ORBITRIR: FRAME-AVERAGED FEW-SHOT RIR SYNTHESIS FOR 3D SOUND SOURCE LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied spatial reasoning requires agents to infer 3D world states from partial sensory observations. Among these observations, room acoustics provides a particularly informative spatial signal, as Room Impulse Responses (RIRs) encode interactions between source–receiver geometry and the surrounding environment. Recent few-shot acoustic models predict RIRs from spatial geometry and sparse room context. However, existing few-shot acoustic generators lack architectural inductive biases that reflect the known receiver-yaw invariance of omnidirectional monaural RIRs. To this end, we introduce ORBITRIR, a geometry-constrained generative acoustic model incorporating this inductive bias through Frame Averaging over a receiver-yaw orbit. We evaluate ORBITRIR on few-shot RIR synthesis tasks and downstream sound source localization tasks through analysis-by-synthesis. On AcousticRooms, ORBITRIR outperforms its geometry-unconstrained counterpart. When inverted for localization, it also improves localization accuracy over learned and physics-based baselines.

1 INTRODUCTION

Embodied spatial reasoning requires agents to infer 3D world state from partial sensory observations. Among these observations, room acoustics provides a particularly informative cue to spatial structure, as Room Impulse Responses (RIRs) encode interactions between source–receiver geometry and the surrounding environment. This spatial information has been exploited in robot learning to recover scene geometry from ambient sound (Chen et al., 2022), localize quietly moving people with passive audio (Yang et al., 2024), and find humans from changes in room acoustics (Wang et al., 2023). In parallel, recent acoustic synthesis methods learn the corresponding forward mapping, predicting RIRs from source–receiver geometry and sparse scene context (Majumder et al., 2022; Liu et al., 2025; Wang et al., 2024; Brunetto, 2026). This raises a natural inverse question: can the spatial structure learned for forward acoustic prediction support 3D spatial reasoning without training a localization-specific model? We study this question through 3D sound source localization from monaural RIR observations.

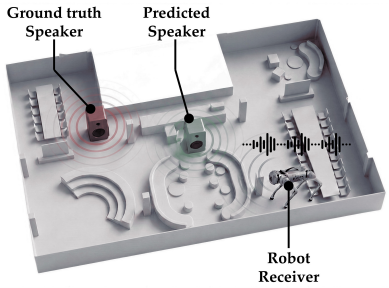

Figure 1: **Spatial reasoning for 3D sound source localization.** A robot localizes an unknown source by comparing its observed RIR against acoustic responses predicted for candidate 3D locations.

Two limitations arise at different stages of the forward-to-inverse pipeline. At the forward-model level, recent few-shot generative models such as FLAC (Brunetto, 2026) capture acoustic uncertainty under sparse room context, but lack architectural inductive biases that enforce known physical symmetries. At the localization level, learned methods (Majumder et al., 2022) based on predicted RIRs remain limited by the fidelity of the underlying acoustic model, while physics-based inversion requires accurate scene models and expensive numerical simulation (Dokmanić & Vetterli, 2012). Together, these limitations motivate a geometry-constrained generative acoustic model that improves forward RIR synthesis and can be effectively inverted for source localization.

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.

We address these limitations with ORBITRIR, a geometry-constrained flow-matching model for few-shot RIR synthesis. For an omnidirectional monaural receiver, rotating the receiver in yaw does not change the RIR. Frame Averaging (FA) provides a principled framework for enforcing such known symmetries in otherwise unconstrained neural architectures (Puny et al., 2022; Atzmon et al., 2022). In ORBITRIR, we average the conditioning representation over the receiver-yaw orbit of equivalent orientations, so that they yield the same conditional RIR distribution. This turns yaw invariance enforced by structure rather than leaving it to be learned from data. By removing non-physical orientation-dependent variation, the constraint improves RIR prediction fidelity and generalization.

ORBITRIR builds on the conditional generative formulation of FLAC (Brunetto, 2026), which models a distribution over plausible RIRs from sparse room context. FA constrains what should remain invariant, while the generative formulation preserves acoustic uncertainty under sparse observations. We further test whether this geometry-constrained forward model supports inverse spatial inference by freezing ORBITRIR and applying analysis-by-synthesis (Yuille & Kersten, 2006). For each candidate source location, we generate plausible RIRs and evaluate their compatibility with the observed response. No localization-specific training or fine-tuning is used; we directly evaluate the utility of the generated RIRs for downstream 3D sound-source localization.

In summary, our main contributions are twofold:

- We introduce ORBITRIR, a geometry-constrained generative acoustic model that incorporates Frame Averaging over the receiver-yaw orbit into few-shot RIR synthesis, turning a known physical symmetry from a property to be learned from data into a hard constraint enforced by construction.
- We invert the geometry-constrained generator for 3D sound-source localization through analysis-by-synthesis, outperforming both learned and physics-based baselines without localization-specific training or fine-tuning.
- We conduct extensive experiments on AcousticRooms under sparse and dense acoustic context, demonstrating improved RIR synthesis over FLAC, stronger generalization than yaw augmentation, and consistent performance across equivalent receiver-yaw frames.

2 RELATED WORK

Forward few-shot acoustic synthesis. Neural acoustic fields model continuous RIR fields from dense measurements of a single scene (Luo et al., 2022; Su et al., 2022; Liang et al., 2023; Brunetto et al., 2025; Lan et al., 2024), with some approaches incorporating differentiable acoustic physics (Wang et al., 2024). Few-shot methods instead generalize across environments from sparse acoustic and geometric context, including Few-ShotRIR (Majumder et al., 2022), MAGIC (Huang et al., 2025), xRIR (Liu et al., 2025), and FLAC (Brunetto, 2026). Although xRIR and FLAC explicitly condition on panoramic geometry and source–receiver poses, neither architecture explicitly enforces the receiver-yaw invariance of omnidirectional monaural RIRs, leaving consistency across equivalent heading references to be learned from data. ORBITRIR addresses this gap by applying Frame Averaging over the receiver-yaw orbit to construct invariant geometry conditioning.

Symmetry-informed neural architectures. Known symmetries are commonly encoded through group convolutions and equivariant architectures (Cohen & Welling, 2016; Fuchs et al., 2020; Finzi et al., 2021). This symmetry-aware design has also been adopted in acoustics, where rotation-equivariant spatial-audio models (Sato et al., 2021; Diaz-Guerra et al., 2023) and reciprocity-based acoustic-field reconstruction (Lan et al., 2025) incorporate domain-specific physical symmetries. Rather than redesigning the backbone, Frame Averaging (FA) provides an architecture-agnostic alternative that enforces invariance or equivariance by aggregating symmetry-related frames (Puny et al., 2022; Atzmon et al., 2022; Duval et al., 2023). However, these symmetry principles have not been incorporated into few-shot generative RIR synthesis to enforce consistency across equivalent receiver headings. ORBITRIR fills this gap by applying FA over the receiver-yaw orbit, yielding yaw-invariant geometry conditioning by construction.

Localization from acoustic observations. Classical sound-source localization relies on multi-channel observations, including GCC-PHAT (Knapp & Carter, 1976), MUSIC (Schmidt, 1986), and SRP-PHAT (DiBiase et al., 2001), with learned variants regressing direction of arrival from array features (Adavanne et al., 2019; Grumiaux et al., 2022). More recently, reverberation has

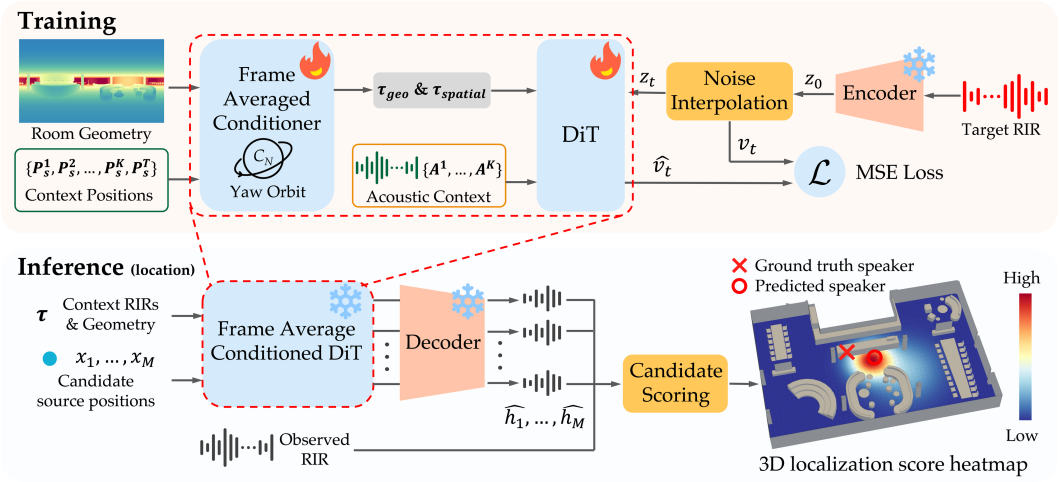

Figure 2: **Overview of ORBITRIR for geometry-constrained RIR synthesis and training-free source localization.** *Top:* During training, ORBITRIR transforms rooms geometry and source–context coordinates over the receiver-yaw orbit and aggregated through FA, enforcing receiver-yaw invariance. Together with the acoustic context, the averaged features condition a DiT trained by latent flow matching on target RIR latents. *Bottom:* At inference, ORBITRIR is frozen and queried at each candidate source position to synthesize candidate-conditioned RIRs. Comparing the generated responses with the observed RIR produces a 3D localization-score heatmap.

enabled localization from a single microphone: multipath methods recover source positions from early reflections given known room geometry (Taghizadeh et al., 2015), while robot-audition systems aggregate reverberation-based distance estimates over microphone motion (Wang et al., 2025). This broader use of reverberation has also supported speaker-distance estimation (Neri et al., 2024), human localization (Wang et al., 2023; Yang et al., 2024), scene-structure recovery (Chen et al., 2022), and inverse reconstruction of spatial layout from sparse RIRs (Wang et al., 2026). Together, these methods demonstrate that room acoustics carries rich spatial information, but localization typically relies on task-specific predictors, explicit physical scene models and numerical simulation, or sensor motion. In contrast, ORBITRIR directly reuses the frozen learned forward model for localization through analysis-by-synthesis (Yuille & Kersten, 2006), avoiding a localization-specific network and expensive acoustic simulation while ranking candidate source locations from a monaural RIR.

