

SonicWorld: Source-Centric Latent Dynamics for Spatial Audio Reasoning

Chenlong Wei¹ Hanling Wang² Mingjin Lou³ Yufan Shi² Yandong Sun⁴
Sheldon Yu² Xiaohui Zhu¹ Xiaobo Jin¹ Yiwei Wang^{†5}

¹Xi'an Jiaotong-Liverpool University

²University of California, San Diego ³Imperial College London

⁴Lyuliang University ⁵Southeast University

† Corresponding author

Chenlong.Wei23@student.xjtlu.edu.cn haw168@ucsd.edu
wangyw.evan@gmail.com

Abstract: Dynamic spatial audio reasoning requires a model to associate changing acoustic observations with the correct source and recover its spatial state at a queried time or interval, a challenge intensified by overlapping speech and source motion. To study this problem, we introduce **Sonic-Bench**, comprising 2.8K first-order Ambisonics clips and 28.8K target-specific question-answer pairs across 480 simulated indoor scenes. The benchmark covers localization, distance estimation, and motion reasoning in single- and dual-source settings. We further propose **SonicWorld**, which integrates physical, learned acoustic, and semantic features into persistent source states through learned transitions and recurrent updates, and projects them into an Omni LLM for spatial reasoning. Compared with Spatial-Omni on the Sonic-Bench test split, SonicWorld improves direction accuracy from 31.98% to 34.69% and reduces azimuth error from 57.22° to 48.01°. In zero-shot transfer to Spatial LibriSpeech, SonicWorld retains strong localization performance, achieving 73.98% direction accuracy and 19.29° azimuth error, compared with 27.14% and 70.74° for Spatial-Omni. Code is available at: <https://anonymous.4open.science/r/2C38/>.

🌐 Project Page: <https://sonic-world.github.io/>

1. Introduction

Recent Large Audio-Language Models (LALMs) and Omni LLMs have demonstrated growing capabilities in recognizing and reasoning about audio content [1, 2, 3]. However, understanding an auditory scene requires more than identifying what is heard: a model must also determine where each sound originates and how its location changes over time. Spatial audio preserves directional, distance, motion, and environmental cues beyond sound content, providing the perceptual basis for understanding dynamic 3D auditory scenes [4].

Spatial-audio benchmarks have expanded from localization to multi-source relations and motion reasoning [5, 6]. However, answering where a particular source is at a given moment or how it moves over an interval calls for jointly resolving source identity and temporal context. We therefore study *dynamic source-specific spatial reasoning*: given a spatial-audio sequence and a query identifying a source and a timestamp or interval, the model is asked to infer the source’s location, distance, or movement direction. Answering these queries requires linking acoustic evidence to the same source across time, yet most existing models treat spatial cues frame-wise or source-agnostically and thus struggle to maintain source identity under overlap and motion.

Current approaches to spatial-audio reasoning typically incorporate binaural features [7], FOA intensity-vector representations [8, 9], or parallel spatial branches [10] into language models. These approaches make spatial cues accessible to language models, yet still struggle to associate evolving observations with a

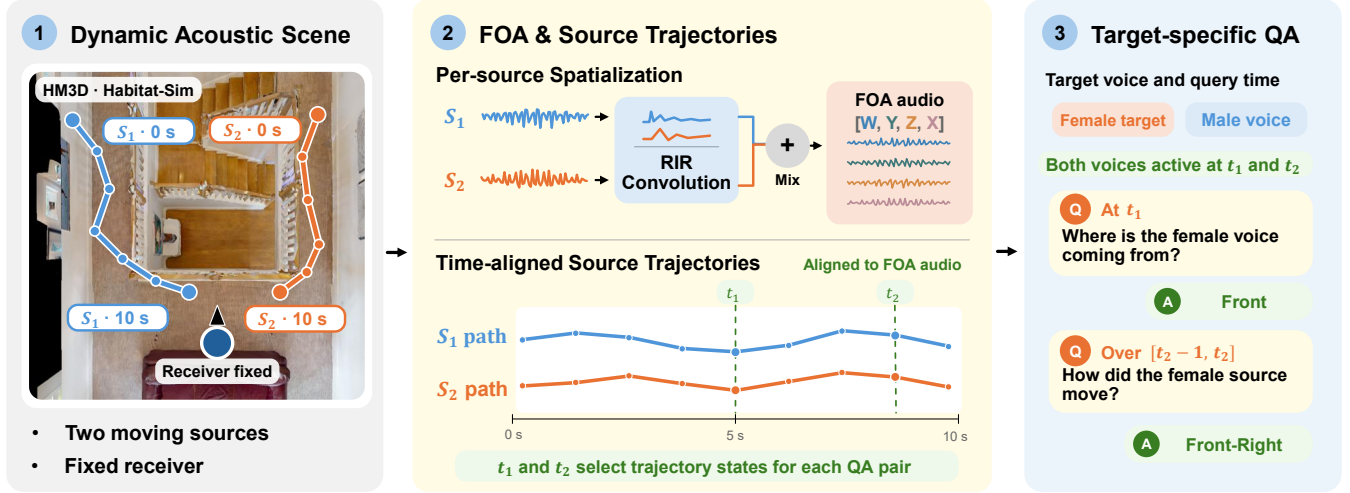


Figure 1: Task overview of Sonic-Bench. Given an FOA recording, the task is to infer a specified source’s spatial attributes at a queried time or its movement direction over a queried interval. Time-aligned source trajectories provide ground-truth answers, illustrated here with two moving sources and a fixed receiver.

specified source and recover its state at an arbitrary queried time or interval [11, 6]. Inspired by structured trajectory modeling in audio-visual systems [12], object-centric dynamics [13], and latent dynamics and world modeling [14, 15], we organize acoustic observations into persistent source states that capture spatial evolution for source- and time-specific question answering.

To support learning and evaluation of this task, we introduce **Sonic-Bench**, comprising 2.8K FOA clips and 28.8K target-specific question–answer pairs across 480 simulated indoor scenes. It covers stationary and moving speech in single- and dual-source settings, with scene-disjoint training and test splits. Each question specifies a target source and a timestamp or interval for localization, distance estimation, or motion reasoning. We further propose **SonicWorld**, a framework that maintains persistent source states using physical, learned acoustic, and semantic evidence. Learned state transitions and recurrent observation updates model their temporal evolution, while trajectory supervision and masked-state prediction ground the states in source geometry. A spatial projector converts the state sequences into tokens that the language model reads alongside audio and question tokens. The main contribution of this paper can be summarized as follows:

- We formulate dynamic spatial audio reasoning as source-centric state inference and introduce **Sonic-Bench** for target- and time-specific localization, distance estimation, and motion reasoning. The benchmark pairs FOA recordings with full-clip source trajectories across single- and dual-source scenes, with scene-disjoint training and test splits.
- We propose **SonicWorld**, a framework that learns persistent source states through predictive state transitions and recurrent observation updates, and projects the resulting state sequences into a language model for source- and time-specific spatial question answering.
- Experiments demonstrate improved source localization on Sonic-Bench and consistent zero-shot direction and azimuth gains on Spatial LibriSpeech across three language backbones.

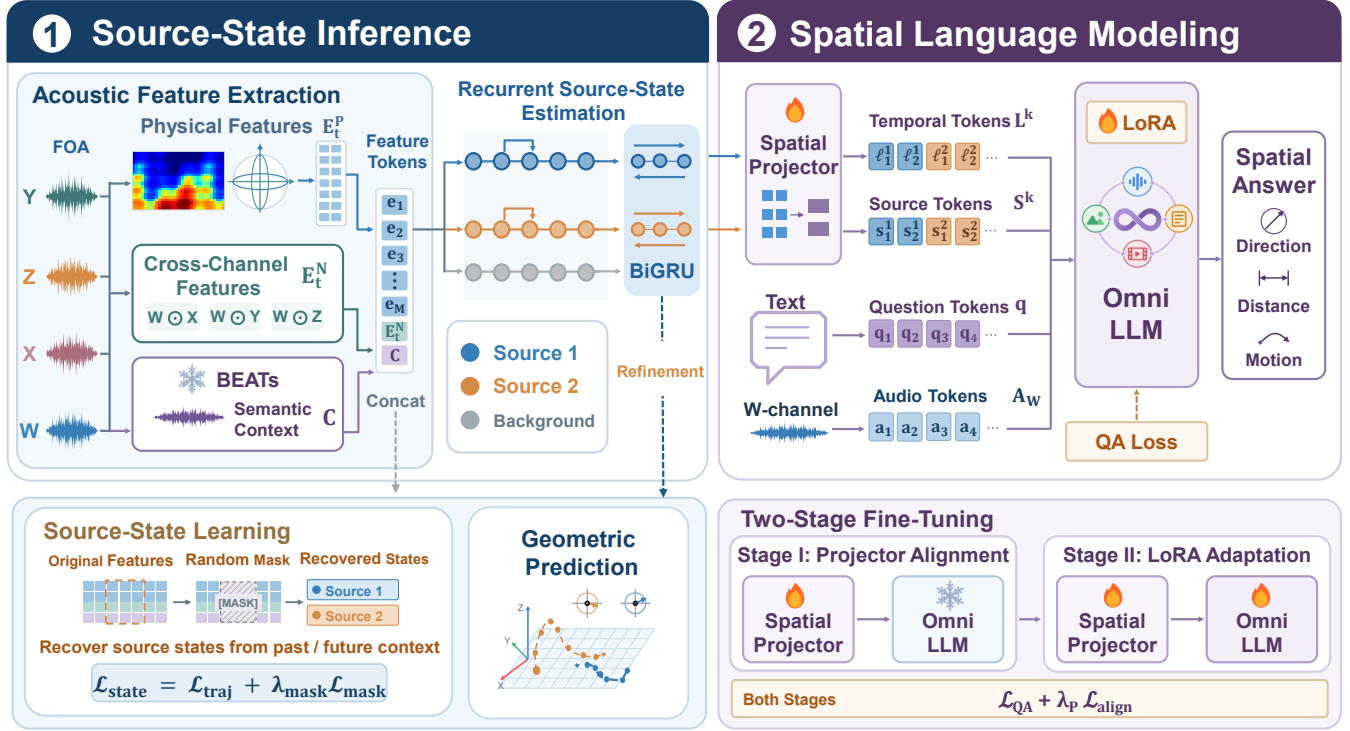


Figure 2: SonicWorld architecture. Physical, neural, and frozen-BEATs semantic evidence are concatenated before recurrent source binding. Smoothed states and decoded geometry are projected into spatial tokens, which join audio and question tokens at the language model.

2. Related Work

2.1 Spatial Audio Perception and Dynamic Acoustic Scenes

Spatial audio encodes scene geometry through binaural recordings, microphone arrays, and Ambisonics, using cues such as inter-channel timing, level differences, and sound-field structure [4, 16]. A central task is sound event localization and detection (SELD), which jointly recognizes active sound events and estimates their directions of arrival [17, 18]. Research on SELD also addresses moving-source tracking [19], localization of overlapping same-class sources [20], and source distance estimation [21]. For FOA signals, intensity vectors additionally provide physically grounded directional evidence from the multichannel sound field [22, 8, 23]. Progress has also been supported by real and simulated spatial-audio environments [24, 25, 26]. STARSS23 provides multichannel recordings with spatiotemporal annotations under natural interference and source overlap [27]. SoundSpaces 2.0 and Habitat-Sim instead offer controllable acoustic rendering in scanned indoor scenes with known geometry and source locations [28, 29]. While localization and tracking recover source geometry over time, they primarily operate at the frame or event level. We instead study source- and time-conditioned spatial reasoning, where a model must recover the spatial state of a specified source at a queried time or interval.

2.2 Audio, Omni, and Spatial Audio Language Models

Large Audio-Language Models connect pretrained audio encoders with language models for audio understanding and reasoning [30, 31]. SALMONN establishes this general architecture [1], while Kimi-Audio

and Audio Flamingo scale audio instruction following [32, 33, 34]. Omni models, including Qwen2.5-Omni and Qwen3-Omni, further integrate audio, vision, and language within unified multimodal models [2, 35]. Together, these models provide strong semantic representations but do not explicitly model spatial scene dynamics. Spatial audio–language alignment has also been explored via contrastive learning (e.g., ELSA [36] and SALM [37]). Spatial-audio LLMs further incorporate explicit spatial cues for question answering. BAT learns binaural representations for localization and spatial question answering [7]; FOA-based methods use intensity-vector features [8]; and JAEGER introduces learned Neural Intensity Vectors for 3D audio-visual grounding [9]. Spatial-Omni adds a parallel FOA branch for direction, distance, motion, and multi-source reasoning [10]. While these approaches expose spatial cues to language models, they do not maintain persistent source states across time. We organize successive acoustic observations into such states and project their temporal evolution into the language model for source- and time-specific spatial question answering.

3. Sonic-Bench

3.1 Task Definition

Given a ten-second FOA recording $\mathbf{X}$ and a query q identifying a source and a timestamp or interval, the task is to infer its spatial state in a fixed receiver-centered frame, as shown in Figure 1. We consider five query types: *position direction* (eight sectors), *azimuth*, *elevation*, and *distance* at time t , and *movement direction* (eight directions or *Stationary*) based on horizontal net displacement over $[t - 1, t]$. These queries span six source configurations: two single-source settings with a stationary or moving target, and four dual-source settings covering all target/interferer motion combinations.

Each clip has one designated target and two query times selected using speech-activity annotations. At each query time, we instantiate five task-specific questions identifying the target voice and a timestamp or one-second interval, with answers derived from the target trajectory, yielding ten QA pairs per clip. QA construction details and examples appear in Appendix A, and evaluation metrics in Appendix D.

