound, 1,418 music, and 891 speech questions. In particular, spatial-audio questions are excluded because their stereo cues are removed by the common mono input protocol. Overall is the unweighted mean of the three domain accuracies.

Table 6 reports the complete domain-level results.

E.5 EasyCom Benchmark Construction and Cross-Format Protocol

For the EasyCom evaluation, we use the final SONATINE checkpoint after 200 GRPO steps. Its policy is trained only on the SO-Dataset training split; no EasyCom examples are used to update its parameters.

We construct the benchmark from all 12 EasyCom (Donley et al., 2021) sessions. Candidate clips are aligned to the provided utterance boundaries and restricted to 1.5–15 seconds. Each sampled single-speaker clip produces both an azimuth question and a distance question. The left–right and order-conditioned azimuth tasks use consecutive speaking turns whose first onsets are separated by at least 0.3 seconds and whose guest speakers are separated by at least 20°. Location-conditioned transcription requires a competing guest at least 30° from the target and a target utterance of at least four words. Using a fixed seed, we select 70 questions for each of the five tasks, yielding 350 questions in total.

Azimuth and distance labels are derived from the tracked wearer and participant poses after transforming them into the headset coordinate frame and averaging over the target's active frames. Speaking order is determined by first onset, and the provided human transcripts supply the transcription references. We verify the resulting labels through automated consistency checks and manual inspection of sampled clips.

At evaluation time, Qwen2.5-Omni-7B and SONATINE listen to the front-center array microphone resampled to 16 kHz. SONATINE retains the policy trained on SO-Dataset, including its parameters, prompt, tool schemas, tool-result fields, decoding settings, and answer rules; only the tool implementation is replaced by the six-channel backend, which operates on the original 48-kHz ar-

ray signal. For both Spatial-Omni-7B and JAEGER, we convert the six-channel recordings to FOA using the same full-band complex least-squares Ambisonics Signal Matching filters derived from the measured EasyCom array transfer function.

We additionally inspect the unexpectedly high EasyCom azimuth errors of Spatial-Omni-7B in Table 3. Across the 140 predictions from the two azimuth tasks, Spatial-Omni-7B outputs exactly -100° for 94 examples (67.1%), while the mono Qwen2.5-Omni-7B baseline outputs 0° for 117 examples (83.6%). The Pearson correlations between the predicted and target azimuths are 0.03 and -0.22, respectively, indicating that neither model reliably tracks the target direction. Because the EasyCom labels are concentrated near the front, the mono baseline's near-zero predictions happen to incur smaller errors, whereas Spatial-Omni-7B collapses to an off-axis direction.

E.6 Per-Task SO-Bench Results

Following Spatial-Omni (Zhu et al., 2026), we abbreviate the tasks as follows: DS, `detect_source`; DT, `detect_time`; CS, `count_sources`; EAzi, `estimate_azimuth`; EEle, `estimate_elevation`; EDis, `estimate_distance`; IS-DoA, `identify_source_by_doa`; IS-Loc, `identify_source_by_location`; RLR, `relative_left_right`; CEle, `compare_elevation`; CDis, `compare_distance`; OL, `onset_from_location`; CM, `classify_motion`; MH, `multi_hop`; ST, `spatial_temporal`; and SC, `speech_content`. Table 7 organizes these tasks using the same four capability categories as Table 1 and reports the corresponding metrics defined in Table 5.

F Adaptive Window Selection Analysis

F.1 Reference Windows and Containment

For the containment analysis in Sec. 6.1, we independently construct minimum-overlap reference windows from scene annotations. Within the queried source's active time, we find the lowest number of concurrent non-target sources attained by a contiguous stretch and select a single canonical least-overlap stretch as the reference window. Let I denote a model-selected window and R the corresponding reference window. We define containment as

$$\text{Containment}(I, R) = \frac{|I \cap R|}{|I|}, \tag{12}$$

where $|\cdot|$ denotes window duration. The SFT trajectory-window construction and the analysis reference-window construction differ slightly because the former fixes a duration before placement, whereas the latter retains uncut reference stretches. Nevertheless, on the direct-estimation questions for which the SFT construction rule returns a window, those windows achieve a mean containment of 0.9768, compared with 0.4994 for windows of the same durations placed uniformly over the recording.

SONATINE-15k includes both single-call trajectories and multi-call trajectories that aggregate repeated measurements. For a multi-call trajectory, the initial tool call uses the unique preferred least-overlap window, while follow-up calls intentionally use other target-active windows to provide complementary evidence for verification or consensus rather than duplicate the same observation. We therefore evaluate containment only for the first call: it captures temporal selection before any spatial observation and is the only call directly associated with the preferred least-overlap window.