3 PROBLEM FORMULATION

We formulate our setting as a coupled forward–inverse problem. The forward problem models the distribution of plausible RIRs for a source–receiver query given sparse observations of an unseen room. The inverse problem holds this learned forward model fixed and identifies the source-location hypothesis that best explains an observed RIR.

Forward Problem: Few-Shot RIR Synthesis. Let $x_s, x_r \in \mathbb{R}^3$ denote the source and receiver positions, respectively, and let $q = (x_s, x_r)$ denote the corresponding query. The observation $h \in \mathbb{R}^{L_h}$ is a monaural, omnidirectional RIR. ORBITRIR is conditioned on the sparse scene context $\mathcal{O} = (\mathcal{A}, \mathcal{P}, G_{x_r})$, where $\mathcal{A} = \{h_1, \ldots, h_{K_{\text{ctx}}}\}$ denotes the reference RIRs and $\mathcal{P} = \{x_s^{(1)}, \ldots, x_s^{(K_{\text{ctx}})}\}$ their corresponding source positions (Brunetto, 2026). Each h_j is measured at the same receiver x_r from $x_s^{(j)}$, and G_{x_r} is a panoramic depth map registered to the receiver coordinate frame. Because this sparse context does not fully specify the room materials and other latent acoustic properties, a query $(q, \mathcal{O})$ need not determine a unique RIR. The forward problem is therefore to learn the conditional distribution $p_\theta(h \mid q, \mathcal{O})$. Furthermore, rotating an omnidirectional monaural receiver about its yaw axis does not change the RIR. Accordingly, the desired forward distribution satisfies

$$p(h \mid q, \mathcal{O}) = p(h \mid R_g q, R_g \mathcal{O}), \qquad g \in \mathcal{G}_{\text{yaw}}, \tag{1}$$

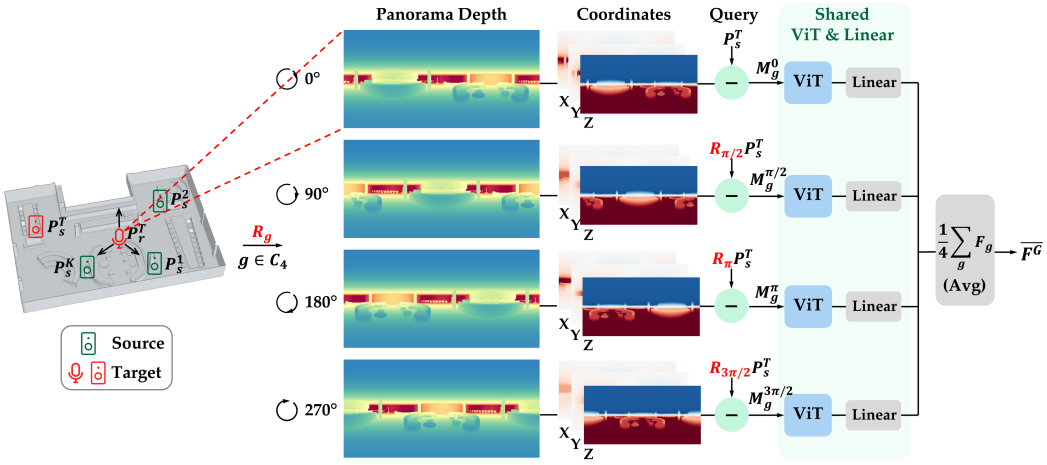

Figure 3: **Frame Averaging over the receiver-yaw orbit.** For visualization, we show the four yaw frames in C_4 (0°, 90°, 180°, and 270°); the same construction applies to a general finite yaw group C_N. In each frame, the panorama, geometry coordinate maps, and target-source query position are transformed consistently, encoded by a shared ViT and linear projection, and averaged to produce yaw-invariant geometry conditioning.

where R_g is a yaw rotation of the spatial and geometric conditioning. ORBITRIR enforces this symmetry through FA (Section 4).

Inverse Problem: RIR-Based Source Localization. At test time, we are given an unseen room with sparse context $\mathcal{O}$, a known receiver position x_r, and an observed RIR h_{obs} produced by an unknown source position $x_s^\star$. The source belongs to a finite candidate set $\mathcal{C} = \{x^{(1)}, \ldots, x^{(M)}\} \subset \mathbb{R}^3$, with $x_s^\star \in \mathcal{C}$. During localization, ORBITRIR is frozen at learned parameters $\theta^\star$. For each candidate $x^{(m)}$, we define $q_m = (x^{(m)}, x_r)$ and draw one candidate-conditioned RIR $\widehat{h}_m \sim p_{\theta^\star}(\cdot \mid q_m, \mathcal{O})$. A candidate-scoring function compares $\widehat{h}_m$ with the observed response. The localization decision is

$$\widehat{x}_s = \arg\max_{x^{(m)} \in \mathcal{C}} S\big(h_{\text{obs}}, \widehat{h}_m\big), \tag{2}$$

where S is the localization score specified in Section 4. Thus, the inverse problem is solved by repeatedly querying the frozen ORBITRIR across candidate positions rather than by training a localization-specific network.

4 METHOD

At illustrated in Fig. 2, at training time, we train ORBITRIR with flow matching under only RIR synthesis supervision. At inference time, we freeze ORBITRIR and invert it through candidate-conditioned analysis by synthesis.

4.1 FRAME-AVERAGED FEW-SHOT RIR SYNTHESIS

ORBITRIR is a geometry-constrained latent flow-matching model for few-shot RIR synthesis. As illustrated in Fig. 3, it applies FA over the receiver-yaw orbit to the spatial–geometric conditioning before conditioning the DiT.

Receiver-yaw orbit. Let $\mathcal{G}_N = \{g_n = 2\pi n/N \mid n = 0, \ldots, N-1\}$ denote the finite cyclic yaw group. Applying these yaw transformations consistently to the spatial and geometric inputs produces the receiver-yaw orbit used for Frame Averaging. Here, $R_z(g) \in SO(3)$ is the rotation matrix for a yaw rotation of angle g about the vertical axis. For each $g \in \mathcal{G}_N$, we transform a source position as $R_g x = x_r + R_z(g)(x - x_r)$, i.e., by rotating its receiver-relative coordinate about the fixed receiver. We apply the corresponding horizontal circular shift to the receiver-centered panoramic geometry

Table 1: RIR synthesis performance under sparse ($K_{\text{ctx}} = 1$) and dense ($K_{\text{ctx}} = 8$) acoustic context. Values are the arithmetic mean ± sample standard deviation over five (seeds 42–46), with context RIRs and generative noise resampled per seed. The best result in each column for each K_{ctx} is bolded.

Method	K	T60 (%) ↓	C50 (dB) ↓	EDT (ms) ↓	R@1 (%) ↑	R@5 (%) ↑	R@10 (%) ↑
FLAC (Brunetto, 2026)	1	10.287 ± 0.026	1.0884 ± 0.0088	43.437 ± 0.397	4.990 ± 0.115	15.111 ± 0.129	22.509 ± 0.181
ORBITRIR	1	9.543 ± 0.054	**1.0559 ± 0.0040**	**41.754 ± 0.347**	**5.166 ± 0.166**	**16.071 ± 0.241**	**23.721 ± 0.150**
Yaw-Augmented FLAC	1	**9.397 ± 0.052**	1.0855 ± 0.0051	42.447 ± 0.379	5.031 ± 0.063	15.919 ± 0.256	23.166 ± 0.135
FLAC (Brunetto, 2026)	8	8.993 ± 0.011	1.0093 ± 0.0035	40.650 ± 0.101	5.173 ± 0.138	15.430 ± 0.197	23.409 ± 0.056
ORBITRIR	8	8.202 ± 0.017	**0.9778 ± 0.0015**	**38.793 ± 0.074**	5.387 ± 0.075	16.456 ± 0.038	**24.198 ± 0.164**
Yaw-Augmented FLAC	8	**7.965 ± 0.014**	1.0132 ± 0.0016	39.923 ± 0.057	**5.391 ± 0.097**	**16.563 ± 0.169**	23.961 ± 0.107

G_{x_r}. The target and context source positions are transformed consistently, while the target RIR h and context RIRs $\mathcal{A}$ remain unchanged.

Frame-averaged conditioning. As illustrated in Fig. 3, each yaw frame produces a source-relative geometry map M_g by consistently transforming the panorama, geometry coordinate maps, and target-source position. The same ViT and linear projection are applied to every frame:

$$F_g = \text{Linear}(\text{ViT}(M_g)), \qquad \overline{F}^G = \frac{1}{|\mathcal{G}_N|} \sum_{g \in \mathcal{G}_N} F_g. \tag{3}$$

The averaged feature $\overline{F}^G$ provides the frame-invariant geometry conditioning; its integration with the spatial and acoustic context is shown in the upper part of Fig. 2. Because $\mathcal{G}_N$ is closed under yaw composition, any rotation $r \in \mathcal{G}_N$ only reindexes the frame-specific features:

$$\overline{F}^G_r = \frac{1}{N} \sum_{g \in \mathcal{G}_N} F_{gr} = \frac{1}{N} \sum_{g' \in \mathcal{G}_N} F_{g'} = \overline{F}^G, \tag{4}$$

where $g' = gr$. Thus, $\overline{F}^G$ is invariant to every yaw transformation in $\mathcal{G}_N$, even though the ViT itself need not be invariant.