3.2 Data Collection

Sonic-Bench evaluates source-specific spatial reasoning in dynamic single- and multi-source scenes. It contains 2,880 ten-second FOA clips, 4,800 time-aligned source trajectories, and 28,800 QA pairs across 480 simulated indoor scenes. The scene-disjoint split comprises 2,400 training clips from 400 scenes and 480 test clips from 80 scenes.

Scene and Speech Construction. We use speech from Mini LibriSpeech, a subset of the LibriSpeech corpus [38], and indoor scene meshes from HM3D [39], configuring source and receiver poses in Habitat-Sim [29]. The receiver has a fixed position and orientation; sources either remain stationary or follow navigable trajectories. Dual-source clips pair one female and one male voice, with one designated as the target and the other as the interferer.

Acoustic rendering and annotations. We render source-specific FOA room impulse responses (RIRs) along each trajectory using Habitat-Sim’s acoustic propagation engine [28], convolve them with dry speech, and sum the spatialized signals into an FOA mixture. For each source, we retain time-aligned positions, azimuth, elevation, distance, and speech activity for QA generation.

4. SonicWorld

Given an FOA waveform $X \in \mathbb{R}^{4 \times N}$ in channel order $[W, Y, Z, X]$ and a question q , SonicWorld answers spatial questions through persistent source-state inference (Figure 2). We combine physical spatial cues, learned waveform features, and a clip-level semantic summary from frozen BEATs [40]. A recurrent estimator associates these features with individual sources and updates their states over time. During training, we randomly mask temporal feature spans and supervise source-geometry recovery from surrounding context. A projector converts the state sequences into local temporal tokens and source summaries, which the language model reads alongside native audio tokens and the question. Architecture details are provided in Appendix B.

4.1 Source-State Inference

Acoustic feature extraction. The physical branch combines channel log-magnitudes with active and reactive intensity and inter-channel coherence from short-time Fourier transforms of X . Intensity uses the real and imaginary parts of the cross-spectra between W and each directional channel, normalized by W -channel power; coherence measures channel-pair consistency. A fixed Mel filterbank and temporal resampling yield $\Phi \in \mathbb{R}^{T \times M \times 13}$, with T time steps and M Mel bands. A two-dimensional convolutional network f_{phys} produces physical tokens:

$$E_t^P = [f_{\text{phys}}(\Phi)]_t \in \mathbb{R}^{M \times d_e}, \quad (1)$$

where d_e is the token feature dimension. These tokens carry normalized active-intensity directions and coherence-derived reliability for geometric aggregation.

In parallel, a shared one-dimensional convolutional network encodes each channel $c \in \{W, X, Y, Z\}$ into H_c . Following Neural IV [9], we concatenate elementwise channel interactions along the feature dimension and map them to one neural token per step:

$$H_{\text{int}} = \text{Concat}(H_W \odot H_X, H_W \odot H_Y, H_W \odot H_Z), \quad (2)$$

$$E_t^N = [g_{\text{neural}}(H_{\text{int}})]_t \in \mathbb{R}^{1 \times d_e}. \quad (3)$$

Here, $\odot$ denotes elementwise multiplication; g_{neural} applies a two-layer MLP followed by linear interpolation to T steps.

Frozen BEATs [40] extracts J patches of width d_B from W , forming $F \in \mathbb{R}^{J \times d_B}$. We normalize and project them to $\tilde{F} \in \mathbb{R}^{J \times d_e}$. A learned query $q_c \in \mathbb{R}^{1 \times d_e}$ pools these patches through attention into a normalized semantic token C . At each step, we concatenate the M physical tokens, one neural token, and the shared semantic token:

$$C = \text{Attention}(q_c, \tilde{F}, \tilde{F}) \in \mathbb{R}^{1 \times d_e}, \quad (4)$$

$$E_t = \text{Concat}_{\text{token}}(E_t^P, E_t^N, C) \in \mathbb{R}^{(M+2) \times d_e}. \quad (5)$$

Recurrent source-state estimation. We associate E_t with K recurrent source slots and one background slot, following sequential slot-based models [41, 42]. Each slot k maintains a d_s -dimensional state $z_t^k \in \mathbb{R}^{d_s}$ at time t , initialized from a learned z_0^k ; the background slot is indexed by $K + 1$. Before reading E_t , a shared two-layer feed-forward network f_{trans} predicts the prior:

$$\tilde{z}_t^k = \text{LayerNorm}(z_{t-1}^k + f_{\text{trans}}(z_{t-1}^k)). \quad (6)$$

For token $E_{t,m} \in \mathbb{R}^{d_e}$, $m = 1, \dots, M+2$, $W_Q \in \mathbb{R}^{d_s \times d_s}$ maps the prior to a query, while $W_K, W_V \in \mathbb{R}^{d_s \times d_e}$ map tokens to keys and values. The scores form $\ell_t \in \mathbb{R}^{(K+1) \times (M+2)}$. Following Slot Attention [43], we apply softmax across slots, then normalize each slot's weights over tokens:

$$\ell_{t,k,m} = \frac{(W_Q \bar{z}_t^k)^\top (W_K E_{t,m})}{\sqrt{d_s}}, \quad (7)$$

$$A_{t,k,m} = \text{softmax}_k(\ell_{t,k,m}),$$

$$\tilde{A}_{t,k,m} = \frac{A_{t,k,m}}{\sum_{m'=1}^{M+2} A_{t,k,m'}}. \quad (8)$$

The resulting weights form a feature summary $o_t^k \in \mathbb{R}^{d_s}$, which a gated recurrent unit (GRU) [44] combines with the prior:

$$o_t^k = \sum_{m=1}^{M+2} \tilde{A}_{t,k,m} W_V E_{t,m}, \quad (9)$$

$$z_t^k = \text{LayerNorm} \left(\text{GRU}(o_t^k, \bar{z}_t^k) \right). \quad (10)$$

Updated states pass to the next step, and the K source sequences are retained for refinement.

State refinement and geometric prediction. For each source, a bidirectional GRU aggregates earlier and later states in $z_{1:T}$. A linear projection f_{ctx} maps its output at t to context $c_t \in \mathbb{R}^{d_s}$. A gate conditioned on z_t and c_t controls the context added to the recurrent state:

$$c_t = f_{\text{ctx}} \left([\text{BiGRU}(z_{1:T})]_t \right),$$

$$g_t = \sigma \left(f_{\text{gate}}([z_t; c_t]) \right), \quad (11)$$

$$h_t = \text{LayerNorm}(z_t + g_t \odot c_t) \in \mathbb{R}^{d_s}.$$

Here, f_{gate} is linear, σ is sigmoid, $[\cdot]_t$ selects time t , and $[\cdot; \cdot]$ concatenates features. Auxiliary heads predict unit direction $\hat{u}_t$, log-distance $\hat{d}_t$, source presence, and positional uncertainty from h_t for trajectory supervision. With $\hat{r}_t = \exp(\hat{d}_t)$, the receiver-centered position is

$$\hat{p}_t = \hat{r}_t \hat{u}_t \in \mathbb{R}^3. \quad (12)$$

Source-state Learning. One Hungarian assignment per clip matches predicted source sequences to ground-truth trajectories. Matched annotations supervise direction, distance, and presence. We also supervise $\hat{p}_t$ with an uncertainty position loss and supervise direction and distance decoded from the transition prior $\bar{z}_t^k$.

The trajectory and combined pretraining objectives are

$$\mathcal{L}_{\text{traj}} = \mathcal{L}_{\text{dir}} + \mathcal{L}_{\text{dist}} + \mathcal{L}_{\text{presence}} \\ + \lambda_u \mathcal{L}_{\text{uncertainty}} + \lambda_{\text{prior}} \mathcal{L}_{\text{prior}}, \quad (13)$$

$$\mathcal{L}_{\text{state}} = \mathcal{L}_{\text{traj}} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}. \quad (14)$$

Loss definitions and masking settings are provided in Appendix C.

4.2 Spatial Language Modeling

Spatial projector. A trainable projector maps source-state features and additional spatial features to frame features. It compresses consecutive frame features into local tokens, while learned queries summarize each complete source sequence. Both token types are projected to the language embedding space and combined with native W -channel audio tokens and the question q for answer generation. Projector details are provided in Appendix B.

Two-stage fine-tuning. As shown in Figure 2, Stage I trains the projector and auxiliary alignment heads with the LLM fixed; Stage II jointly trains the projector and low-rank adaptation (LoRA) parameters in its attention layers. The source-state model, native audio encoder, and original LLM weights remain frozen. Both stages optimize

$$\mathcal{L}_{\text{LLM}} = \mathcal{L}_{\text{QA}} + \lambda_P \mathcal{L}_{\text{align}}, \quad (15)$$

where $\mathcal{L}_{\text{QA}}$ is cross-entropy over answer tokens. Algorithm 1 details the training procedure.

5. Experimental Setup

5.1 Training Details

We train SonicWorld on the training split of Sonic-Bench using Qwen2.5-Omni-7B [2] as the backbone and pretrained BEATs [40] for semantic audio feature extraction. We first train the source-state model for 25 epochs at a learning rate of 10^{-4} , then freeze it for the two fine-tuning stages. The projector is trained for five epochs at a learning rate of 10^{-4} with the LLM frozen. We then jointly optimize the projector and LLM LoRA [45] parameters for three epochs, using the learning rates of 3×10^{-5} and 5×10^{-5} , respectively. LoRA is applied to the LLM’s attention layers with rank 16, a scaling parameter of 32, and dropout of 0.05. Detailed optimization settings and loss weights are provided in Appendix C.

5.2 Evaluation Metrics and Baselines

Evaluation metrics. We report classification accuracy (%) for target-direction (Dir.) and movement-direction (Mot.) questions on Sonic-Bench using exact label matching. For azimuth (Az.), elevation (El.), and distance (Dist.) estimation, we report mean absolute error (MAE), with angular errors measured in degrees and distance errors in meters. Azimuth error is computed as the shortest angular distance between the predicted and ground-truth angles. We additionally report accuracy at error thresholds of 20° , 1° , and 0.5 m for azimuth, elevation, and distance, respectively. Detailed metric definitions and evaluation protocols are provided in Appendix D.

Baselines. We compare SonicWorld with the spatial audio baselines Spatial-Omni [10], BAT [7], JAEGER(Classical-IV) and JAEGER(Neural-IV) [9]. General-purpose multimodal baselines include the open-weight models Qwen2.5-Omni-7B [2], InternOmni [46, 47], Audio Flamingo 3 [34], Phi-4-MM-5.6B [48] and Kimi-Audio-7B-Instruct [32], as well as the closed-source models Gemini 3 Flash and Gemini 3.1 Pro [49].

6. Results

6.1 Main results

With the same Qwen2.5-Omni-7B backbone, SonicWorld improves direction accuracy over Spatial-Omni from 43.75% to 50.00% for single sources and from 26.09% to 27.03% for dual sources. Azimuth MAE decreases from 46.87° to 30.85° and from 62.40° to 56.59° , respectively. These gains persist under source overlap, although their magnitude is smaller. Dual-source distance error is 0.60 m, compared with 0.62 m for Classical-IV.

SonicWorld obtains 52.50%/53.91% movement-direction accuracy on single/dual sources, versus BAT at 51.25%/52.66%. Figure 3 illustrates two cases where SonicWorld correctly predicts the target’s interval-specific movement, while Spatial-Omni and JAEGER (Classical-IV) do not. Comparisons across language backbones are provided in Appendix D.2.

6.2 Zero-Shot Transfer

Zero-shot evaluation tests whether SonicWorld checkpoints trained on Sonic-Bench transfer to static-source localization without additional training. The evaluation uses selected test subsets of SpatialSceneQA [9] and Spatial LibriSpeech (SLS) [50], retaining recordings of at least ten seconds and using their first ten seconds. SpatialSceneQA tests azimuth and elevation separately, while single-source SLS covers direction, azimuth, elevation, and distance. Appendix D details sample selection and source-count breakdowns.

Table 2 reports zero-shot transfer results on two external datasets. On SpatialSceneQA, SonicWorld reduces azimuth MAE by 80.3% relative to Spatial-Omni, with improvements in both single- and dual-source settings (Appendix D). On SLS, SonicWorld achieves the highest direction accuracy (73.98%) and lowest azimuth MAE (19.29°) among the models reported in Table 2. Zero-shot comparisons across the three evaluated backbones are consolidated in Table D.2.

6.3 Ablation studies

Effectiveness of Modules. The full model achieves the best results across all six metrics in Table 3. Physical-only and no-BEATs have azimuth MAEs 7.67° and 7.58° higher than Full, and Az20 scores 9.17 and 8.02 points lower, respectively. No-BiGRU is close in azimuth MAE (48.11° vs. 48.01°), but has lower Dir. (32.19% vs. 34.69%) and higher distance MAE (0.63 m vs. 0.59 m).

Effectiveness of Source-state Modeling. As shown in Table 4, removing prior-state supervision increases azimuth MAE by 13.24° , highlighting the importance of geometrically grounded priors. Removing persistent source slots yields the highest elevation and distance errors, while removing the transition MLP causes the largest motion-accuracy drop (1.98 percentage points). Without masked-state loss, Az20 decreases by 4.69 percentage points, demonstrating the benefit of learning to recover source geometry from temporal context.

Table 1: Experimental results on Sonic-Bench. Spatial baselines and SonicWorld use Qwen2.5-Omni-7B. Dir./Mot. are label accuracies (%); Az./El. are MAE in degrees and Dist. is MAE in meters. Best values in each column are bold.