F.2 Window-Supervision Ablation Details

The three policies in Table 2(a) use Qwen2.5-Omni-7B and are trained only with SFT; no GRPO optimization is applied. Each policy is trained for two epochs on 2,977 direct-estimation trajectories, comprising 1,000 azimuth, 1,000 elevation, and 977 distance trajectories. All other optimization settings follow Appendix D.2.

The full-recording condition uses the entire recording for each spatial measurement. The high-energy condition computes mean squared amplitude over the mono W channel in 0.1-s frames and

selects the contiguous window with the highest mean energy. The target-informative condition uses the adaptive placement rule described in Appendix B.2. The high-energy control uses the same per-question window-length budget as the target-informative construction, so the two conditions differ only in where the window is placed.

The evaluation contains 2,791 direct-estimation questions: 1,152 azimuth, 1,153 elevation, and 486 distance questions. The *None*, *Partial*, and *Heavy* overlap buckets contain 1,548, 397, and 846 questions, respectively.

Table 7: Per-task SO-Bench results for the models in Table 1, organized by our four capability categories. All values are percentages and higher is better. Best and second-best evaluated results are shown in bold and underlined, respectively; ties share the same marking.

	Non-Spatial Perception			Direct Spatial Estimation			Grounding & Relations						Complex Spatial-Semantic			
Model	DS	DT	CS	EAzi	EEle	EDis	IS-DoA	IS-Loc	RLR	CEle	CDis	OL	CM	MH	ST	SC
General-purpose Audio LLMs																
GPT-Audio 1.5	1.78	31.77	5.88	8.77	28.97	44.86	0.60	1.35	36.34	39.04	49.55	39.63	27.42	1.20	6.74	9.57
Gemini 3.8 Flash	4.06	53.78	5.88	11.63	22.12	47.33	10.04	1.05	45.95	51.35	49.55	33.06	17.74	0.30	4.46	23.78
Qwen2.5-Omni 7B	3.60	51.94	5.88	9.81	32.96	58.44	17.99	7.96	51.05	51.35	50.45	41.07	20.97	3.00	11.68	17.13
Qwen3-Omni 30B	0.64	55.72	5.88	9.20	33.13	56.17	11.09	3.60	49.25	51.35	53.15	50.51	22.58	0.60	11.63	24.98
Gemma 4 12B	0.00	28.57	26.47	10.94	30.88	62.76	0.60	0.45	48.95	50.75	52.25	38.60	14.52	0.30	0.00	6.22
Kimi-Audio 7B	0.00	14.46	32.35	8.68	9.80	34.98	15.29	7.81	48.05	46.25	49.25	41.89	16.13	3.30	5.17	24.27
Audio Flamingo Next	0.34	26.23	14.71	9.72	25.33	60.08	4.05	2.55	49.85	51.95	49.85	50.92	4.84	1.20	4.52	17.30
Internalized spatial baselines																
BAT	0.00	2.33	5.88	0.00	0.00	55.56	9.30	7.96	42.04	0.90	1.20	1.64	0.00	3.90	13.22	0.00
JAEGER (Audio)	2.55	22.04	5.88	16.32	27.32	58.02	4.95	1.50	33.93	47.75	48.35	23.00	35.48	0.60	3.12	0.26
Spatial-Omni-7B	**63.57**	**83.45**	26.47	**71.44**	**77.19**	**85.39**	**62.67**	**58.11**	51.35	66.37	59.76	**83.37**	45.16	**28.23**	32.17	24.78
SONATINE (closed-source prompting)																
GPT-Audio 1.5	0.06	37.05	20.59	56.94	64.01	79.22	1.80	1.50	42.64	36.34	41.74	34.70	37.10	0.00	14.68	23.92
Gemini 3.8 Flash	6.62	55.13	14.71	68.06	72.85	80.66	5.85	1.80	**64.86**	69.97	56.46	45.17	43.55	1.50	14.15	47.47
SONATINE (tool-use training)																
SONATINE 7B (SFT)	52.24	79.74	55.88	68.66	71.47	82.30	56.07	52.70	63.96	71.17	61.56	80.90	43.55	25.83	**61.17**	38.49
SONATINE 30B (SFT)	49.70	81.65	47.06	70.49	74.85	80.25	53.67	50.60	58.56	70.57	**62.16**	78.44	41.94	25.23	57.59	47.87
SONATINE 7B (SFT + GRPO)	50.23	77.45	**58.82**	70.23	73.55	82.30	55.17	51.95	**64.86**	**73.87**	61.26	79.06	**46.77**	24.32	60.62	39.89
SONATINE 30B (SFT + GRPO)	52.71	81.88	52.94	70.23	74.85	81.89	54.87	50.00	62.76	67.27	61.56	81.93	43.55	23.72	60.01	**52.27**