Latent flow matching. We reuse FLAC's pretrained VAE (Brunetto, 2026) and keep its parameters frozen. Its encoder $\mathcal{E}$ maps the target RIR to the data latent $z_0 = \mathcal{E}(h)$. Given Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and a timestep $t \in [0, 1]$, the interpolated latent is $z_t = (1-t)z_0 + t\epsilon$, with target velocity $v_t = \epsilon - z_0$. We denote the complete DiT conditioning by $\bar{\tau}$, where the bar indicates that its spatial–geometric component is frame-averaged and acoustic context remains unchanged. The training objective for conditional velocity field is $\mathcal{L}_{\text{FM}} = \mathbb{E}\,\|u_\theta(z_t, t; \overline{\tau}) - v_t\|_2^2$. At inference, we apply classifier-free guidance (CFG) (Ho & Salimans, 2022). Let $u^{\text{c}}_\theta = u_\theta(z_t, t; \overline{\tau})$ and $u^{\text{u}}_\theta = u_\theta(z_t, t; \varnothing)$. The guided velocity is $\widehat{u}_\theta = u^{\text{u}}_\theta + \omega(u^{\text{c}}_\theta - u^{\text{u}}_\theta)$, where ω controls the conditioning strength. Starting from Gaussian noise at $t = 1$, we integrate the guided velocity field backward to $t = 0$ and decode the terminal latent into an RIR sample.

4.2 ANALYSIS-BY-SYNTHESIS LOCALIZATION

After training, we freeze ORBITRIR at parameters $\theta^\star$ and invert the forward model through analysis-by-synthesis (Yuille & Kersten, 2006). At inference, we evaluate each candidate by synthesizing the RIRs predicted at that location and comparing them with the observed RIR; the best-matching candidate is returned as the predicted source location. For each candidate position $x^{(m)}$, we generate a single candidate-conditioned RIR $\widehat{h}_m$. We compare $\widehat{h}_m$ with the observed RIR using an acoustic similarity measure (e.g., compute cosine similarity in the embedding space of the frozen AGREE audio encoder E_a (Brunetto, 2026): $a_m = \cos\big(E_a(h_{\text{obs}}), E_a(\widehat{h}_m)\big)$). The predicted source position is $\widehat{x}_s = \arg\max_{x^{(m)} \in \mathcal{C}} a_m$. Thus, we select the candidate whose synthesized RIR most closely matches the observed response. No localization-specific parameters are learned or tuned.

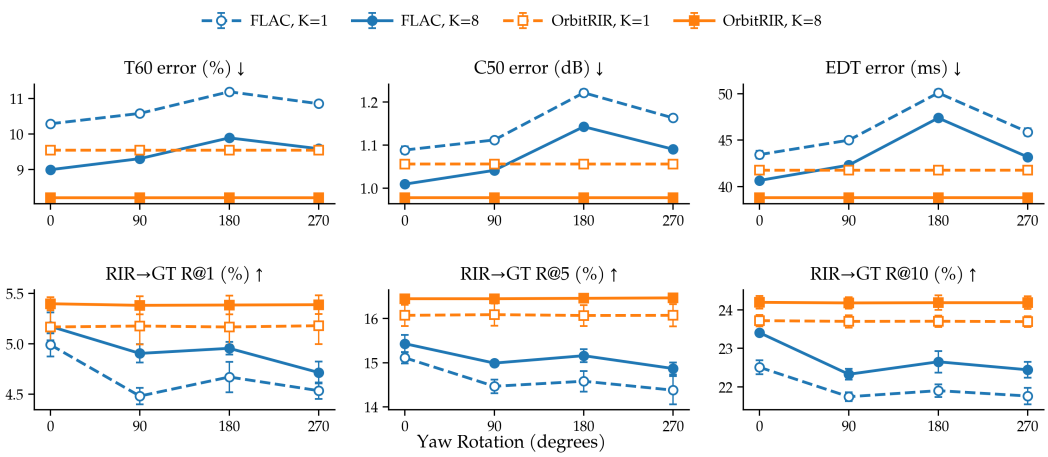

Figure 4: **Receiver-yaw invariance on unseen AcousticRooms.** We evaluate FLAC and ORBITRIR after re-expressing the same acoustic scenes under the four receiver-yaw frames in C_4, with sparse ($K_{\text{ctx}} = 1$) and dense ($K_{\text{ctx}} = 8$) context. ORBITRIR maintains nearly identical acoustic-parameter errors and RIR-to-ground-truth retrieval recall across all four frames, whereas FLAC degrades under rotated coordinate frames, particularly at $180°$. Error bars denote standard deviation. This experiment verifies invariance over the chosen C_4 orbit, rather than arbitrary $SO(2)$ rotations.

5 EXPERIMENTS

5.1 SETUP

Dataset and protocol. We evaluate on AcousticRooms (AR) dataset (Liu et al., 2025), following the 243/17 seen/unseen-room split used by xRIR and FLAC (Liu et al., 2025; Brunetto, 2026). For few-shot RIR synthesis, We evaluate sparse context ($K_{\text{ctx}} = 1$) and dense acoustic context ($K_{\text{ctx}} = 8$). For localization, we sample 64 queries from the 16 unseen rooms. Each query is conditioned on eight RIRs measured at the same receiver from other source positions ($K_{\text{ctx}} = 8$). For each candidate position, we synthesize an RIR and score its cosine similarity to the observed RIR in the forzen AGREE audio-embedding space; the highest-scoring candidate is selected. Candidate-grid construction is detailed in Appendix B.4.

Acoustics and Localization Metrics. For acoustic metrics, we evaluate RIR synthesis using T60, C50, and EDT errors and AGREE retrieval recall (R@1/5/10) following FLAC (Brunetto, 2026). For localization, we report median Euclidean localization error, SR@0.5m/SR@1m (Taira et al., 2021; Wang et al., 2023; He et al., 2025) and resolution-aware localization error (See Appendix B.5 for more details). For efficiency, we report both per-query and candidate-normalized localization latency; profiling details are provided in Appendix B.6.

Baselines. For few-shot RIR synthesis, we compare with FLAC (Brunetto, 2026) and Yaw-Augmented FLAC, which applies yaw augmentation without enforcing invariance by constrcution, isolating the benefit of FA beyond data augmentation. For localization, we compare against FEW-SHOTRIR (AR-adapted) (Majumder et al., 2022) (adapted for AR dataset), FEM-based localization (Dokmanić & Vetterli, 2012) and Random Selection. Implementation details are descibed in Appendix B.3. Additional results for FEW-SHOTRIR (AR-adapted) are provided in Appendix D.2.

5.2 FEW-SHOT RIR SYNTHESIS RESULTS

RIR synthesis quality. Table 1 compares RIR synthesis under sparse ($K_{\text{ctx}} = 1$) and dense ($K_{\text{ctx}} = 8$) acoustic context. Across both settings, ORBITRIR consistently outperforms FLAC across all metrics. Relative to FLAC, it reduces the T60 error by 7.2% and 8.8%, the C50 error by 3.0% and 3.1%, and the EDT error by 3.9% and 4.6% for

$K_{\text{ctx}} = 1$ and $K_{\text{ctx}} = 8$, respectively. It also improves R@1, R@5, and R@10 in both context settings. For seen rooms, detailed results are provided in Appendix C.1.

Furthermore, ORBITRIR outperforms Yaw-Augmented FLAC on eight of the twelve reported metrics. Together, these results indicate that enforcing the receiver-yaw invariance of omnidirectional monaural RIRs through FA improves RIR estimation at new positions in seen rooms and generalizes better to novel places. Although yaw augmentation improves performance, it leaves symmetry-violating solutions in the hypothesis class. FA excludes these solutions by construction, reducing the estimation burden under the same backbone. Two samples from AR dataset have been selected to demonstrate the synthesize waveforms and STFT results, as shown in Figure 7. In the cafe room, ORBITRIR precisely recovers both early reflections and decay (T60 error: 0.2,%). In the living room, FLAC decays too rapidly (T60 error: 19,%); yaw augmentation halves this error, while ORBITRIR outperforms both by halving it again. To assess sensitivity to checkpoint selection, all six synthesis metrics throughout training for both context sizes are reported in Appendix C.3.

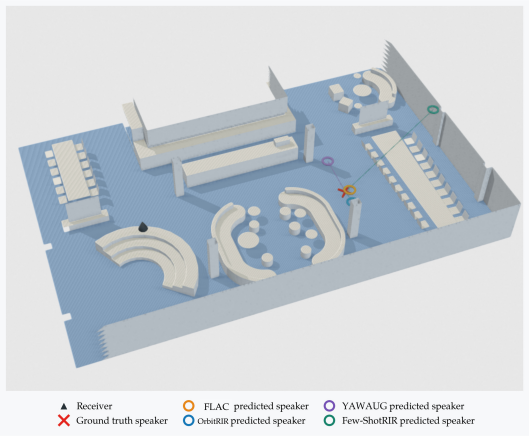

Figure 5: **Four-method comparison on a shared localization query.** Predicted source locations for the same query in `Cafe_idx_1`. All methods use identical acoustic observations and candidate locations.

Receiver-yaw invariance. Figure 4 evaluates the same acoustic scenes after re-expressing their spatial and geometric conditioning under the four receiver-yaw frames in C_4. The reported metrics for ORBITRIR remain unchanged across $0°$, $90°$, $180°$, and $270°$ for both context sizes. In contrast, FLAC exhibits higher acoustic-parameter errors and lower retrieval recall under the rotated coordinate frames, with particularly large degradation near $180°$. These results verify that ORBITRIR removes the corresponding coordinate-frame dependence and enforces invariance.