Model	Single source					Dual source				
	Dir.↑	Az.↓	El.↓	Dist.↓	Mot.↑	Dir.↑	Az.↓	El.↓	Dist.↓	Mot.↑
Closed-source models										
Gemini 3 Flash	12.81	88.55	4.40	1.11	7.81	15.47	88.23	4.49	2.71	8.28
Gemini 3.1 Pro	9.06	90.87	15.78	1.86	12.19	14.53	90.32	28.88	2.03	14.38
Open-source models										
Qwen2.5-Omni-7B	12.50	91.18	84.91	98.83	51.88	12.66	88.33	83.58	34.30	48.59
InternOmni	12.81	93.72	22.65	40.58	8.13	12.34	92.90	32.51	43.16	5.94
Phi-4-MM-5.6B	13.44	81.21	3.29	1.93	6.25	12.66	79.90	4.13	2.26	5.31
Audio Flamingo 3	14.06	85.62	9.60	2.41	7.81	12.34	88.80	17.32	3.12	5.16
Kimi-Audio-7B-Instruct	12.50	91.78	1.58	2.20	35.31	11.72	92.23	3.89	1.54	23.75
Spatial baselines										
BAT	8.75	87.51	0.94	0.68	51.25	11.09	90.34	0.95	0.64	52.66
Classical-IV	29.38	47.92	1.14	0.58	51.56	20.94	66.06	1.08	0.62	52.66
Neural-IV	10.63	84.00	0.99	0.59	50.31	11.88	87.83	1.21	0.67	52.34
Spatial-Omni	43.75	46.87	1.57	1.36	49.06	26.09	62.40	1.12	3.08	50.94
SONICWORLD	50.00	30.85	0.91	0.56	52.50	27.03	56.59	0.93	0.60	53.91

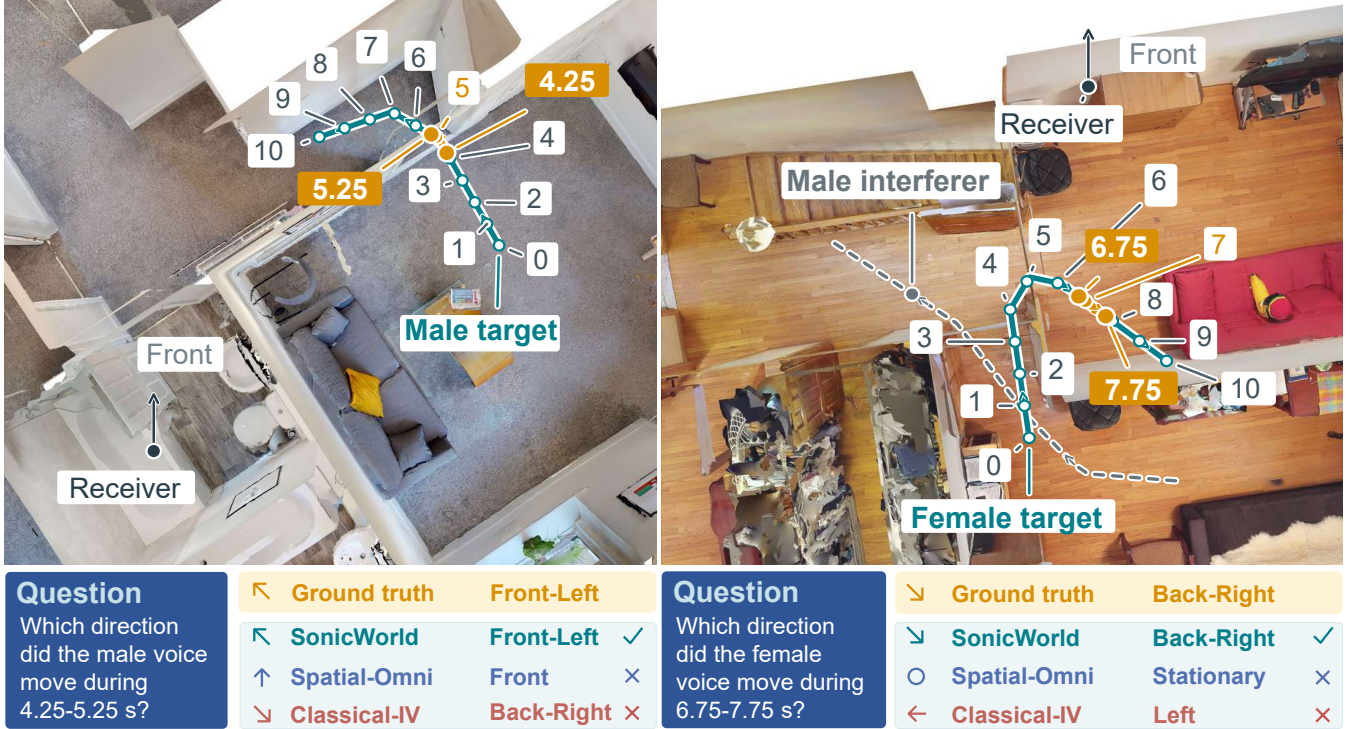


Figure 3: Single-source (left) and dual-source (right) motion examples, with queried intervals in orange. SonicWorld correctly predicts net movement directions; Spatial-Omni and JAEGER (Classical-IV) do not.

Table 2: Zero-shot transfer to SpatialSceneQA and SLS. Spatial baselines and SonicWorld use Qwen2.5-Omni-7B.

Model	SpatialSceneQA		Dir.↑	SLS		
	Az.↓	El.↓		Az.↓	El.↓	Dist.↓
Open-source models						
Qwen2.5-Omni-7B	60.47	80.14	12.69	90.26	88.47	55.04
InternOmni	80.54	78.05	12.38	89.69	57.59	98.83
Phi-4-MM-5.6B	61.86	18.31	12.61	89.43	7.08	1.99
Audio Flamingo 3	132.66	24.92	11.87	89.92	28.08	33.40
Kimi-Audio-7B-Instruct	55.37	35.44	11.63	90.74	9.41	3.90
Spatial baselines						
BAT	125.37	20.51	11.85	90.75	7.07	0.84
Classical-IV	33.56	19.17	14.49	75.53	7.70	0.89
Neural-IV	88.33	18.49	12.71	89.38	7.74	0.78
Spatial-Omni	108.28	22.84	27.14	70.74	16.64	5.02
SonicWorld	21.35	20.51	73.98	19.29	7.09	0.89

Table 3: Module ablations with Qwen2.5-Omni-7B. ✓ denotes enabled and ✗ disabled.

Physical	Neural	BEATs	BiGRU	Dir.↑	Mot.↑	Az20↑	Az.↓	El.↓	Dist.↓
✓	✗	✗	✓	28.33	52.40	25.73	55.68	0.97	0.62
✓	✓	✗	✓	28.65	52.19	26.88	55.59	0.99	0.62
✓	✓	✓	✗	32.19	51.98	33.13	48.11	0.96	0.63
✓	✓	✓	✓	34.69	53.44	34.90	48.01	0.92	0.59

Table 4: Ablations of source-state modeling with Qwen2.5-Omni-7B.

Variant	Dir.↑	Mot.↑	Az20↑	Az.↓	El.↓	Dist.↓
w/o transition MLP	33.44	51.46	32.71	49.94	0.96	0.64
w/o prior-state supervision	24.48	51.67	22.19	61.25	1.05	0.65
w/o masked-state loss	33.02	52.81	30.21	50.66	0.98	0.64
w/o persistent source slots	33.65	52.81	25.94	54.13	1.09	0.68
Full model	34.69	53.44	34.90	48.01	0.92	0.59

7. Conclusion

Dynamic spatial audio reasoning requires associating acoustic observations with individual sources and tracking their spatial evolution over time. We introduce **Sonic-Bench**, which pairs FOA recordings with time-aligned trajectories and source-specific spatial questions in dynamic single- and multi-source scenes. To address these tasks, we propose **SonicWorld**, which integrates physical, learned acoustic, and semantic features into persistent source states through trajectory supervision and masked-state prediction, and connects these states to an Omni LLM for spatial question answering. Experiments show that persistent source-state modeling improves source-specific localization and motion reasoning on Sonic-Bench, while zero-shot evaluations demonstrate transferable localization gains across different language backbones. These results suggest that maintaining explicit source-level temporal states provides an effective interface between spatial audio perception and language-based reasoning, enabling models to reason about evolving auditory scenes beyond individual acoustic observations.

Limitations

This work has several limitations:

- ▶ **Simulated indoor environments.** Our study focuses on simulated indoor scenes. Broader acoustic conditions, source categories, and denser multi-source scenes remain to be explored, along with validation on real-world recordings.
- ▶ **Stationary receiver.** The receiver remains fixed, leaving receiver motion and changing egocentric reference frames outside the current scope. Future work could incorporate moving receivers and extend source-state modeling toward action-conditioned prediction of acoustic observations for robotic active listening.
- ▶ **Offline inference.** The bidirectional temporal module processes the complete audio clip, making the current implementation better suited to offline analysis than streaming perception. Causal state updates could support continuous auditory interaction.

References

- [1] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models. In *International Conference on Learning Representations*, volume 2024, pages 16607–16629, 2024.
- [2] 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, 2025. URL <https://arxiv.org/abs/2503.20215>.
- [3] Zhihan Guo, Wenqian Cui, Guan-Ting Lin, Daxin Tan, Jingyao Li, Qiyong Zheng, Dingdong Wang, Jing Xiong, Han Shi, Jiaya Jia, et al. A survey of audio reasoning in multimodal foundation models. *arXiv preprint arXiv:2605.21008*, 2026.
- [4] Zhiyuan Zhu, Yu Zhang, Wenxiang Guo, Changhao Pan, and Zhou Zhao. Asaudio: A survey of advanced spatial audio research. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 417–442, 2025.

- [5] Zihan Liu, Zhikang Niu, Qiuyang Xiao, Zhisheng Zheng, Ruoqi Yuan, Yuhang Zang, Yuhang Cao, Xiaoyi Dong, Jianze Liang, Xie Chen, et al. Star-bench: Probing deep spatio-temporal reasoning as audio 4d intelligence. In *International Conference on Learning Representations*, volume 2026, pages 134703–134731, 2026.
- [6] Arvind Krishna Sridhar, Yinyi Guo, and Erik Visser. Spatial audio question answering and reasoning on dynamic source movements. *arXiv preprint arXiv:2602.16334*, 2026.
- [7] Zhisheng Zheng, Puyuan Peng, Ziyang Ma, Xie Chen, Eunsol Choi, and David Harwath. Bat: Learning to reason about spatial sounds with large language models. *arXiv preprint arXiv:2402.01591*, 2024.
- [8] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Jun Zhang, Lu Lu, Zejun Ma, Yuxuan Wang, et al. Can large language models understand spatial audio? *arXiv preprint arXiv:2406.07914*, 2024.
- [9] 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.
- [10] Zhiyuan Zhu, Yixuan Chen, Yiwen Shao, Wenxiang Guo, Changhao Pan, Yu Zhang, Yuxiang Wang, Wei Liu, Houhua Zhang, Chengkuan Zeng, et al. Spatial-omni: Spatial audio understanding integration in multimodal llms via foa encoding. *arXiv preprint arXiv:2606.10738*, 2026.
- [11] Zhe Sun, Yujun Cai, Jiayu Yao, and Yiwei Wang. Spatial blind spot: Auditory motion perception deficits in audio llms. *arXiv e-prints*, page arXiv:2511.13273, 2025.
- [12] Mingfei Chen, Zijun Cui, Xiulong Liu, Jinlin Xiang, Yang Zheng, Jingyuan Li, and Eli Shlizerman. Savvy: Spatial awareness via audio-visual llms through seeing and hearing. *Advances in Neural Information Processing Systems*, 38:118999–119038, 2025.
- [13] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *arXiv preprint arXiv:2210.05861*, 2022.
- [14] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.
- [15] Jiahua Wang, Shannan Yan, Leqi Zheng, Jialong Wu, and Yaoxin Mao. Audio-visual world models: Towards multisensory imagination in sight and sound. *arXiv e-prints*, page arXiv:2512.00883, 2025.
- [16] Pierre-Amaury Grumiaux, Srđan Kitić, Laurent Girin, and Alexandre Guérin. A survey of sound source localization with deep learning methods. *The Journal of the Acoustical Society of America*, 152(1): 107–151, 2022.
- [17] Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):34–48, 2018.
- [18] Kazuki Shimada, Yuichiro Koyama, Naoya Takahashi, Shusuke Takahashi, and Yuki Mitsufuji. Accdoa: Activity-coupled cartesian direction of arrival representation for sound event localization and detection. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 915–919. IEEE, 2021.