Training Dynamics of ORBITRIR and FLAC. Figure 10 shows training dynamics for sparse ($K_{\text{ctx}} = 1$) and dense ($K_{\text{ctx}} = 8$) acoustic context. Both FLAC and ORBITRIR improve rapidly early in training, with decreasing C_{50}/EDT errors and increasing retrieval rates, while T_{60} is more variable, especially for sparse context. Dense context generally yields better performance. ORBITRIR achieves stronger acoustic performance results nearly along all the way of training than FLAC, outperforming it after ∼20,000 steps and improving all six metrics at 40,000 steps. These gains persist across multiple late checkpoints, indicating a sustained benefit from frame-averaged conditioning rather than a single favorable checkpoint. More details are provided in Appendix C.3.

5.3 LOCALIZATION RESULTS

Figure 5 provides a qualitative example of the different inverse decisions produced by the four acoustic models under an identical localization query. The visualization complements the aggregate results in Table 2 by showing where the selected candidates lie within the room geometry.

Table 2: Comparison of acoustic models for sound-source localization. Median localization errors are reported in meters and success rates in percentages. Best results are shown in bold.

Model	Localization Error [m] ↓	SR@0.5 m ↑	SR@1.0 m ↑	Resolution-Aware SR @ 0.5 m ↑
ORBITRIR	**0.928**	**28.1%**	**53.1%**	**40.6%**
FEW-SHOTRIR (AR-adapted) (Majumder et al., 2022)	1.595	1.6%	26.6%	15.6%
FLAC (Brunetto, 2026)	1.009	15.6%	50.0%	23.4%
FEM-Based Localization (Dokmanić & Vetterli, 2012)	1.059	**28.1%**	50.0%	35.9%
Random Selection	2.320	7.8%	18.8%	14.1%

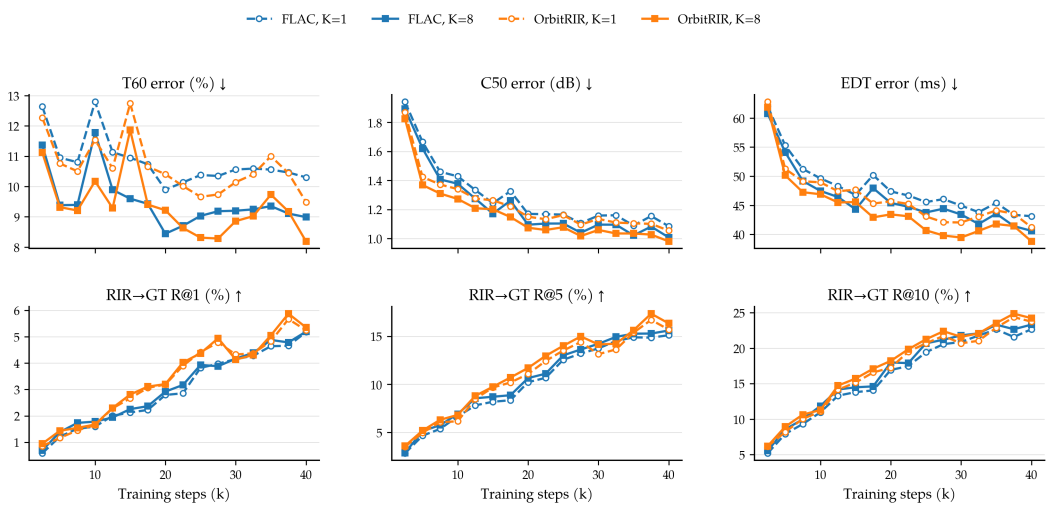

Figure 6: **RIR synthesis performance throughout training.** We compare OrbitRIR and FLAC on unseen AcousticRooms environments at checkpoints from 4k to 40k training steps, under sparse ($K_{\text{ctx}} = 1$) and dense ($K_{\text{ctx}} = 8$) context.

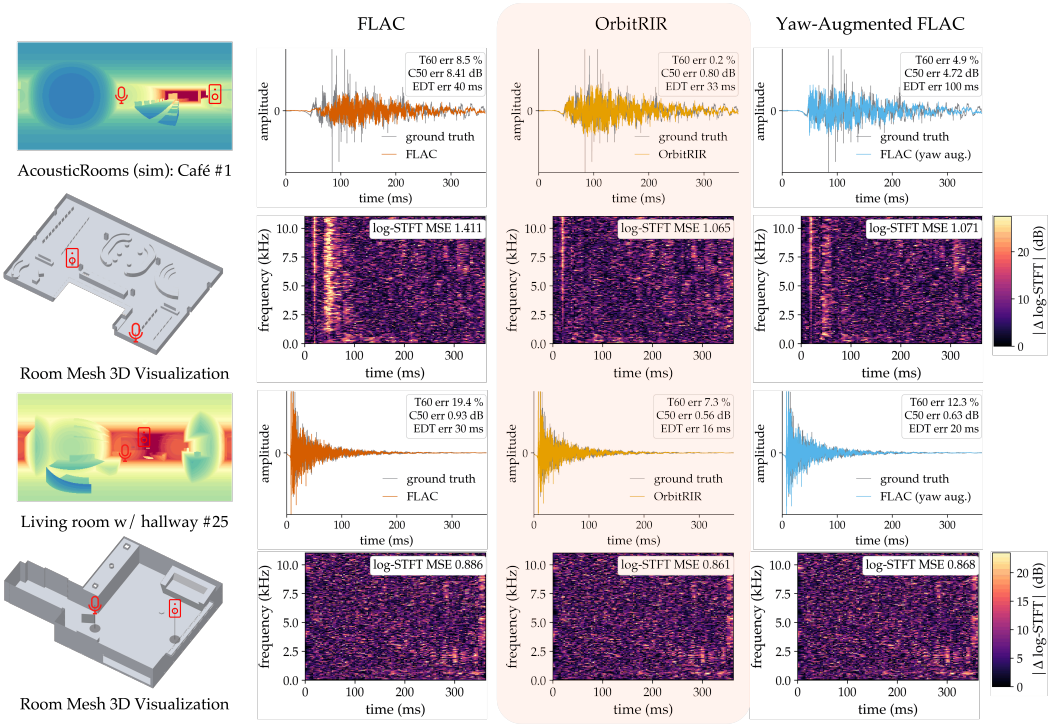

Figure 7: **Qualitative comparison on two unseen AcousticRooms scenes.** Top: *Cafe 1*, source S10 → receiver R94; bottom: *Living room with hallway 25*, S8 → R24; every model receives K=8 reference measurements. Left: the depth panorama rendered at the receiver and the room mesh, with the source (loudspeaker icon) and receiver (microphone icon). Columns: FLAC, OrbitRIR and Yaw-Augmented FLAC (40k checkpoint). Upper panel of each row: predicted RIR (colour) over the ground truth (grey). Lower panel: $|\Delta \log|\text{STFT}||$ between prediction and ground truth in dB.

Localization fidelity improvements. As shown in Table 2, OrbitRIR reduces the median localization error by 8.0%, increases SR@0.5m, SR@1m and resolution-aware SR@0.5m by 80.1%, 6.2% and 73.5% compared with FLAC. OrbitRIR also outperforms the deterministic Few-ShotRIR

Table 3: **Localization inference latency.** We report per-query wall-clock latency and candidate-normalized latency. For the latter, dataset-wide mean denotes total latency divided by the total number of candidate evaluations, whereas median and P90 are computed across query-normalized latencies. Best and second-best results are shown in bold and italics, respectively.

Method	Per query [s/query]			Per candidate [ms/candidate]		
	Mean ↓	Median ↓	P90 ↓	Dataset-wide mean ↓	Median ↓	P90 ↓
ORBITRIR	*10.919*	*3.719*	*52.829*	*15.091*	*16.539*	*20.351*
FEW-SHOTRIR (AR-adapted) (AR-adapted) (Majumder et al., 2022)	**1.263**	**0.400**	**6.457**	**1.745**	**1.769**	**1.907**
FEM-Based Localization (Dokmanić & Vetterli, 2012)	695.461	70.496	1547.088	961.245	408.057	655.186

(AR-adapted) baseline and Random Selection across all localization metrics. Overall, the localization results show that enforcing receiver-yaw symmetry in the forward RIR model improves its effectiveness when inverted for downstream spatial inference, without localization-specific training. Additional localization visualizations are provided in Appendix D.1.

Localization inference latency. Table 3 reports per-query and candidate-normalized inference latency. Compared with FEM-Based Localization, ORBITRIR reduces per-query latency by 94.7–98.4% and candidate-normalized latency by 95.9–98.4%, reflecting the computational advantage of learned acoustic synthesis over numerical FEM simulation. FEW-SHOTRIR (AR-adapted) is approximately $8.6\times$ faster than ORBITRIR, but achieves substantially lower localization accuracy (Table 2). Overall, ORBITRIR provides a favorable accuracy–efficiency trade-off, combining strong localization performance with substantially lower inference cost than the physics-based approach.

6 CONCLUSION AND LIMITATIONS

We introduced ORBITRIR, a geometry-constrained generative model that applies Frame Averaging over the receiver-yaw orbit for few-shot RIR synthesis. We further showed that the frozen model can be inverted through analysis-by-synthesis for 3D source localization without localization-specific training or fine-tuning. On AcousticRooms, ORBITRIR outperform the state-of-the-art FLAC (Brunetto, 2026) model and Yaw-Augmented FLAC on both few-shot acoustic synthesis task and sound source localization tasks. Advances in acoustic synthesis and localization may support embodied applications such as robot navigatiom using sound. However, ORBITRIR's construction averages over the finite group C_N, rather than continuous $SO(2)$, and therefore does not guarantee invariance to arbitrary yaw rotations. Moreover, localization searches over a finite candidate mesh, so its achievable precision is limited by the mesh resolution, while using a finer mesh increases inference cost.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs, design or provide feedback on research methodology or experiments, clean and reformat dataset and implement methods. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses and interpret results. And generate synthetic data sets, assist with translation and support qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, creation of artifacts, draft parts of a research paper, summarize or analyse existing literature, brainstorming, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature and format references. We have reviewed all AI-assisted work. The authors reviewed the AI-assisted material, checked numerical statements against experimental results, verified references against the cited publications, and inspected generated figures and scripts for consistency with the data and evaluation protocols. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

We describe the experimental settings and implementation details to support reproducibility, and provide the full implementation of our method in an anonymized repository linked in Appendix A. The coupled forward-inverse problem formulation and spatial symmetry requirements are stated in Section 3. Section 4 describes the frame-averaged conditioning, latent flow matching, and analysis-by-synthesis inference, while Appendix B.1 details the distributed training configuration and memory management. Architecture and optimization hyperparameters for the generator and geometry conditioning are summarized in Table 4. Section 5.1 outlines the AcousticRooms dataset splits, evaluation metrics, and baseline comparisons. Appendices B.2 and B.3 document the exact implementation of the yaw-augmented generator, the AR-adapted Few-ShotRIR, and FEM-based localization baselines. Appendix B.4 specifies the 3D candidate-grid construction and mesh-validity mask, while Appendices B.5 and B.6 detail the localization metrics and inference latency profiling protocol.

References

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), 2019.

Matan Atzmon, Koki Nagano, Sanja Fidler, Sameh Khamis, and Yaron Lipman. Frame averaging for equivariant shape space learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Amandine Brunetto. Few-shot acoustic synthesis with multimodal flow matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Amandine Brunetto, Sascha Hornauer, and Fabien Moutarde. NeRAF: 3D scene infused neural radiance and acoustic fields. In *International Conference on Learning Representations (ICLR)*, 2025.

Ziyang Chen, Xixi Hu, and Andrew Owens. Structure from silence: Learning scene structure from ambient sound. In *Conference on Robot Learning (CoRL)*, 2022.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning (ICML)*, 2016.

Jeffrey Dean and Luiz André Barroso. The tail at scale. *Communications of the ACM*, 56(2), 2013.

David Diaz-Guerra, Antonio Miguel, and Jose R. Beltran. Direction of arrival estimation of sound sources using icosahedral CNNs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31, 2023.

Joseph H. DiBiase, Harvey F. Silverman, and Michael S. Brandstein. Robust localization in reverberant rooms. In *Microphone Arrays: Signal Processing Techniques and Applications*. Springer, 2001.

Ivan Dokmanić and Martin Vetterli. Room helps: Acoustic localization with finite elements. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

Alexandre Agm Duval, Victor Schmidt, Alex Hernández-García, Santiago Miret, Fragkiskos D. Malliaros, Yoshua Bengio, and David Rolnick. FAENet: Frame averaging equivariant GNN for materials modeling. In *International Conference on Machine Learning (ICML)*, 2023.

Linfeng Feng, Yijun Gong, and Xiao-Lei Zhang. Soft label coding for end-to-end sound source localization with ad-hoc microphone arrays. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International Conference on Machine Learning (ICML)*, 2021.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

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), 2022.

Yuhang He, Sangyun Shin, Anoop Cherian, Niki Trigoni, and Andrew Markham. SoundLoc3D: Invisible 3D sound source localization and classification using a multimodal RGB-D acoustic camera. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint*, 2022.

Diwei Huang, Kunyang Lin, Peihao Chen, and Qing Du. Map-guided few-shot audio-visual acoustics modeling. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

Charles H. Knapp and G. Clifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4), 1976.

Zitong Lan, Chenhao Zheng, Zhiwei Zheng, and Mingmin Zhao. Acoustic volume rendering for neural impulse response fields. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Zitong Lan, Yiduo Hao, and Mingmin Zhao. Resounding acoustic fields with reciprocity. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. AV-NeRF: Learning neural fields for real-world audio-visual scene synthesis. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Xiulong Liu, Anurag Kumar, Paul Calamia, Sebastia V. Amengual, Calvin Murdock, Ishwarya Ananthabhotla, Philip Robinson, Eli Shlizerman, et al. Hearing anywhere in any environment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Andrew Luo, Yilun Du, Michael J. Tarr, Joshua B. Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Sagnik Majumder, Changan Chen, Ziad Al-Halah, and Kristen Grauman. Few-shot audio-visual learning of environment acoustics. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Michael Neri, Archontis Politis, Daniel Krause, Marco Carli, and Tuomas Virtanen. Speaker distance estimation in enclosures from single-channel audio. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32, 2024.

Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J. Smith, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations (ICLR)*, 2022.

Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, et al. MLPerf inference benchmark. In *ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, 2020.

Ryotaro Sato, Kenta Niwa, and Kazunori Kobayashi. Ambisonic signal processing DNNs guaranteeing rotation, scale and time translation equivariance. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 2021.

Ralph O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3), 1986.

Kun Su, Mingfei Chen, and Eli Shlizerman. INRAS: Implicit neural representation for audio scenes. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Mohammad J. Taghizadeh, Afsaneh Asaei, Saeid Haghighatshoar, Philip N. Garner, and Hervé Bourlard. Spatial sound localization via multipath euclidean distance matrix recovery. *IEEE Journal of Selected Topics in Signal Processing*, 9(5), 2015.

Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4), 2021.

Haoran Wang, Shengzhi Qiang, Qi Wang, Qing Lu, and Yuanqing Wang. Physics-regularized neural acoustic fields for spatial layout inference from sparse room impulse responses. *The Journal of the Acoustical Society of America*, 160(1), 2026.

Jiang Wang, Runwu Shi, Benjamin Yen, He Kong, and Kazuhiro Nakadai. Single-microphone-based sound source localization for mobile robots in reverberant environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

Mason Long Wang, Samuel Clarke, Jui-Hsien Wang, Ruohan Gao, and Jiajun Wu. SoundCam: A dataset for finding humans using room acoustics. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Mason Long Wang, Ryosuke Sawata, Samuel Clarke, Ruohan Gao, Shangzhe Wu, and Jiajun Wu. Hearing anything anywhere. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Mengyu Yang, Patrick Grady, Samarth Brahmbhatt, Arun Balajee Vasudevan, Charles C. Kemp, and James Hays. The un-kidnappable robot: Acoustic localization of sneaking people. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Alan Yuille and Daniel Kersten. Vision as Bayesian inference: Analysis by synthesis? *Trends in Cognitive Sciences*, 10(7), 2006.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv preprint*, 2018.

A Code Availability

To facilitate the review process, we provide the full implementation of our method in an anonymized repository: `https://anonymous.4open.science/r/ORBITRIR`. Instructions for setting up and reproducing the proposed method are provided in the repository `README`.

B Experimental Details

B.1 OrbitRIR Training Hyperparameters & Configuration

We train all models for 40k optimization steps using distributed data parallelism on two NVIDIA A6000 GPUs, with a per-GPU batch size of 32 and a global batch size of 64. Checkpoints are evaluated at 2.5k-step intervals, while all reported comparisons use the final checkpoint at 40k steps (Tables 1, 5 and 6).

Controlled training configuration. To isolate the effect of frame-averaged conditioning, FLAC and OrbitRIR are trained from the same initialization (seed 42) using the same data order, optimizer, learning-rate schedule, batch size, and number of training steps. The two runs differ only in their conditioning paths: OrbitRIR applies a shared ViT geometry encoder to the four consistently rotated scene representations in the C_4 yaw orbit and averages the resulting features, whereas FLAC conditions on a single receiver-yaw representation without orbit averaging. The models are optimized independently after initialization.

Distributed training configuration. Training uses two-GPU distributed data parallelism with a per-GPU batch size of 32. Synchronized Batch Normalization aggregates statistics across both workers, yielding an effective BatchNorm batch size of 64 and matching the reference single-GPU configuration. To reduce activation memory, we apply activation checkpointing to the ViT backbone of OrbitRIR using PyTorch's non-reentrant implementation (`use_reentrant=False`). This is required for OrbitRIR to fit the per-GPU batch size of 32 and is retained for FLAC to keep the memory-management configuration matched. Unless otherwise stated in the conditioning-mismatch ablation (Appendix C.2), each checkpoint is evaluated using its corresponding training conditioner: `fa_invariant` with the C_4 orbit for OrbitRIR and `vanilla` for FLAC.

Training resource usage. Under the two-A6000 DDP configuration described above, the cumulative active wall-clock time required to reach 40k optimization steps was 40.3 hours for FLAC and 124.9 hours for OrbitRIR (80.6 and 249.8 GPU-hours, respectively). These measurements include data loading, forward and backward computation, optimizer updates, DDP communication, and checkpoint writing, but exclude queueing, downtime, and separate checkpoint evaluation. Representative clean training windows showed device-memory footprints of approximately 5.7 GiB per GPU for FLAC and 15.3 GiB per GPU for OrbitRIR.

B.2 Few-Shot Acoustic Synthesis Baseline Implementation

Yaw-Augmented FLAC. This baseline retains FLAC's architecture and vanilla single-frame conditioning, but encourages yaw robustness through training-time data augmentation rather than frame averaging. At each optimizer step, every training example independently draws an integer offset $d_i \sim \mathrm{Unif}\{0, \ldots, 511\}$, corresponding to $\alpha_i = 2\pi d_i/512$ (0.703125° per column). We use the same physically consistent yaw transformation defined in Sec. 4.1. However, for each training example, we randomly sample one of the 512 panorama-aligned rotations, rather than evaluating and averaging the four C_4 rotations used by frame averaging. The equirectangular depth point cloud is circularly rolled by d_i columns, and its per-pixel Cartesian vectors together with all source and context pose fields are rotated by the corresponding yaw angle α_i about the vertical axis. The monaural target RIR and the context RIRs are left unchanged, since this common yaw transformation is target-preserving under the symmetry in Eq. 1. Yaw offsets are generated deterministically from a base seed of 42 for reproducibility. No random yaw augmentation or frame averaging is used at inference.