- [19] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen. Localization, detection and tracking of multiple moving sound sources with a convolutional recurrent neural network. *arXiv preprint arXiv:1904.12769*, 2019.
- [20] Kazuki Shimada, Yuichiro Koyama, Shusuke Takahashi, Naoya Takahashi, Emiru Tsunoo, and Yuki Mitsufuji. Multi-acddoa: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 316–320. IEEE, 2022.
- [21] David Diaz-Guerra, Archontis Politis, Parthasaarathy Ariyakulam Sudarsanam, Kazuki Shimada, Daniel Krause, Kengo Uchida, Yuichiro Koyama, Naoya Takahashi, Shusuke Takahashi, Takashi Shibuya, et al. Baseline models and evaluation of sound event localization and detection with distance estimation in dcase 2024 challenge. In *Workshop on Detection and Classification of Acoustic Scenes and Events*, pages 41–45. DCASE, 2024.
- [22] Lauréline Perotin, Romain Serizel, Emmanuel Vincent, and Alexandre Guérin. Crnn-based multiple doa estimation using acoustic intensity features for ambisonics recordings. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):22–33, 2019.
- [23] Thi Ngoc Tho Nguyen, Karn N Watcharasupat, Ngoc Khanh Nguyen, Douglas L Jones, and Woon-Seng Gan. Salsa: Spatial cue-augmented log-spectrogram features for polyphonic sound event localization and detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1749–1762, 2022.
- [24] Christine Evers, Heinrich W Löllmann, Heinrich Mellmann, Alexander Schmidt, Hendrik Barfuss, Patrick A Naylor, and Walter Kellermann. The locata challenge: Acoustic source localization and tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1620–1643, 2020.
- [25] Archontis Politis, Sharath Adavanne, Daniel Krause, Antoine Deleforge, Prerak Srivastava, and Tuomas Virtanen. A dataset of dynamic reverberant sound scenes with directional interferers for sound event localization and detection. *arXiv preprint arXiv:2106.06999*, 2021.
- [26] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *European conference on computer vision*, pages 17–36. Springer, 2020.
- [27] Kazuki Shimada, Archontis Politis, Parthasaarathy Sudarsanam, Daniel A Krause, Kengo Uchida, Sharath Adavanne, Aapo Hakala, Yuichiro Koyama, Naoya Takahashi, Shusuke Takahashi, et al. Starss23: An audio-visual dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events. *Advances in neural information processing systems*, 36:72931–72957, 2023.
- [28] Changan Chen, Carl Schissler, Sanchit Garg, Philip Kobernik, Alexander Clegg, Paul Calamia, Dhruv Batra, Philip Robinson, and Kristen Grauman. Soundspaces 2.0: A simulation platform for visual-acoustic learning. *Advances in Neural Information Processing Systems*, 35:8896–8911, 2022.
- [29] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9338–9346. IEEE, 2019.
- [30] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

- [31] Yuan Gong, Hongyin Luo, Alexander Liu, Leonid Karlinsky, and James R Glass. Listen, think, and understand. In *International Conference on Learning Representations*, volume 2024, pages 18516–18545, 2024.
- [32] Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.
- [33] Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities. *arXiv preprint arXiv:2402.01831*, 2024.
- [34] Sreyan Ghosh, Arushi Goel, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, et al. Audio flamingo 3: Advancing audio intelligence with fully open large audio language models. *Advances in Neural Information Processing Systems*, 38:41819–41886, 2025.
- [35] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfu Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.
- [36] Bhavika Devnani, Skyler Seto, Zakaria Aldeneh, Alessandro Toso, Elena Menyaylenko, Barry-John Theobald, Jonathan Sheaffer, and Miguel Sarabia. Learning spatially-aware language and audio embeddings. *Advances in Neural Information Processing Systems*, 37:33505–33537, 2024.
- [37] Jinbo Hu, Yin Cao, Ming Wu, Zhenbo Luo, and Jun Yang. Salm: Spatial audio language model with structured embeddings for understanding and editing. *arXiv preprint arXiv:2507.16724*, 2025.
- [38] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.
- [39] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.
- [40] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*, 2022.
- [41] Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021.
- [42] Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd Van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *Advances in Neural Information Processing Systems*, 35:28940–28954, 2022.
- [43] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

- [44] Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1724–1734, 2014.
- [45] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [46] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.
- [47] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.
- [48] Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.
- [49] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [50] Miguel Sarabia, Elena Menyaylenko, Alessandro Toso, Skyler Seto, Zakaria Aldeneh, Shadi Pirhoseinloo, Luca Zappella, Barry-John Theobald, Nicholas Apostoloff, and Jonathan Sheaffer. Spatial librispeech: An augmented dataset for spatial audio learning. *arXiv preprint arXiv:2308.09514*, 2023.
- [51] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [52] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.
- [53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

A. Sonic-Bench

A.1 Data Sources and Scene-Level Split

Sonic-Bench combines speech from Mini LibriSpeech [38], distributed under CC BY 4.0, with scanned indoor scene meshes from HM3D [39], subject to the provider’s access and licensing terms. We partition the benchmark by scene: 400 scenes from the HM3D training set form our training split, and 80 scenes from its validation set form our test split. Table A.1 summarizes the resulting dataset.

Table A.1: Sonic-Bench statistics by split. Each clip contains two target query times and five questions per query time.

Unit	Train	Test	Total
Scenes	400	80	480
Single-source clips	800	160	960
Dual-source clips	1,600	320	1,920
Source trajectories	4,000	800	4,800
Question–answer pairs	24,000	4,800	28,800
Audio duration	6 h 40 min	1 h 20 min	8 h

We use one utterance per speaker from 24 speakers (14 female-labeled, 10 male-labeled) in Mini LibriSpeech’s `train-clean-5` and `dev-clean-2` archives. Utterances of at least 10 s are converted to 16 kHz mono and randomly cropped to 10 s, with approximately 10 ms fades at both ends. Dual-source clips pair independently sampled female and male speech, with target gender balanced in both single- and dual-source settings.

A.2 Comparison with Existing Resources

Table A.2 summarizes representative spatial-audio resources. Sonic-Bench combines full-clip source trajectories with target- and time-specific questions, covering stationary and moving targets in single- and dual-source scenes with all target/interferer motion combinations.

A.3 Scene Layouts and Source Trajectories

The receiver remains at a fixed position and orientation. Moving trajectories follow navigable shortest paths between sampled endpoints and are resampled uniformly in cumulative path length. This produces constant path speed, with direction changes at turns. Stationary sources retain a single position throughout the clip. The geometric constraints and rendering parameters are listed in Table A.3.

The six target/interferer motion configurations in Table A.4 are crossed with eight bearing sectors: Front, Front-Left, Left, Back-Left, Back, Back-Right, Right, and Front-Right. The receiver heading places the complete target path within its assigned sector. Both position-direction questions in a clip therefore have the same sector label; azimuth, elevation, distance, and interval displacement are computed at their respective query times. Figure A.1 visualizes the sector boundaries and the balanced clip distribution.

A.4 Acoustic Synthesis

Habitat-Sim’s geometry-based propagation engine renders a source-specific FOA room impulse response at selected path positions [29, 28]. At each candidate time, a new response is computed when displacement

Table A.2: Comparison with spatial-audio resources. Scale uses each resource’s reporting unit; Spatial supervision refers to geometry available during construction or training.

Resource	Format	Scale	Motion	Spatial supervision	Queries
SpatialSoundQA [7]	Binaural	872K QA	Static	Static source geometry	Source attributes / relations
SpatialSceneQA [9]	FOA, RGB-D	61K samples	Static	Static 3D poses	Source identity / visual grounding
SAVVY-Bench [12]	7-ch audio, video	1,522 QA	Static / Moving	Camera-derived speaker tracks	Sound event-anchored
STAR-Bench [5]	Binaural	2,353 QA	Static / Moving	Motion / trajectory QA labels	Motion-pattern reasoning
SO-Bench [10]	FOA	7K QA	Static / Moving	Linked SO-Dataset tracks	Task-dependent source / time
Sonic-Bench (ours)	FOA	28.8K QA	Static / Moving	Full-clip source tracks	Target- and time-specific

Table A.3: Scene and acoustic generation parameters.

Parameter	Setting
Duration / sample rate	10 s / 16 kHz
Geometric sampling	12 Hz; 121 frames (including endpoints)
Receiver height	1.50 m above NavMesh
Source height	1.55 m above NavMesh
Source–receiver range	1.5–4 m along the path
Moving-path length	2–5 m
Source separation	At least 0.8 m
RIR candidate rate	At most 4 Hz
Displacement trigger	0.35 m
RIR support	0.25 s
Terminal fade	0.025 s
Propagation components	Direct, indirect, diffraction, transmission
Acoustic materials	Default uniform materials
FOA output	Four-channel PCM16
FOA metadata	N3D

from the previous acoustic keyframe reaches 0.35 m. Stationary sources generally use one response. Each RIR is truncated to 0.25 s and tapered over its final 0.025 s.

Dry speech is partitioned at acoustic keyframes. Each segment is convolved with its corresponding RIR; response tails are added where they overlap. With dry waveform x_s , keyframe interval $\mathcal{I}_{s,j}$, and channel-specific response $h_{s,c}^{(j)}$, the mixture is

$$y_c[n] = G \sum_s \sum_j \left(h_{s,c}^{(j)} * (x_s \mathbf{1}_{\mathcal{I}_{s,j}}) \right) [n]. \quad (\text{A.1})$$

Responses are rotated and reordered to the receiver convention. The shared gain G limits the peak amplitude

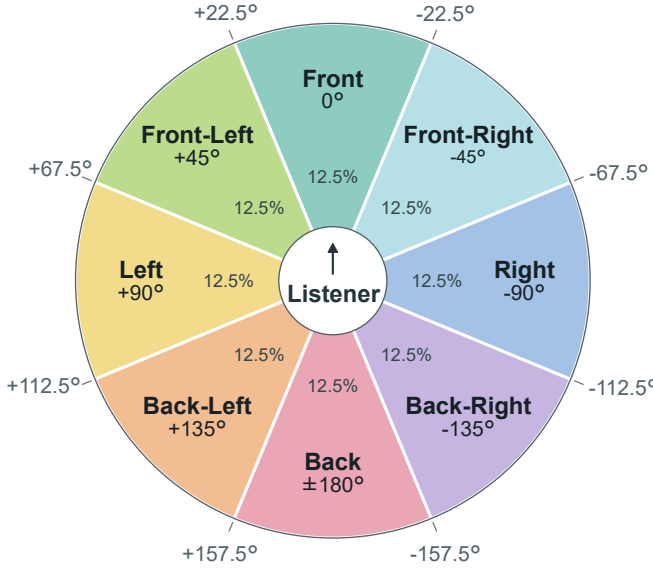


Table A.4: Source-motion configurations. Each row is balanced across eight target-bearing sectors.

Sources	Target	Interferer	Train	Test
Single	Moving	–	400	80
Single	Stationary	–	400	80
Dual	Moving	Moving	400	80
Dual	Moving	Stationary	400	80
Dual	Stationary	Moving	400	80
Dual	Stationary	Stationary	400	80

Figure A.1: Target-bearing sectors and clip distribution. Each sector spans 45° and contains 12.5% of the clips in each split.

of all four channels together, and the output is cropped to 160,000 samples.

A.5 Coordinates and Source Annotations

All model positions are receiver-centered, with $+x$ forward, $+y$ left, and $+z$ upward. The receiver’s rotation transforms the world-space source–receiver displacement into a local simulator vector p_{sim} . The model-coordinate vector is $p = (-z_{\text{sim}}, -x_{\text{sim}}, y_{\text{sim}})$. For $p = (x, y, z)$, distance, azimuth, and elevation are

$$\begin{aligned} r &= \|p\|_2, \quad \theta = \text{atan2}(y, x), \\ \phi &= \text{atan2}\left(z, \sqrt{x^2 + y^2}\right). \end{aligned} \tag{A.2}$$

Angles are reported in degrees: azimuth is positive leftward in $[-180^\circ, 180^\circ]$, and elevation is positive upward in $[-90^\circ, 90^\circ]$.

Each source is annotated at $t_i = i/12$ seconds for $i = 0, \dots, 120$. Annotations include position, azimuth, elevation, distance, path speed, source presence, speech activity, and dry-speech energy. Source presence denotes an existing trajectory, including silent speech intervals. Speech activity is determined from dry-waveform RMS energy in an approximately 0.25-second window, with a threshold of the greater of -50 dBFS and 25 dB below the clip’s maximum local energy. Presence supervises the source-state model; speech activity determines valid question times.

A.6 Target-Specific Questions

Task definitions. At each query time, position direction assigns the target’s azimuth to one of eight sectors, while azimuth, elevation, and distance return the corresponding numerical attributes. The sector index is

$$\text{sector}(\theta) = \left\lfloor \frac{\text{wrap}_{360}(\theta + 22.5^\circ)}{45^\circ} \right\rfloor, \tag{A.3}$$

where wrap_{360} maps angles to $[0^\circ, 360^\circ)$. Indices follow the sector order given above. Movement direction uses the horizontal component of $\Delta p = p_t - p_{t-1s}$ in the same fixed receiver frame. Displacements of norm at most 10^{-5} m receive the Stationary label; other displacements use the sector of $\text{atan2}(\Delta p_y, \Delta p_x)$.

Question construction. Questions identify the target voice and specify either a timestamp or the preceding one-second interval. Answers are calculated from the annotated target trajectory. Five tasks at each of two query times yield ten QA pairs per clip, with 4,800 training and 960 test questions per task. Table A.5 shows examples. The complete question templates and SonicWorld input instruction are presented in Appendix F.

Table A.5: Examples of target-specific spatial questions.

Task	Example question	Answer
Position direction	At 3.00 s, which direction is the female voice coming from?	Front
Azimuth	At 6.00 s, what is the azimuth of the female voice?	-9.47°
Elevation	At 4.50 s, what is the vertical angle of the male voice?	$+2.25^\circ$
Distance	At 7.00 s, how far away is the male voice?	3.12 m
Movement direction	Between 1.50 and 2.50 s, which direction did the female source move relative to the listener’s fixed orientation?	Front-Right

A.7 Dataset Distributions

Table A.6 describes the source geometry across all annotated source states. At the target query times, moving and stationary targets each account for 2,400 training queries and 480 test queries. Table A.7 gives the movement-direction distribution, and Figure A.2 shows the detailed attributes at the query times.

Table A.6: Source geometry and trajectory statistics.