Shared evaluation protocol. All models are evaluated using the EMA parameters from 40k-step checkpoints. FLAC and Yaw-Augmented FLAC use the standard, unaveraged FLAC conditioner,

Table 4: Training configuration for FLAC and ORBITRIR. Unless otherwise noted, both runs use identical hyperparameters; the final block lists settings specific to ORBITRIR.

Category	Parameter	Value
Generator (DiT)	Depth / width / heads	12 / 256 / 8
	Latent input/output channels	32
	Cross-attention / global-conditioning dimensions	256 / 512 (AdaLN)
	Room-context dropout probability	0.1
VAE pretransform	Latent channels / waveform length	32 / 10,240 samples at 22.05 kHz
	State	Frozen
Geometry conditioning	Backbone	DINOv3 ViT-S/16, fine-tuned
Training data	Examples / rooms / context size K	291,210 / 243 / 8
Optimization	Optimizer	AdamW, learning rate 5×10^{-5}, $\beta = (0.9, 0.999)$, weight decay 10^{-3}
	Learning-rate schedule	InverseLR ($\gamma_{\text{inv}} = 10^6$, power 0.5, warmup 0.99)
	Global batch / BatchNorm statistics batch	64 / 64 (SyncBN; 32 per GPU)
	LR-schedule horizon	40,000
	Precision / random seed	BF16 mixed precision / 42
	Exponential moving average (EMA)	Decay 0.9999, power $3/4$
	Hardware	2× NVIDIA A6000 48 GB, DDP
ORBITRIR settings	Conditioning method	`fa_invariant` (FLAC uses `vanilla`)
	Frame-averaging orbit	$C_4 = \{0°, 90°, 180°, 270°\}$
	ViT gradient checkpointing	Enabled only for ORBITRIR to reduce activation memory

whereas ORBITRIR uses C_4 frame-averaged conditioning. Table 7 separately evaluates the effect of exchanging these conditioning pathways at test time.

RIRs are generated with a single Euler step of the rectified-flow sampler. Following FLAC (Brunetto, 2026), we set the classifier-free guidance scale to $\omega = 1$, which directly uses the conditional velocity field without guidance extrapolation. All model and encoder parameters remain frozen during evaluation. The unseen-room results in Tables 1 and 5 use the complete AcousticRooms held-out split, comprising 6,337 queries from 17 rooms. Table 6 reports results on the corresponding seen-room evaluation split. We evaluate both sparse ($K_{\text{ctx}} = 1$) and dense ($K_{\text{ctx}} = 8$) acoustic context; each context RIR is measured at the query receiver from a source position different from the target.

For each evaluation seed, metrics are first averaged within each room and then across rooms, assigning equal weight to every room. We report the mean and standard deviation over five seeds, $\{42, \ldots, 46\}$. These seeds control only the Gaussian noise used for RIR generation; model parameters, queries, and acoustic contexts are fixed across seeds. Retrieval metrics (R@1/5/10) are computed using a frozen AGREE audio encoder trained on AcousticRooms and used only for evaluation. In the receiver-yaw experiment, each scene is re-expressed in the four C_4 frames using consistent panorama shifts and coordinate rotations. The queries, contexts, and generation seeds are shared across frames, so the evaluation conditions differ only in the receiver-yaw representation presented to the model.

B.3 LOCALIZATION BASELINE IMPLEMENTATION DETAILS

FEW-SHOTRIR (AR-adapted) (Majumder et al., 2022) implementation details. We implement an independent, depth-only and monaural AcousticRooms adaptation of the released FEW-SHOTRIR (AR-adapted) architecture. Because AcousticRooms contains no colocated source–receiver measurements, we approximate each context echo by selecting, for every source, the available RIR whose receiver is closest to that source. We train the randomly initialized model with $K_{\text{ctx}} = 8$ using RIR-reconstruction supervision only. The model conditions on context-RIR magnitude spectrograms,

depth panoramas at the corresponding context receivers, and six-dimensional context and query poses containing receiver and source coordinates relative to the first context receiver. It predicts a monaural log-magnitude STFT and is optimized using linear-magnitude ℓ_1 loss plus a 0.01-weighted frequency-collapsed spectral energy-decay loss. For the reported $K_{\text{ctx}} = 8$ localization setting, For each candidate, we use 32 iterations of deterministic Griffin–Lim phase reconstruction to convert the predicted magnitude spectrogram into a 10,240-sample waveform. And then we select the candidate with the highest cosine similarity to the observed RIR in the frozen AGREE audio-embedding space.

FEM-based method implementation details. We approximate the room as a simple box using the depth panorama observed at the receiver. We expand the box by 0.05 m and convert it into a tetrahedral mesh with a maximum edge length of 0.22 m. A uniform Sabine boundary condition is estimated from the median T_{60} of eight context RIRs, and the FEM solver predicts the acoustic response from every candidate location at frequencies between 80 and 300 Hz. To estimate the source location, we treat each candidate response as a dictionary entry and apply one-step Orthogonal Matching Pursuit (OMP) following Room Helps (Dokmanić & Vetterli, 2012). OMP selects the candidate whose simulated response best explains the observed response across all frequencies while allowing for an unknown complex amplitude. Queries that fail the geometry-coverage check use a deterministic random-candidate fallback with seed 42.

B.4 Candidate-Grid Construction

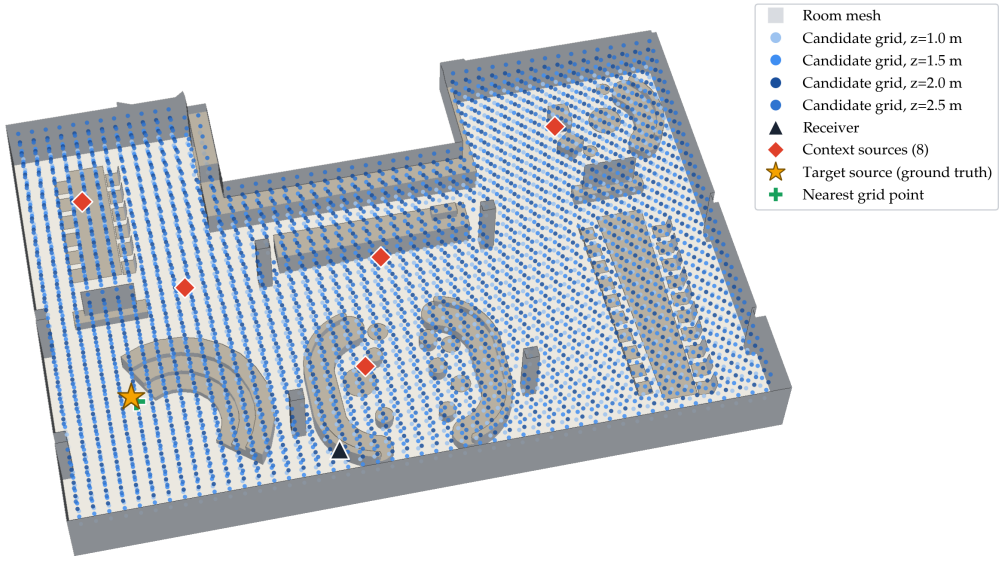

Figure 8: **3D candidate grid for a representative localization query in `Cafe_idx_1`.** Blue points show the valid candidates from the 0.5 m lattice at heights $z \in \{1.0, 1.5, 2.0, 2.5\}$ m, overlaid on the room mesh. The black triangle denotes the receiver, red diamonds denote the eight context sources, and the gold star denotes the continuous ground-truth source. The green marker identifies the nearest grid candidate, illustrating the discretization error imposed by the finite candidate grid. Height-wise views of the same grid are provided in Fig. 9.

For each room, we construct candidate locations in the global AcousticRooms (Liu et al., 2025) coordinate frame using its official triangular mesh $\mathcal{M}$. We load and process the triangular room meshes using Open3D Zhou et al. (2018). For room r, let $\mathcal{V}_r$ denote the vertex set of its triangular mesh. For each coordinate axis $j \in \{x, y, z\}$, we define the lower and upper mesh bounds as

$$a_{r,j}^{\min} = \min_{\mathbf{v} \in \mathcal{V}_r} v_j, \qquad a_{r,j}^{\max} = \max_{\mathbf{v} \in \mathcal{V}_r} v_j, \tag{5}$$

where v_j denotes the coordinate of vertex $\mathbf{v}$ along axis j. The six scalar values $\{a_{r,j}^{\min}, a_{r,j}^{\max}\}_{j \in \{x,y,z\}}$ define the axis-aligned bounding box of the room mesh.