Quantity	Statistic	Train	Test
Source range (m)	Minimum	1.50	1.50
	Maximum	4.00	4.00
	Mean	2.85	2.82
	Median	2.90	2.87
Elevation ($^\circ$)	Minimum	-16.76	-13.45
	Maximum	19.94	20.30
	P25	0.81	0.83
	P50 (median)	0.99	1.01
	P75	1.32	1.35
Mean moving-path length (m)		2.74	2.74
Mean moving speed (m/s)		0.27	0.27
Minimum dual-source separation (m)		0.80	0.82

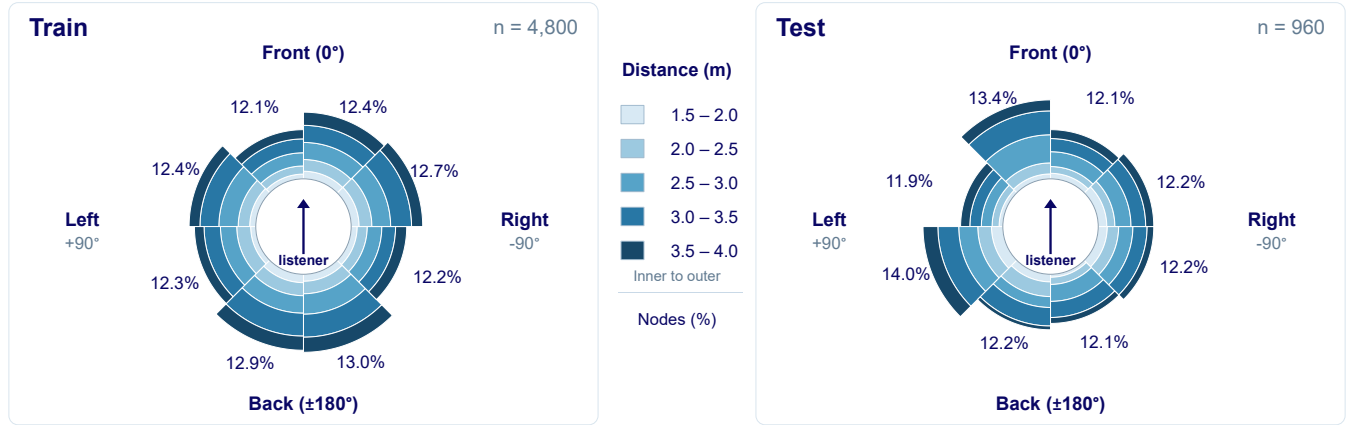
B. Model Architecture

The Qwen2.5-Omni-7B configuration processes ten-second FOA recordings at 16 kHz in channel order $[W, Y, Z, X]$. Each clip is represented by 121 source-state steps.

Table A.7: Test movement-direction distribution. Counts denote questions in the single- and dual-source subsets. S denotes Stationary; F/B/L/R denote Front/Back/Left/Right, with combined letters indicating diagonal directions. Stationary labels account for half of each subset.

Subset	S	F	FL	L	BL	B	BR	R	FR	Total
Single	160	15	30	14	23	23	21	13	21	320
Dual	320	55	34	42	37	46	38	40	28	640
Total	480	70	64	56	60	69	59	53	49	960

Target spatial distributions



Target elevation

Query nodes in each elevation group (%)

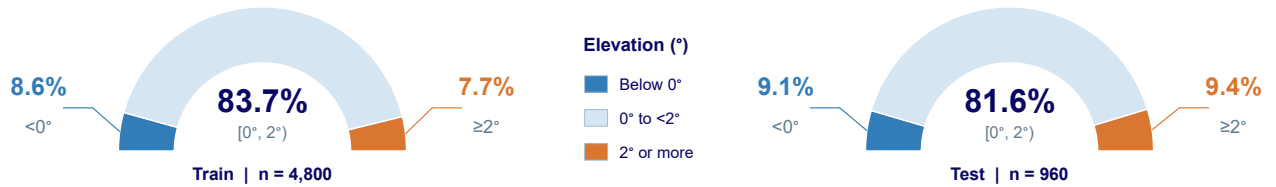


Figure A.2: Target geometry at query times. Distributions use 4,800 training and 960 test query times, expressed as percentages within each split. The upper panels group azimuth into 45° bins starting at 0° and distance into 0.5 m bands from 1.5 to 4.0 m. The lower panels show elevation below 0° , in $[0^\circ, 2^\circ)$, and at least 2° .

B.1 Source-State Inference

Acoustic feature extraction. The physical branch applies a 400-point short-time Fourier transform with a 400-sample Hann window and a 160-sample hop. For directional channel $d \in \{X, Y, Z\}$, F_d and F_W denote its spectrum and the omnidirectional spectrum at the same time–frequency bin. The cross-spectrum and intensity components are

$$\begin{aligned} C_d &= F_W^* F_d, \\ A_d &= \text{clip}\left(\frac{\text{Re } C_d}{|F_W|^2}, -10, 10\right), \\ R_d &= \text{clip}\left(\frac{\text{Im } C_d}{|F_W|^2}, -10, 10\right). \end{aligned} \quad (\text{B.1})$$

Coherence uses averaging over seven adjacent analysis frames, denoted by $\mathcal{A}$:

$$\gamma_d = \frac{|\mathcal{A}(C_d)|}{\sqrt{\mathcal{A}(|F_W|^2)\mathcal{A}(|F_d|^2)}}. \quad (\text{B.2})$$

A fixed 64-band Mel filterbank maps four $\log(1 + |F_c|)$ spectra and three components each of active intensity, reactive intensity, and coherence to thirteen feature channels. Coherence is clipped to $[0, 1]$ before and after Mel projection; numerical normalizations use $\epsilon = 10^{-8}$. Resampling to 121 steps and a convolutional network ($13 \rightarrow 96 \rightarrow 192 \rightarrow 192$) produce 64 physical tokens of width $d_e = 192$ per step. All kernels are 3×3 with unit stride and padding one; the first two layers use eight-group normalization and GELU.

A shared 1D CNN separately encodes each waveform channel (Table B.1). Following Neural IV [9], elementwise products of encoded W with each of X, Y, Z are concatenated into 576 features. A two-layer mapping ($576 \rightarrow 384 \rightarrow 192$, with GELU) combines these interactions. Temporal interpolation and a learned branch embedding produce one neural token per step.

Table B.1: Shared waveform encoder. Each convolution uses one-group normalization and GELU.

Layer	Channels	Kernel	Stride
1	$1 \rightarrow 48$	10	5
2	$48 \rightarrow 96$	7	4
3	$96 \rightarrow 128$	5	4
4	$128 \rightarrow 192$	5	4

A frozen BEATs encoder [40] extracts patch-level semantic features from the omnidirectional W channel. Layer normalization and a linear projection map them from 768 to 192 dimensions. A learned query pools the patches through four-head attention and output normalization to obtain one clip-level semantic token. At each step, this shared token joins the 64 physical tokens and one neural token to form $E_t \in \mathbb{R}^{66 \times 192}$.

Recurrent state estimation and refinement. Two source slots and one background slot maintain 256-dimensional states, starting from learned initial vectors. At each step, a shared MLP ($256 \rightarrow 512 \rightarrow 256$, with GELU) predicts a state change. Residual addition and layer normalization produce the prior $\bar{z}_t$. Linear projections map priors to queries and acoustic tokens to keys and values, all of width 256. Following Slot Attention [43], scaled dot-product scores undergo softmax across slots, followed by normalization over tokens within each slot. The weighted value average forms the observation o_t . A 256-dimensional GRU cell [44] combines o_t with $\bar{z}_t$, followed by layer normalization, to obtain z_t for the next step. The two source sequences continue to refinement.

A shared one-layer bidirectional GRU processes each source sequence, with 256 hidden units per direction. Its forward and backward outputs are concatenated and projected to temporal context $c_t \in \mathbb{R}^{256}$. A linear layer and sigmoid compute a gate g_t from $[z_t; c_t]$. The refined state is $h_t = \text{LayerNorm}(z_t + g_t \odot c_t)$. Four linear heads predict unit direction $\hat{u}_t$, log-distance $\hat{d}_t$, source presence, and positional log-variance from h_t . Direction is unit-normalized, and log-distance is clipped to $[\log 0.05, \log 20]$ before exponentiation. The receiver-centered position is $\hat{p}_t = \exp(\hat{d}_t) \hat{u}_t$. Positional log-variance is clipped to $[-6, 4]$.

Shared prediction heads supervise refined trajectories and transition-prior geometry. For masked-state prediction, the recurrent module processes masked sequences in both temporal directions, and the heads decode the center states before bidirectional refinement. The objectives are defined in Appendix C.

B.2 Spatial Language Modeling

Spatial projector. For each source and time step, the projector concatenates the prior $\bar{z}_t$, observation o_t , recurrent state z_t , refined state h_t , and difference $o_t - \bar{z}_t$ with the additional features in Table B.2. The direction anchor pools active-intensity directions using the source-assignment weights. A frame MLP maps the 1,295-dimensional input to 512 dimensions. Adding a learned source-slot embedding and sinusoidal time encoding forms the frame sequence used by both aggregation branches.

Table B.2: Projector input per source and time step.

Input	Dimensions
Prior, observation, recurrent state, refined state, observation minus prior	5×256
Direction anchor; predicted direction and distance	$3 + 3 + 1$
Source-presence probability; positional log-variance	$1 + 1$
Auxiliary features	6
Total	1,295

The local branch concatenates non-overlapping groups of four consecutive frame features and maps each group to one token, producing 31 local tokens per source. In parallel, the summary branch uses two learned queries, each augmented with the source-slot embedding, to attend to all 121 frame features through a two-layer, eight-head Transformer decoder with pre-layer normalization and a 2,048-dimensional feed-forward network using GELU. Its outputs are two source-summary tokens. The query parameters and decoder weights are shared across sources.

Both branches produce 512-dimensional tokens. Their outputs are concatenated across the two sources, giving 66 spatial tokens. An output MLP maps these tokens to the language embedding width $d_\ell = 3584$. The language model reads them alongside native W -channel audio tokens and the question tokens. Table B.3 gives the MLP configurations.

Table B.3: Projector MLPs. Each uses layer normalization at its input and output, with SiLU between the two linear layers.

Module	Linear widths
Frame encoder	$1295 \rightarrow 512 \rightarrow 512$
Local temporal compressor	$2048 \rightarrow 2048 \rightarrow 512$
Language-space projector	$512 \rightarrow 2048 \rightarrow 3584$

Algorithm 1 Training of SonicWorld

Input: Source-trajectory data $\mathcal{D}_S$ and spatial QA data $\mathcal{D}_Q$.
Parameters: source-state model θ_S ; projector θ_P ; alignment heads θ_A ; LoRA θ_L .
Source-state pretraining

- 1: Freeze pretrained BEATs and set V to 4. X denotes FOA audio and Y the corresponding ground-truth source trajectories.
- 2: **for** each minibatch $(X, Y) \in \mathcal{D}_S$ **do**
- 3: $E \leftarrow \text{AcousticFeatures}(X; \theta_S)$
- 4: $\hat{Y} \leftarrow \text{SourceStates}(E; \theta_S)$
- 5: $\pi \leftarrow \text{Hungarian}(\hat{Y}, Y)$
- 6: $\mathcal{L}_{\text{traj}} = \text{TrajectoryLoss}(\hat{Y}, Y; \pi)$ ▷ Eq. 13
- 7: $\mathcal{L}_{\text{mask}} = \text{MaskedLoss}(E, Y; \theta_S, V)$ ▷ Appendix C.2
- 8: $\mathcal{L}_{\text{state}} = \mathcal{L}_{\text{traj}} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}$
- 9: $\theta_S \leftarrow \text{AdamW}(\theta_S, \nabla_{\theta_S} \mathcal{L}_{\text{state}})$
- 10: **end for**
- Stage I: projector alignment**
- 11: Freeze θ_S and original LLM weights. Set λ_P to 0.5; freeze θ_L .
- 12: Let q denote the question text, y the ground-truth answer, H the source-state features, G the geometric predictions of the frozen source-state model, Z the spatial tokens for the LLM.
- 13: **for** each minibatch $(X, q, y) \in \mathcal{D}_Q$ **do**
- 14: $(H, G) \leftarrow \text{SourceFeatures}(X; \theta_S)$
- 15: $(Z, \hat{G}) \leftarrow \text{SpatialProjector}(H, G; \theta_P, \theta_A)$ ▷ Appendix B.2
- 16: $\mathcal{L}_{\text{QA}} = \text{CrossEntropy}(\text{LLM}_{\theta_L}(X, q, Z), y)$ ▷ Appendix C.3
- 17: $\mathcal{L}_{\text{align}} = \text{AlignLoss}(\hat{G}, G)$ ▷ Eq. C.8
- 18: $\mathcal{L}_{\text{LLM}} = \mathcal{L}_{\text{QA}} + \lambda_P \mathcal{L}_{\text{align}}$
- 19: $(\theta_P, \theta_A) \leftarrow \text{AdamW}((\theta_P, \theta_A), \nabla_{(\theta_P, \theta_A)} \mathcal{L}_{\text{LLM}})$
- 20: **end for**
- Stage II: projector and LoRA adaptation**
- 21: Set λ_P to 0.25 and enable θ_L updates.
- 22: **for** each minibatch $(X, q, y) \in \mathcal{D}_Q$ **do**
- 23: Repeat lines 14–18.
- 24: $(\theta_P, \theta_A, \theta_L) \leftarrow \text{AdamW}((\theta_P, \theta_A, \theta_L), \nabla_{(\theta_P, \theta_A, \theta_L)} \mathcal{L}_{\text{LLM}})$
- 25: **end for**
- 26: **return** $\theta_S, \theta_P, \theta_A, \theta_L$.

C. Training Objectives and Optimization

Algorithm 1 summarizes source-state pretraining and the two LLM fine-tuning stages. The trainable source-state parameters θ_S include the physical and neural feature branches, semantic projection, recurrent slots, transition MLP, bidirectional refinement, and prediction heads. The pretrained BEATs encoder remains frozen.

Table C.1: Loss coefficients.

Objective	Weights
Trajectory supervision	Direction 1; log-distance 1; presence 1; uncertainty 0.05; prior 0.20
Masked center prediction	Direction 1; log-distance 0.3; presence 0.2
Projector training	Answer cross-entropy 1; geometric alignment 0.5
Projector + LoRA	Answer cross-entropy 1; geometric alignment 0.25

C.1 Trajectory Matching and Supervision

Let p denote a ground-truth position, $r = \|p\|_2$ its distance, $u = p/r$ its unit direction, and a the source-presence label. For predicted slot k and annotated source s , the matching cost is

$$C_{k,s} = \overline{1 - \hat{u}_k^\top u_s} + \overline{\text{SL1}(\hat{d}_k - \log r_s)} + 0.25 \text{ BCE}(\hat{a}_k, a_s). \quad (\text{C.1})$$

The bars average geometry errors over frames where the source is present; binary cross-entropy averages over the whole clip. One Hungarian assignment per clip associates slots with trajectories [51]. Unmatched slots receive zero-presence targets.

For matched present-source frames indexed by $\mathcal{A}$, let $d_i = \log r_i$ and let s_i denote predicted positional log-variance [52]. Trajectory supervision comprises

$$\mathcal{L}_{\text{dir}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} (1 - \hat{u}_i^\top u_i), \quad (\text{C.2})$$

$$\mathcal{L}_{\text{dist}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \text{SL1}(\hat{d}_i - d_i), \quad (\text{C.3})$$

$$\mathcal{L}_{\text{presence}} = \frac{1}{BTK} \sum_{b,t,k} \text{BCE}(\hat{a}_{btk}, a_{btk}), \quad (\text{C.4})$$

$$\mathcal{L}_{\text{uncertainty}} = \frac{1}{2|\mathcal{A}|} \sum_{i \in \mathcal{A}} [e^{-s_i} \|\hat{p}_i - p_i\|_2^2 + s_i], \quad (\text{C.5})$$

$$\mathcal{L}_{\text{prior}} = \frac{1}{2|\mathcal{A}|} \sum_{i \in \mathcal{A}} \left[1 - (\hat{u}_i^{\text{prior}})^\top u_i + \text{SL1}(\hat{d}_i^{\text{prior}} - d_i) \right]. \quad (\text{C.6})$$

Smooth-L1 is $\text{SL1}(x) = x^2/2$ for $|x| < 1$ and $|x| - 1/2$ otherwise. BCE denotes binary cross-entropy for source-presence prediction. The five terms are combined into $\mathcal{L}_{\text{traj}}$ with the coefficients in Table C.1.

C.2 Masked State Prediction

We generate $V = 4$ masked feature sequences per clip. For each sequence, we sample an interval length uniformly from five to ten state steps and select its start with a six-step boundary margin. All tokens within the interval are replaced with a learned mask embedding.

The shared recurrent filter processes each sequence v in both temporal directions, $\delta \in \{\rightarrow, \leftarrow\}$. At the masked interval’s center, shared heads predict direction, log-distance, and presence from the recurrent states

before bidirectional refinement. These predictions are matched to the center annotations independently for each (v, δ) . We average the resulting center-prediction losses $\ell_{v, \delta}$:

$$\mathcal{L}_{\text{mask}} = \frac{1}{2V} \sum_{v=1}^V \sum_{\delta \in \{\rightarrow, \leftarrow\}} \ell_{v, \delta}, \quad (\text{C.7})$$

$$\ell_{v, \delta} = \mathcal{L}_{\text{dir}}^{v, \delta} + 0.3 \mathcal{L}_{\text{dist}}^{v, \delta} + 0.2 \mathcal{L}_{\text{presence}}^{v, \delta}.$$

C.3 Two-stage LLM Fine-Tuning

After source-state pretraining, we train the spatial projector and its alignment heads in Stage I. In Stage II, we additionally train LoRA parameters in the LLM’s attention layers. The source-state model, native audio encoder, and original LLM weights remain frozen in both stages. Both stages optimize cross-entropy on answer tokens together with geometric alignment. Alignment heads predict direction, log-distance, and presence from projector frame features after source and time encoding and before temporal compression. Table C.2 lists the intermediate representations.

P and $*$ denote alignment-head predictions and geometric predictions from the frozen source-state model, respectively. We set $w_i = \max(\hat{a}_i^*, 0.05)$. The alignment loss is

$$\begin{aligned} \mathcal{L}_{\text{align}} = & \frac{\sum_i w_i (1 - (\hat{u}_i^P)^\top \hat{u}_i^*)}{\sum_i w_i} \\ & + 0.5 \text{ mean}_i \text{SL1}(\hat{d}_i^P - \hat{d}_i^*) \\ & + 0.25 \text{ mean}_i \text{BCE}(\hat{a}_i^P, \hat{a}_i^*). \end{aligned} \quad (\text{C.8})$$

Here, i indexes clip, time-step, and source-slot combinations, and $\hat{a}_i^*$ is the target source-presence probability. The overall objective is $\mathcal{L}_{\text{QA}} + \lambda_P \mathcal{L}_{\text{align}}$, with stage-specific weights in Table C.1.

Table C.2: Model representations from input to output. B denotes batch size. Slot–token weights include two source slots and one background slot; subsequent source representations retain only the two source slots.

Representation	Shape
FOA waveform, $[W, Y, Z, X]$	$[B, 4, 160000]$
Physical input features	$[B, 13, 64, 121]$
Physical tokens	$[B, 121, 64, 192]$
Neural tokens / semantic tokens	$[B, 121, 1, 192]$
Combined acoustic tokens	$[B, 121, 66, 192]$
Slot–token weights	$[B, 3, 66]$
Retained source assignments	$[B, 121, 2, 66]$
Prior / observation / recurrent / refined states	$[B, 121, 2, 256]$
Direction / Cartesian position	$[B, 121, 2, 3]$
Distance / presence / log-variance	$[B, 121, 2]$
Projector frame features	$[B, 121, 2, 512]$
Local tokens	$[B, 2, 31, 512]$
Summary tokens	$[B, 2, 2, 512]$
Spatial tokens for the LLM	$[B, 66, 3584]$

C.4 Optimization Settings

The main experiments use the 2,400-clip training split and its 24,000 questions. All stages use AdamW [53], weight decay 0.01, cosine learning-rate decay, 5% learning-rate warmup, bfloat16, gradient clipping at norm 1, and a micro-batch size one. Table C.3 lists the stage-specific settings.

Table C.3: SonicWorld training stages. Updates count optimizer steps.

Stage	Epochs	Accum.	Updates	Trainable modules and peak learning rate
Source states	25	2	30,000	Physical and neural branches, semantic pooling, recurrent filter, bidirectional GRU, state heads: 10^{-4}
Projector	5	8	15,000	Projector and alignment heads: 10^{-4}
Projector + LoRA	3	8	9,000	Projector : 3×10^{-5} ; LoRA: 5×10^{-5}
GPUs				$3 \times$ NVIDIA RTX 4090; $6 \times$ NVIDIA RTX 3090

LoRA modifies the attention query, key, value, and output projections with rank 16, scaling 32, and dropout 0.05 [45]. Each backbone retains its native frozen audio encoder and uses five projector epochs followed by three joint projector–LoRA epochs.

D. Supplementary Results

Table D.1: Effect of SonicWorld on different backbones on Sonic-Bench. Values in parentheses indicate changes relative to the corresponding original backbone without SonicWorld.

Backbone	Dir.↑	Az.↓	El.↓	Dist.↓	Mot.↑
Single source					
Qwen2.5-Omni-7B	50.00 (+37.50)	30.85 (−60.33)	0.91 (−84.00)	0.56 (−98.27)	52.50 (+0.62)
InternOmni	51.88 (+39.07)	36.49 (−57.23)	0.92 (−21.73)	0.61 (−39.97)	55.94 (+47.81)
Phi-4-MM-5.6B	48.75 (+35.31)	32.03 (−49.18)	0.92 (−2.37)	0.53 (−1.40)	53.44 (+47.19)
Dual source					
Qwen2.5-Omni-7B	27.03 (+14.37)	56.59 (−31.74)	0.93 (−82.65)	0.60 (−33.70)	53.91 (+5.32)
InternOmni	30.16 (+17.82)	62.49 (−30.41)	0.93 (−31.58)	0.62 (−42.54)	52.19 (+46.25)
Phi-4-MM-5.6B	30.78 (+18.12)	60.50 (−19.40)	0.95 (−3.18)	0.60 (−1.66)	52.03 (+46.72)

D.1 Evaluation Protocol and Metrics

Metrics details. We use the metrics defined in Section 5.2. MAE is computed over finite parsed numerical responses.

D.2 Effect of SonicWorld on Different Backbones

On Sonic-Bench, SonicWorld improves all five metrics in both source settings relative to the original Qwen2.5-Omni-7B, InternOmni, and Phi-4-MM-5.6B backbones (Table D.1). Direction accuracy gains range from 35.31–39.07 percentage points for single sources and 14.37–18.12 points for dual sources.

On SLS (Table D.2), SonicWorld increases direction accuracy by 54.42–61.29 percentage points and reduces azimuth MAE by 67.49–71.21° across the three matched backbones. On SpatialSceneQA, all three variants reduce azimuth MAE; elevation MAE decreases for Qwen2.5-Omni-7B and InternOmni but increases by 2.03° for Phi-4-MM-5.6B.

Table D.2: Effect of SonicWorld on zero-shot transfer across backbones. Values in parentheses indicate changes relative to the corresponding original backbone without SonicWorld.

Backbone	SpatialSceneQA			SLS		
	Az.↓	El.↓	Dir.↑	Az.↓	El.↓	Dist.↓
Qwen2.5-Omni-7B	21.35 (−39.12)	20.51 (−59.63)	73.98 (+61.29)	19.29 (−70.97)	7.09 (−81.38)	0.89 (−54.16)
InternOmni	22.57 (−57.97)	20.58 (−57.47)	66.80 (+54.42)	22.20 (−67.49)	7.08 (−50.51)	1.24 (−97.59)
Phi-4-MM-5.6B	19.79 (−42.07)	20.34 (+2.03)	73.44 (+60.83)	18.22 (−71.21)	7.95 (+0.87)	0.81 (−1.18)

D.3 Localization Tolerance Accuracy

Table D.3 shows that SonicWorld achieves the highest localization tolerance accuracies in both source settings. Az20 exceeds Spatial-Omni by 22.82 percentage points on single-source clips and 4.53 points on dual-source clips. From single to dual sources, SonicWorld’s Az20 drops from 54.38% to 25.16%, while El1 remains near 81% and D0.5 stays at 53.75%, suggesting that source interference particularly affects precise horizontal localization. The concentrated elevation distribution (Table A.6) may partly explain the stable El1.

On El1, SonicWorld exceeds BAT by 0.93 percentage points for single sources and 0.94 points for dual sources. Its D0.5 accuracy is 53.75% in both settings, compared with 52.81% for Neural-IV on single-source clips and 50.47% for Classical-IV on dual-source clips.

D.4 Zero-Shot Transfer Protocol and Results

We evaluate the official test splits of SpatialSceneQA [9] and Spatial LibriSpeech (SLS) [50], retaining recordings of at least 10 seconds and using their first 10 seconds at 16 kHz. Table D.4 summarizes their scale and task coverage: SpatialSceneQA tests single- and dual-source angular localization, while SLS covers four single-source spatial attributes. Both assess static localization transfer.

For SpatialSceneQA, we split each joint azimuth/elevation query into two scalar questions, preserving the target and labels. For SLS, we construct direction, azimuth, elevation, and distance questions from speech-source annotations, converting angles to degrees and assigning direction labels via Equation A.3. SpatialSceneQA channels are mapped as $(W, Y, Z, X) = (c_0, -c_3, c_1, -c_2)$; SLS already uses this order. Spatial encoders receive all four channels, while native audio encoders and direct-audio baselines receive the same crop’s W channel. Scoring uses the Sonic-Bench answer parser.

Table D.5 reports SpatialSceneQA results with Qwen2.5-Omni-7B as SonicWorld’s backbone. On dual-source clips, adaptation reduces azimuth MAE from 86.77° to 24.69° and raises Az20 from 23.20% to 51.73%. On single-source clips, furthermore, azimuth MAE decreases from 20.54° to 16.28°, showing that the azimuth transfer gains extend to both single- and dual-source settings. Aggregate external results appear in Table 2; backbone comparisons are provided in Appendix D.2.

Table D.3: Localization tolerance accuracies. Spatial baselines and SonicWorld use Qwen2.5-Omni-7B. Az20, El1, and D0.5 denote azimuth, elevation, and distance accuracies within 20°, 1°, and 0.5 m, respectively.

Model	Single source			Dual source		
	Az20↑	El1↑	D0.5↑	Az20↑	El1↑	D0.5↑
Closed-source models						
Gemini 3 Flash	10.00	30.31	25.94	11.72	27.34	26.41
Gemini 3.1 Pro	12.19	31.88	1.88	12.03	18.59	6.25
Open-source models						
Qwen2.5-Omni-7B	14.06	1.88	0.00	11.72	1.09	0.00
InternOmni	9.06	23.75	23.44	9.06	22.34	24.69
Phi-4-MM-5.6B	5.00	1.25	4.69	7.03	1.09	1.56
Audio Flamingo 3	9.69	5.63	18.13	10.63	6.09	11.25
Kimi-Audio-7B-Instruct	11.25	43.13	9.06	11.72	40.16	9.84
Spatial baselines						
BAT	10.31	79.38	45.63	11.09	80.47	49.84
Classical-IV	30.63	75.94	50.94	18.91	75.00	50.47
Neural-IV	14.69	76.56	52.81	12.34	74.22	45.00
Spatial-Omni	31.56	67.50	30.31	20.63	72.34	28.13
SONICWORLD	54.38	80.31	53.75	25.16	81.41	53.75

Table D.4: Evaluation-set scale and selection. Sonic-Bench uses its full test set; SpatialSceneQA and SLS use the selected test subsets. Dir., Az., El., Dist., and Mot. denote direction, azimuth, elevation, distance, and movement direction.