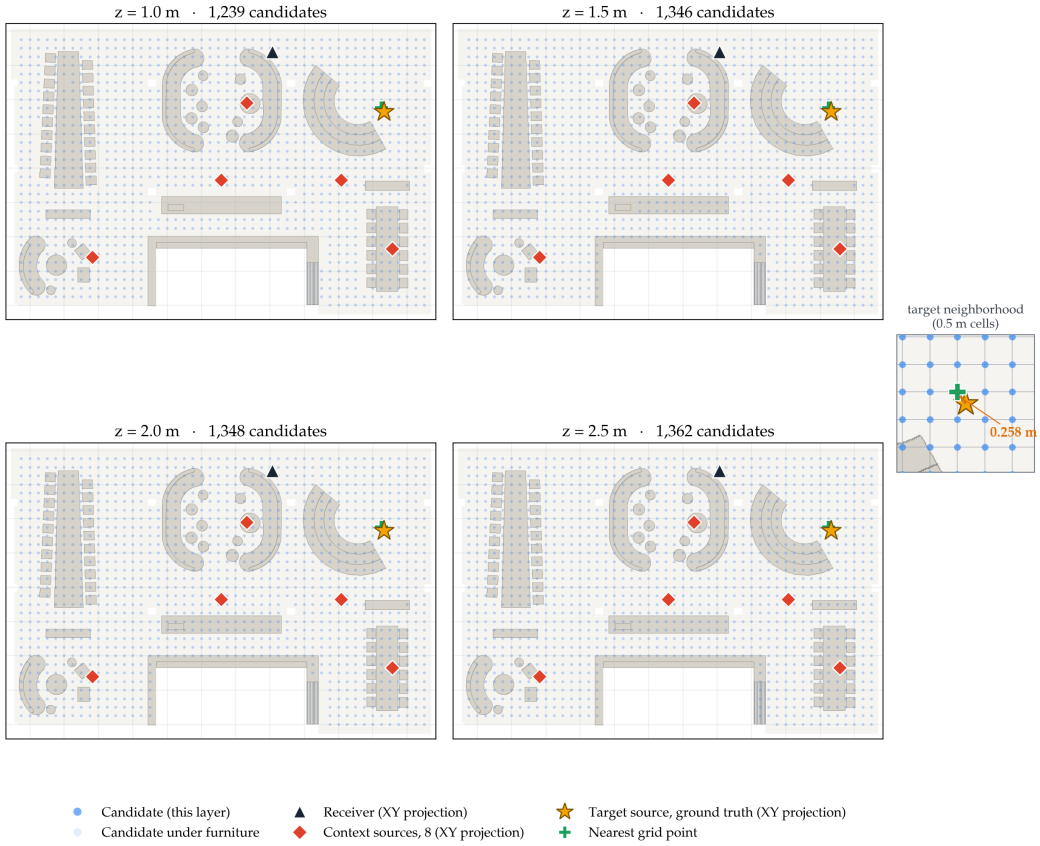

Figure 9: **Height-wise slices of the 3D candidate grid in `Cafe_idx_1`.** The four panels show the valid candidate locations at $z \in \{1.0, 1.5, 2.0, 2.5\}$ m, with the number of candidates reported above each slice. Blue dots denote candidate positions, while lighter dots indicate candidates whose top-down projections lie beneath furniture. The receiver, eight context sources, and continuous ground-truth source are projected onto each slice. The inset enlarges the target neighborhood, showing the nearest grid candidate and the resulting 0.258 m discretization error under the 0.5 m grid spacing.

The lattice is anchored at the global coordinate origin. Let $\Delta = 0.5\,\mathrm{m}$ denote the lattice spacing and let $k \in \mathbb{Z}$ denote an integer lattice index. For axis j, we define the set of admissible lattice coordinates as

$$\mathcal{A}_{r,j} = \left\{ k\Delta \,\middle|\, k \in \mathbb{Z},\ a_{r,j}^{\min} \le k\Delta \le a_{r,j}^{\max} \right\}. \tag{6}$$

Thus, $\mathcal{A}_{r,j}$ contains all multiples of $0.5\,\mathrm{m}$ that lie within the mesh bounds along axis j.

Finally, the raw three-dimensional candidate grid for room r is formed by taking every combination of admissible x, y, and z coordinates:

$$\mathcal{G}_r^{\mathrm{raw}} = \mathcal{A}_{r,x} \times \mathcal{A}_{r,y} \times \mathcal{A}_{r,z}. \tag{7}$$

Because every coordinate has the form $k\Delta$, the lattices of all rooms are aligned to the same global origin rather than being offset from their room-specific bounding-box minima. (See Figure 8 and Figure 9).

We next apply a mesh-validity mask. Because the architectural meshes are non-watertight, we determine free space using Open3D (Zhou et al., 2018) ray casting along 31 fixed, non-axis-aligned Fibonacci-sphere directions. For each direction, an odd number of ray–mesh intersections constitutes a free-space vote, and a point is retained only if a strict majority (> 15 of 31 directions) votes for free space. We additionally require its unsigned distance from the nearest mesh surface to satisfy

$$d(\mathbf{x}, \mathcal{M}) + \epsilon \ge 0.20\,\mathrm{m}, \qquad \epsilon = 10^{-4}\,\mathrm{m}. \tag{8}$$

The ray-parity test determines physical validity, whereas the $0.20\,\mathrm{m}$ clearance is a source-distribution prior.

For query q, we further remove points within $0.50\,\mathrm{m}$ of the known receiver $\mathbf{r}_q$ or within $0.25\,\mathrm{m}$ of any selected context-source position. Candidate heights are restricted using only the observed context coordinates:

$$z \in \left[\min_n z_{q,n}^{\mathrm{ctx}} - 0.50,\ \max_n z_{q,n}^{\mathrm{ctx}} + 0.50\right]\mathrm{m}. \tag{9}$$

The same numerical tolerance is applied at all boundaries. Before localization scoring, we adopted this height-restricted branch globally after verifying that all 5,337 audited query grids remained nonempty and that it introduced no additional grid-oracle errors above $0.5\,\mathrm{m}$ relative to the full-height grid. The ground-truth source position is never inserted into the candidate grid.

B.5 Localization Metrics

For each localization query q, we define the Euclidean localization error as $e_{\mathrm{loc}}^{(q)} = \|\widehat{\mathbf{x}}^{(q)} - \mathbf{x}_{\mathrm{gt}}^{(q)}\|_2$. Following prior acoustic and indoor localization protocols, we report the median localization error and thresholded success rates (Taira et al., 2021; Wang et al., 2023; He et al., 2025). A query is considered successful at radius r if $e_{\mathrm{loc}}^{(q)} \le r$, and SR@r denotes the proportion of successful queries. We use $r \in \{0.5, 1.0\}$ m and report the resulting metrics as SR@0.5m and SR@1m. To account for candidate-grid resolution, we additionally define the resolution-aware error $e_{\mathrm{RA}}^{(q)} = e_{\mathrm{loc}}^{(q)} - \delta_{\mathrm{grid}}^{(q)}$. This metric measures the error beyond the resolution limit of each query-specific candidate set (Feng et al., 2023).

B.6 Latency Profiling Setup

This section describes the profiling protocol underlying Table 3. The localization benchmark is drawn from the mesh-available portion of the AcousticRooms unseen split, comprising 16 of the 17 unseen rooms; `ListeningRoom_idx_2` is excluded upstream because its official geometry is unavailable.

Evaluation set and instrumentation. The profiling source pool contains eight frozen queries from each of the 16 mesh-available unseen rooms, yielding 128 queries in total. Auditorium and Cafe are excluded before coverage filtering because they are extreme computational outliers (including them would require multi-hour FEM executions for individual queries–several orders of magnitude above the typical per-query cost of OrbitRIR–and would cause the aggregate latency statistics to be dominated by these two atypically large environments). Of the 16 source queries from these two rooms, 15 satisfy the bounding criterion (see Appendix B.4) and collectively contain 66,852 candidate evaluations, with an average of 4,457 candidates per query. In comparison, the other 14-room profiling subset contains 17,245 candidate evaluations over 97 queries, or 178 candidates per query on average. Including these two rooms would increase the candidate workload nearly fivefold, rendering a complete matched profiling run computationally impractical.

OrbitRIR. For each query, timing begins before loading the observed RIR, eight context RIRs, and the receiver depth panorama. Checkpoint loading, candidate filtering, localization-metric computation, and result serialization are excluded from timing. The AGREE-based candidate-selection latency is measured separately and added to the model latency. The reported per-query latency is therefore

$$t_{\text{OrbitRIR}}^{(q)} = t_{\mathrm{ctx},\,K_{\mathrm{ctx}}=8}^{(q)} + t_{\mathrm{cond+gen},\,K_{\mathrm{gen}}=1}^{(q)} + t_{\mathrm{AGREE}}^{(q)}. \tag{10}$$

Here, q indexes an evaluation query, $t_{\mathrm{ctx}}^{(q)}$ denotes the query-level conditioning time for the eight context RIRs, $t_{\mathrm{cond+gen}}^{(q)}$ denotes the time for candidate-dependent conditioning and one RIR generation per candidate, and $t_{\mathrm{AGREE}}^{(q)}$ denotes the time required to encode and compare the observed and generated RIRs and select the highest-scoring candidate.

Few-ShotRIR (AR-adapted) (Majumder et al., 2022). We profile Few-ShotRIR (AR-adapted) on 128 queries from 16 rooms using $K_{\mathrm{ctx}} = 8$ context RIRs and generating one deterministic RIR per candidate location ($K_{\mathrm{gen}} = 1$). Ordinary input loading, checkpoint loading, and

candidate filtering are excluded from timing. For each query, the reported latency is

$$t^{(q)}_{\text{FewShotRIR}} = t^{(q)}_{\text{cond+gen},\,(K_{\text{ctx}}=8,\,K_{\text{gen}}=1)} + t^{(q)}_{\text{AGREE}}. \tag{11}$$

Here, q indexes an evaluation query, $t^{(q)}_{\text{cond+gen}}$ denotes the time required to condition on the eight context RIRs and generate one RIR for every candidate, and $t^{(q)}_{\text{AGREE}}$ denotes the time required to encode and compare the observed and generated RIRs and select the highest-scoring candidate.