	Sonic-Bench	Spatial SceneQA	SLS
Evaluated clips	480	622	8,512
Single-source clips	160	247	8,512
Dual-source clips	320	375	0
Motion state	Stationary / Moving	Stationary	Stationary
Questions per evaluated clip	10	2	4
Single-source questions	1,600	494	34,048
Dual-source questions	3,200	750	0
Total questions	4,800	1,244	34,048
Query types	Dir., Az., El., Dist., Mot.	Az., El.	Dir., Az., El., Dist.

Table D.5: Single- and dual-source SpatialSceneQA results. Spatial baselines and SonicWorld use Qwen2.5-Omni-7B.

Model	Single source				Dual source			
	Az.↓	El.↓	Az20↑	El1↑	Az.↓	El.↓	Az20↑	El1↑
Open-source audio models								
Qwen2.5-Omni-7B	20.54	100.49	53.85	0.00	86.77	66.73	23.20	0.00
InternOmni	75.35	83.90	5.26	0.00	83.89	74.26	4.53	0.00
Phi-4-MM-5.6B	78.49	16.42	10.93	0.40	47.69	21.14	30.13	0.00
Audio Flamingo 3	117.98	27.04	14.17	0.40	145.38	22.64	3.47	0.00
Kimi-Audio-7B-Instruct	53.70	34.07	30.77	0.81	56.48	36.28	29.33	0.00
Spatial baselines								
BAT	117.35	18.01	8.91	0.00	130.65	22.15	0.00	0.00
Classical-IV	25.47	15.78	45.75	1.62	38.88	21.40	22.40	0.00
Neural-IV	90.45	14.61	22.27	2.83	86.94	21.05	20.27	0.00
Spatial-Omni	132.55	21.71	0.40	0.00	92.36	23.59	17.60	0.53
SONICWORLD	16.28	17.84	66.40	0.00	24.69	22.27	51.73	0.00

Table D.6: Results on SLS with official spatial interference. Evaluation covers 8,512 clips and 34,048 questions. Spatial baselines and SonicWorld use Qwen2.5-Omni-7B. Dir. is accuracy (%); Az./El. are MAE in degrees and Dist. is MAE in meters. Best values are bold.

Model	Dir.↑	Az.↓	El.↓	Dist.↓
Open-source models				
Qwen2.5-Omni-7B	12.01	90.68	14.40	19.61
InternOmni	12.76	89.19	54.55	7.12
Phi-4-MM-5.6B	12.48	90.42	7.04	0.96
Audio Flamingo 3	12.31	89.46	16.39	3.11
Kimi-Audio-7B-Instruct	11.83	90.81	8.59	2.81
Spatial baselines				
BAT	12.50	90.06	7.07	0.84
Classical-IV	13.15	77.32	8.07	1.38
Neural-IV	12.91	90.50	8.28	1.34
Spatial-Omni	26.97	71.69	13.88	6.55
SONICWORLD	63.35	26.74	7.08	1.16

D.5 SLS with Spatial Interference

We evaluate robustness to spatial interference using the official paired speech and noise recordings from SLS [50]. Answers follow the speech-source annotations. As shown in Table D.6, SonicWorld improves direction accuracy from 26.97% to 63.35% and reduces azimuth MAE from 71.69° to 26.74° compared with Spatial-Omni. Its elevation MAE (7.08°) is close to the best result (7.04°), while distance MAE (1.16 m) remains above BAT’s 0.84 m. These results show a clear advantage in bearing estimation under spatial interference, with more limited benefits for elevation and distance.

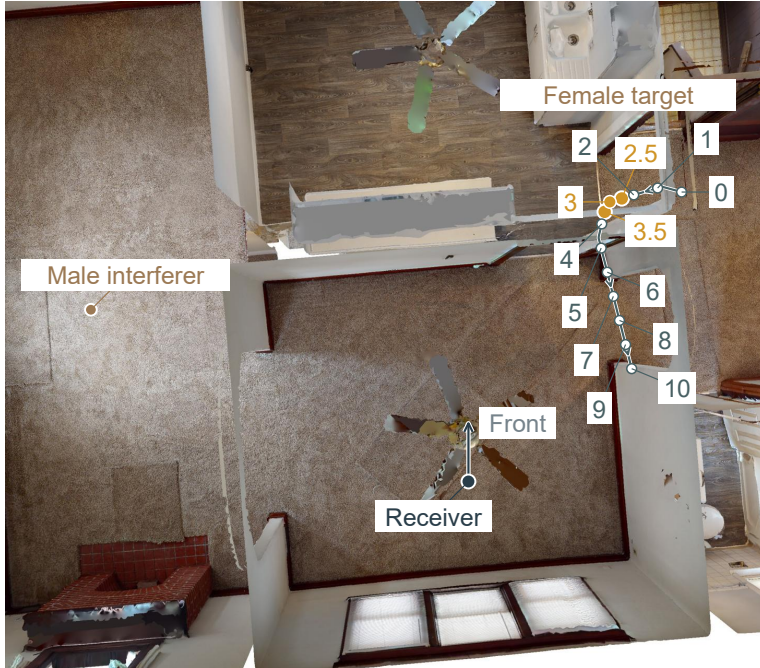
E. Case Studies

In this section, we examine SonicWorld’s motion predictions and decoded source trajectories in single- and dual-source scenes with moving and stationary speakers.

E.1 Prediction Comparison

Figures E.1 and E.2 illustrate target-specific motion reasoning through turns and under interference. SonicWorld correctly predicts Back-Left with a stationary interferer and Back for a male target with a moving interferer, while Spatial-Omni and JAEGER (Classical-IV) return incorrect directions or Stationary.

Moving Target with Stationary Interferer



Which direction did the female voice move during 2.5-3.5 s?

Ground truth: Back-Left

	SonicWorld Back-Left	✓
	Spatial-Omni Stationary	×
	JAEGER(Classical-IV) Left	×

Figure E.1: Movement predictions with a stationary interferer.

E.2 Persistent Source-State Analysis

We next examine the spatial representations underlying motion reasoning. Each source-state sequence is decoded into receiver-centered coordinates and distances at 121 time steps over 0–10 s and compared with

Moving Target with Moving Interferer



Which direction did the male voice move during 3-4 s?

Ground truth: Back

	SonicWorld Back	✓
	Spatial-Omni Right	×
	JAEGER(Classical-IV) Front-Right	×

Figure E.2: Movement predictions for a male target with a moving interferer.

ground truth. For multi-source clips, a single sequence-level assignment matches the source slots to the annotated trajectories.

Following a moving source. In Figure E.3, the predicted front coordinate decreases from positive to negative values, capturing the source’s progression from the receiver’s front side to its rear side. The decoded trajectory has an average displacement error of 0.342 m.

Separating two moving sources. Figure E.4 shows that two ground-truth trajectories exhibit opposing trends in their front coordinates: one decreases over time, while the other increases. The decoded trajectories preserve these respective trends in separate slots.

Distinguishing moving and stationary sources. In Figure E.5, the decoded left coordinate changes substantially for the moving source, whereas the stationary source’s lateral estimate remains nearly constant. These patterns appear in separate source slots before language generation.

E.3 Tracking Across Speech Pauses

We examine whether a source slot remains associated with the same speaker after a speech pause. Both examples contain a moving target that temporarily stops speaking while continuing along its path, together with a stationary or moving interferer. Trajectories are decoded from the complete clip using the source-state model with bidirectional refinement.

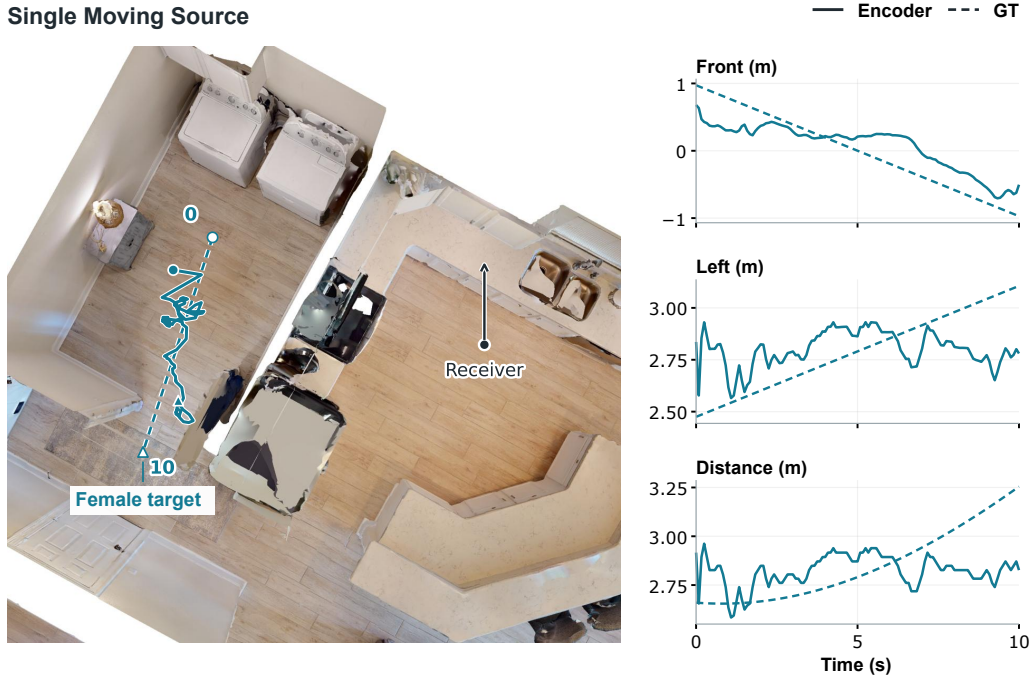


Figure E.3: Decoded states of a single moving source. Solid lines show decoded trajectories and dashed lines show ground truth. The scene view shows the spatial paths; the time series show receiver-centered front and left coordinates and source distance. Circles and triangles mark trajectory starts and ends.

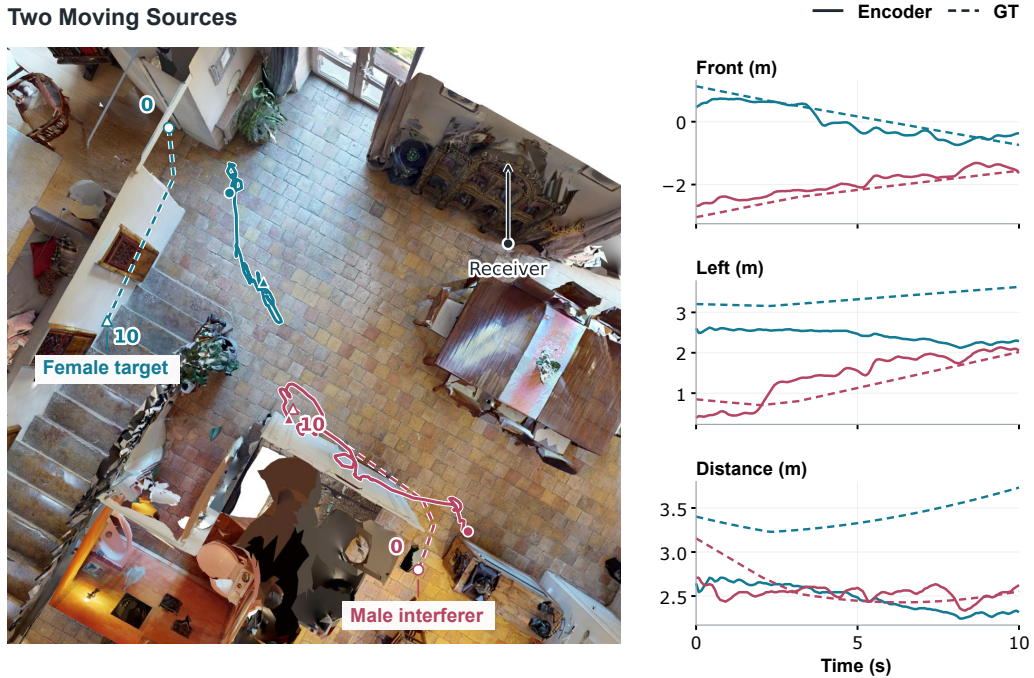


Figure E.4: Decoded states of two moving sources. Colors distinguish the sources. Solid lines show decoded trajectories and dashed lines show ground truth.

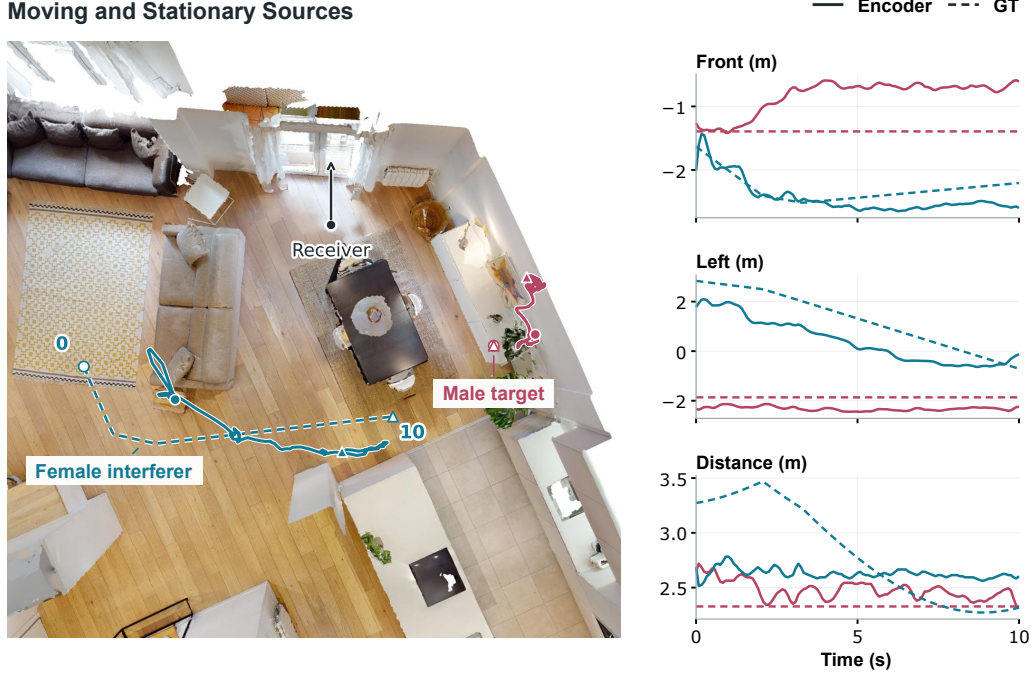


Figure E.5: Decoded states of a moving source and a stationary source. Colors distinguish the sources. Solid lines show decoded trajectories and dashed lines show ground truth.

Table E.1: Source-state continuity across speech pauses. ADE (m) and ID retention (%) are evaluated after speech resumes, with pre-pause slot assignments fixed. ID retention uses distance thresholds of 0.5 m and 0.25 m for stationary and moving interferers, respectively.

Variant	ADE↓	ID retention↑
<i>Stationary interferer</i>		
w/o transition MLP	0.82	0.00
w/o masked-state loss	0.66	9.09
Full SonicWorld	0.52	30.30
<i>Moving interferer</i>		
w/o transition MLP	0.33	14.81
w/o masked-state loss	0.34	25.93
Full SonicWorld	0.32	33.33

Moving Target with Stationary Interferer

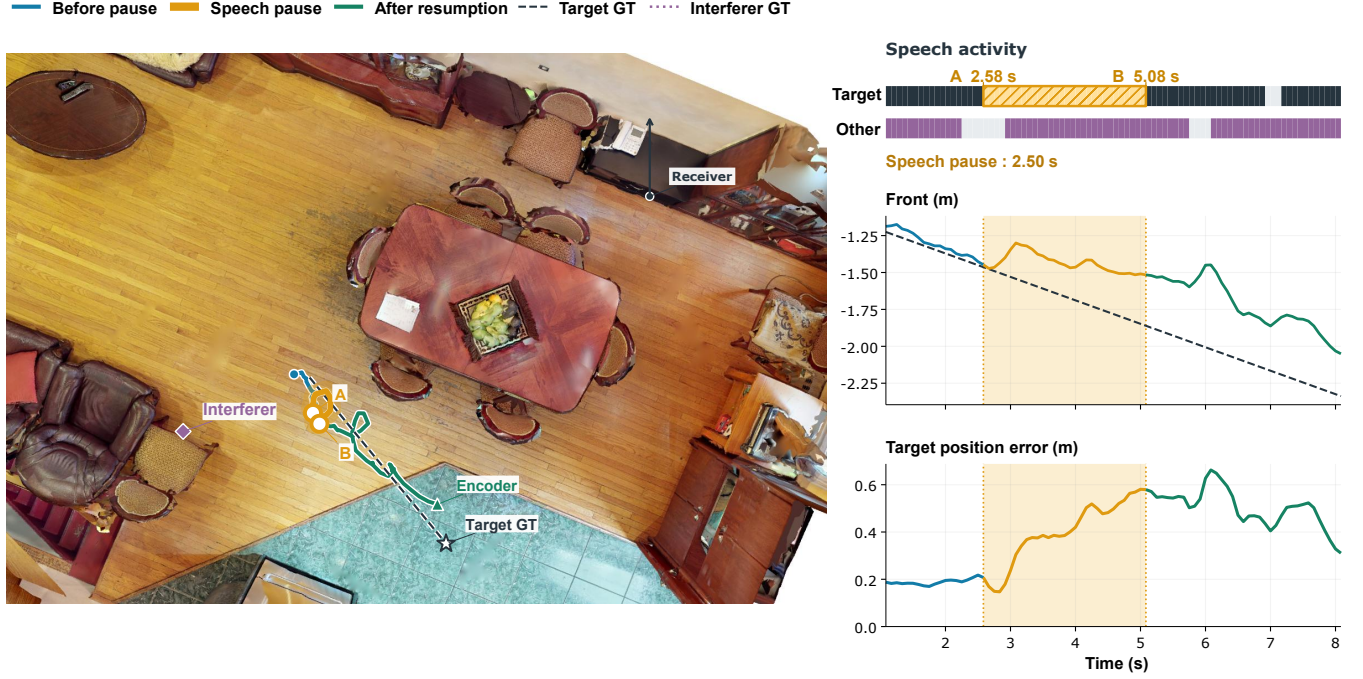


Figure E.6: Tracking across a speech pause with a stationary interferer. Blue/gold/green show the decoded target trajectory before/during/after the pause; A/B mark pause onset/resumption. Dark dashed/purple dotted lines show target/interferer ground truth. Right panels show speech activity, receiver-centered front coordinate, and target position error.

For each case, we assign predicted source slots to ground-truth sources using their speech-active frames in the 1.5 s preceding the pause and keep this assignment fixed after resumption. Let s_0 denote the assigned target slot and $\mathcal{T}$ the target speech-active frames within 3 s after resumption, clipped at the end of the recording. Given the predicted target position $\hat{\mathbf{p}}_{t,s_0}$ and ground-truth position $\mathbf{p}_t$, we compute average displacement error (ADE):

$$\text{ADE} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \|\hat{\mathbf{p}}_{t,s_0} - \mathbf{p}_t\|_2.$$

We additionally report ID retention, where $r_t = 1$ if the target remains uniquely matched to the assigned slot at frame t , and $r_t = 0$ otherwise:

$$\text{ID retention}(\%) = \frac{100}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} r_t.$$

Frame-level matching uses a distance threshold of 0.5 m and 0.25 m for stationary and moving interferers, respectively.

Table E.1 shows that the full model achieves the lowest post-resumption ADE and highest ID retention in both examples. Removing either component substantially worsens ADE with a stationary interferer, while a moving interferer mainly reveals differences in ID retention. These case-level results support the role of the transition MLP and masked-state loss in maintaining source-state continuity across speech pauses.

Moving Target with Moving Interferer

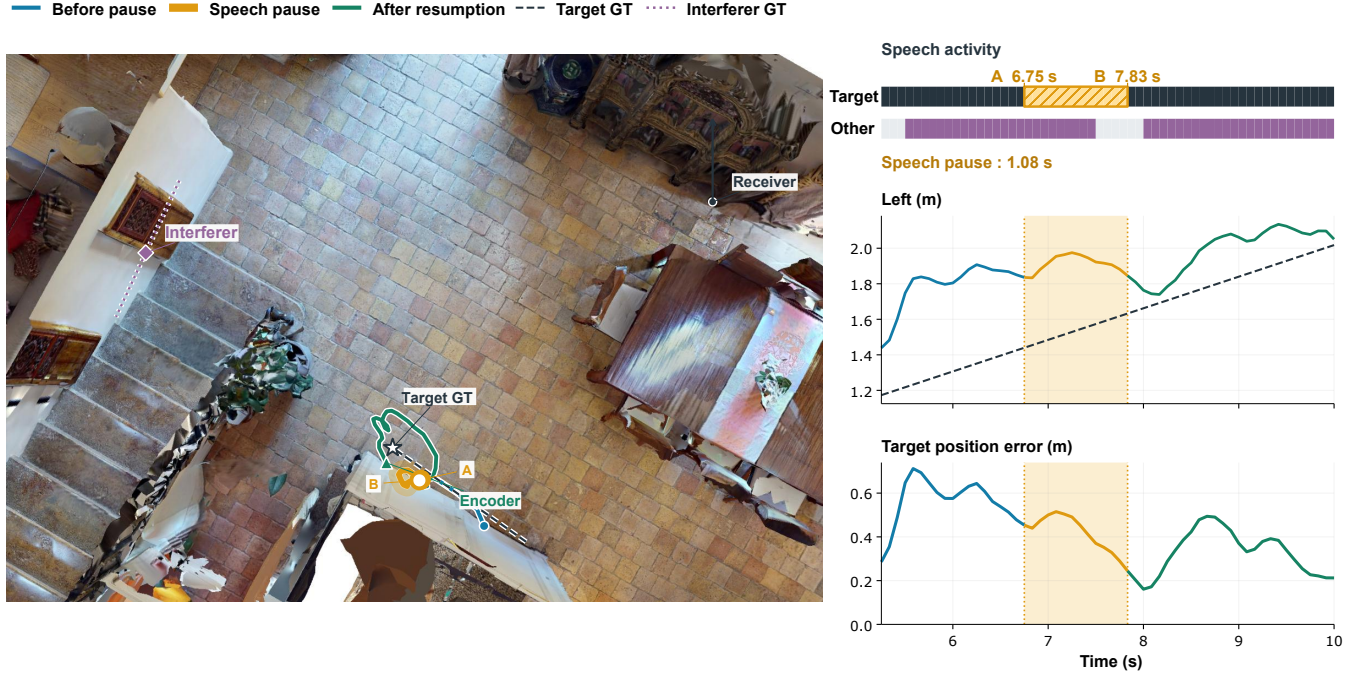


Figure E.7: Tracking across a speech pause with a moving interferer. Both speakers continue moving during the target’s 1.08 s speech pause. Colors and markers follow Figure E.6. The coordinate panel shows the receiver-centered left position.

Stationary interferer. In Figure E.6, the target pauses from 2.58 to 5.08 s while the interferer remains stationary. After speech resumes, the slot matched before the pause follows the target’s decreasing front coordinate. The decoded trajectory retains a positional lag, with a post-resumption ADE of 0.52 m. Thus, the same slot captures the target’s continuing front-back movement after the 2.50 s speech interruption.

Moving interferer. In Figure E.7, both speakers move, and the target pauses from 6.75 to 7.83 s. Following this 1.08 s interruption, the same slot captures the target’s overall increase in the left coordinate. Its post-resumption ADE is 0.32 m, evaluated on target-active frames during the remaining 2.17 s of the clip.

In both examples, the slot matched to the target before the pause remains closer to the target than to the interferer at every evaluated post-resumption frame. These observations illustrate continuity of the target-associated spatial readout across interrupted speech.

F. Questions and Prompts

This section presents the task instructions and complete question templates used for Sonic-Bench and the two external evaluations.

F.1 Sonic-Bench

The five question templates below identify the target voice and include the required answer format. The placeholders {gender}, {time}, {start}, and {end} are instantiated for each question, with times

written to two decimal places and the movement interval spanning one second. SonicWorld places its input instruction after the audio and <world> spatial-token block and before the task-specific question.

SonicWorld input instruction for Sonic-Bench

You are an audio spatial reasoning assistant. Use the raw mono audio semantics and the complete source-centric dynamic acoustic world enclosed by <world> tags. The world contains persistent anonymous source slots across the full clip. Use the question's source identity, timestamp, and interval to select the relevant slot. Follow the requested answer format exactly and output no explanation.

Sonic-Bench question: Position direction

At {time} seconds, which direction is the {gender} voice coming from? Answer with exactly one label from: Front, Front-Left, Left, Back-Left, Back, Back-Right, Right, Front-Right.

Sonic-Bench question: Azimuth

At {time} seconds, what is the horizontal azimuth of the {gender} voice relative to the listener? Answer exactly as a number in degrees.

Sonic-Bench question: Elevation

At {time} seconds, what is the elevation of the {gender} voice relative to the listener? Answer exactly as a number in degrees.

Sonic-Bench question: Distance

At {time} seconds, how far away is the {gender} voice? Answer exactly in this format: distance: <meters> m.

Sonic-Bench question: Movement direction

Between {start} and {end} seconds, which direction did the {gender} sound source move relative to the listener's fixed orientation? Answer with exactly one label from: Front, Front-Left, Left, Back-Left, Back, Back-Right, Right, Front-Right, Stationary.

F.2 Spatial LibriSpeech Questions

SLS provides spatial annotations rather than question-answer pairs; we construct the following four questions for each retained single-source speech clip.

SLS question: Position direction

Which direction is the sound source coming from? Answer with exactly one label from: Front, Front-Left, Left, Back-Left, Back, Back-Right, Right, Front-Right.

SLS question: Azimuth

What is the horizontal azimuth of the sound source relative to the listener? Answer exactly as a number in degrees.

SLS question: Elevation

What is the elevation of the sound source relative to the listener? Answer exactly as a number in degrees.

SLS question: Distance

How far away is the sound source from the listener? Answer exactly in this format:
distance: <meters> m.

F.3 SpatialSceneQA Questions

Azimuth and elevation are requested separately for both source-count subsets. In dual-source clips, {target_description} is “male voice” or “female voice”, as specified by the original question.

SpatialSceneQA question: Single-source azimuth

What is the horizontal azimuth of the sound source relative to the listener? Answer exactly as a number in degrees.

SpatialSceneQA question: Single-source elevation

What is the elevation of the sound source relative to the listener? Answer exactly as a number in degrees.

SpatialSceneQA question: Dual-source azimuth

What is the horizontal azimuth of the {target_description} relative to the listener? Answer exactly as a number in degrees.

SpatialSceneQA question: Dual-source elevation

What is the elevation of the {target_description} relative to the listener? Answer exactly as a number in degrees.