FEM-based localization (Dokmanić & Vetterli, 2012). The reported FEM based method latency is

$$t^{(q)}_{\text{loc,FEM-OMP}} = \begin{cases} t^{(q)}_{\text{mesh}} + t^{(q)}_{\text{op}} + t^{(q)}_{\text{solve}} + t^{(q)}_{\text{OMP}}, & q \in \mathcal{Q}_{\text{pass}}, \\ t^{(q)}_{\text{cand}} + t^{(q)}_{\text{gate}} + t^{(q)}_{\text{rand}}, & q \in \mathcal{Q}_{\text{fail}}. \end{cases} \tag{12}$$

Here, q indexes an evaluation query, while $\mathcal{Q}_{\text{pass}}$ and $\mathcal{Q}_{\text{fail}}$ denote the queries that pass and fail the geometry-coverage check, respectively. For a successful query, $t^{(q)}_{\text{mesh}}$ measures construction of the depth-derived tetrahedral mesh with a maximum edge length of 0.22 m, and $t^{(q)}_{\text{op}}$ covers FEM matrix assembly and construction of the receiver and candidate interpolation operators. The solve term $t^{(q)}_{\text{solve}}$ includes estimating a uniform Sabine boundary condition from eight context RIRs and computing the complex candidate responses at 102 frequency bins in the 80–300 Hz band. Finally, $t^{(q)}_{\text{OMP}}$ measures one-step Orthogonal Matching Pursuit (OMP), which selects the candidate response that best explains the observed response across these frequencies while allowing for an unknown complex gain. For failed queries, FEM and OMP are skipped, and the measured latency instead includes candidate preparation, coverage detection, and deterministic random-candidate selection with seed 42.

Reported statistics. Following established inference-benchmarking practice (Reddi et al., 2020; Dean & Barroso, 2013), we summarize the per-query latency distribution using its mean, median, and empirical 90th percentile:

$$\begin{aligned} \bar{t}_{\text{loc}} &= \frac{1}{Q}\sum_{q=1}^{Q} t^{(q)}_{\text{loc}}, \\ t_{\text{loc},50} &= \text{median}_q\; t^{(q)}_{\text{loc}}, \\ t_{\text{loc},90} &= \text{Quantile}_{0.9}\left(\{t^{(q)}_{\text{loc}}\}_{q=1}^{Q}\right). \end{aligned} \tag{13}$$

These quantities correspond to the mean, median, and P90 columns in the per-query portion of Table 3 and are reported in seconds per query.

Because candidate-set sizes vary across queries, we additionally define the query-normalized per-candidate latency

$$\ell^{(q)} = 10^3 \frac{t^{(q)}_{\text{loc}}}{|\mathcal{C}^{(q)}|} \quad [\text{ms/candidate}]. \tag{14}$$

The candidate-normalized table reports the median and P90 of $\ell^{(q)}$ across queries. Its dataset-wide mean is

$$\bar{\ell}_{\text{data}} = 10^3 \frac{\sum_{q=1}^{Q} t^{(q)}_{\text{loc}}}{\sum_{q=1}^{Q} |\mathcal{C}^{(q)}|} \quad [\text{ms/candidate}]. \tag{15}$$

The query median and P90 assign equal weight to each query, whereas $\bar{\ell}_{\text{data}}$ weights queries in proportion to their candidate counts.

C Additional RIR Synthesis Results

C.1 Detailed Experimental Results on AcousticRooms

Unseen-room results. Table 5 reports the complete RIR-synthesis results on unseen AcousticRooms environments for both context sizes (K=1 and K=8) and all four receiver-yaw frames in the chosen C_4 orbit.

Table 5: Full RIR-synthesis results on unseen AcousticRooms environments under the four receiver-yaw frames in C_4. All models are evaluated at 40,000 training steps under sparse ($K_{\mathrm{ctx}} = 1$) and dense ($K_{\mathrm{ctx}} = 8$) acoustic context. Results are reported as mean $\pm$ standard deviation under five evaluations (seeds 42–46) of a fixed checkpoint. The upper and lower panels report acoustic-parameter and retrieval metrics, respectively.

Acoustic-parameter metrics

Method	K	Yaw	T60 (%) ↓	C50 (dB) ↓	EDT (ms) ↓
FLAC	1	0°	10.287 ± 0.026	1.0884 ± 0.0088	43.437 ± 0.397
ORBITRIR	1	0°	9.543 ± 0.054	1.0559 ± 0.0040	41.754 ± 0.347
Yaw-Augmented FLAC	1	0°	9.397 ± 0.052	1.0855 ± 0.0051	42.447 ± 0.379
FLAC	1	90°	10.579 ± 0.041	1.1117 ± 0.0054	44.995 ± 0.266
ORBITRIR	1	90°	9.541 ± 0.053	1.0560 ± 0.0041	41.754 ± 0.348
Yaw-Augmented FLAC	1	90°	9.391 ± 0.063	1.0946 ± 0.0066	42.803 ± 0.352
FLAC	1	180°	11.190 ± 0.063	1.2212 ± 0.0074	50.094 ± 0.362
ORBITRIR	1	180°	9.542 ± 0.054	1.0559 ± 0.0041	41.755 ± 0.347
Yaw-Augmented FLAC	1	180°	9.407 ± 0.057	1.0980 ± 0.0063	42.706 ± 0.364
FLAC	1	270°	10.853 ± 0.039	1.1628 ± 0.0066	45.876 ± 0.393
ORBITRIR	1	270°	9.542 ± 0.054	1.0559 ± 0.0041	41.756 ± 0.348
Yaw-Augmented FLAC	1	270°	9.444 ± 0.065	1.0957 ± 0.0054	42.765 ± 0.357
FLAC	8	0°	8.993 ± 0.011	1.0093 ± 0.0035	40.650 ± 0.101
ORBITRIR	8	0°	8.202 ± 0.017	0.9778 ± 0.0016	38.793 ± 0.074
Yaw-Augmented FLAC	8	0°	7.965 ± 0.014	1.0132 ± 0.0016	39.923 ± 0.057
FLAC	8	90°	9.304 ± 0.013	1.0413 ± 0.0005	42.304 ± 0.086
ORBITRIR	8	90°	8.202 ± 0.017	0.9779 ± 0.0015	38.796 ± 0.074
Yaw-Augmented FLAC	8	90°	7.955 ± 0.014	1.0224 ± 0.0021	40.276 ± 0.081
FLAC	8	180°	9.889 ± 0.017	1.1425 ± 0.0034	47.388 ± 0.051
ORBITRIR	8	180°	8.202 ± 0.017	0.9778 ± 0.0015	38.795 ± 0.075
Yaw-Augmented FLAC	8	180°	7.975 ± 0.015	1.0272 ± 0.0012	40.181 ± 0.069
FLAC	8	270°	9.585 ± 0.020	1.0902 ± 0.0015	43.182 ± 0.060
ORBITRIR	8	270°	8.202 ± 0.017	0.9778 ± 0.0015	38.793 ± 0.074
Yaw-Augmented FLAC	8	270°	8.026 ± 0.016	1.0229 ± 0.0017	40.198 ± 0.088

Retrieval metrics

Method	K	Yaw	R@1 (%) ↑	R@5 (%) ↑	R@10 (%) ↑
FLAC	1	0°	4.990 ± 0.115	15.111 ± 0.129	22.509 ± 0.181
ORBITRIR	1	0°	5.166 ± 0.166	16.071 ± 0.241	23.721 ± 0.150
Yaw-Augmented FLAC	1	0°	5.031 ± 0.063	15.919 ± 0.256	23.166 ± 0.135
FLAC	1	90°	4.482 ± 0.081	14.467 ± 0.154	21.742 ± 0.113
ORBITRIR	1	90°	5.176 ± 0.181	16.086 ± 0.253	23.702 ± 0.160
Yaw-Augmented FLAC	1	90°	4.993 ± 0.033	15.632 ± 0.158	22.964 ± 0.079
FLAC	1	180°	4.671 ± 0.151	14.581 ± 0.234	21.900 ± 0.164
ORBITRIR	1	180°	5.166 ± 0.176	16.068 ± 0.237	23.705 ± 0.149
Yaw-Augmented FLAC	1	180°	4.889 ± 0.169	15.749 ± 0.207	23.121 ± 0.189
FLAC	1	270°	4.535 ± 0.081	14.379 ± 0.319	21.761 ± 0.215
ORBITRIR	1	270°	5.179 ± 0.182	16.071 ± 0.249	23.696 ± 0.136
Yaw-Augmented FLAC	1	270°	5.015 ± 0.147	15.777 ± 0.166	23.118 ± 0.286
FLAC	8	0°	5.173 ± 0.138	15.430 ± 0.197	23.409 ± 0.056
ORBITRIR	8	0°	5.397 ± 0.064	16.446 ± 0.057	24.198 ± 0.164
Yaw-Augmented FLAC	8	0°	5.391 ± 0.097	16.563 ± 0.169	23.961 ± 0.107
FLAC	8	90°	4.905 ± 0.091	14.991 ± 0.058	22.326 ± 0.138
ORBITRIR	8	90°	5.381 ± 0.089	16.446 ± 0.063	24.182 ± 0.151
Yaw-Augmented FLAC	8	90°	5.223 ± 0.151	16.282 ± 0.137	23.730 ± 0.117
FLAC	8	180°	4.955 ± 0.063	15.159 ± 0.147	22.651 ± 0.281
ORBITRIR	8	180°	5.384 ± 0.092	16.456 ± 0.044	24.188 ± 0.189
Yaw-Augmented FLAC	8	180°	5.119 ± 0.111	16.355 ± 0.155	23.677 ± 0.200
FLAC	8	270°	4.715 ± 0.107	14.868 ± 0.139	22.443 ± 0.203
ORBITRIR	8	270°	5.387 ± 0.092	16.465 ± 0.055	24.188 ± 0.164
Yaw-Augmented FLAC	8	270°	5.356 ± 0.073	16.437 ± 0.113	23.872 ± 0.204

Seen-room results. Table 6 reports results on the seen-room split under sparse and dense acoustic context.

Across the two context settings, ORBITRIR achieves the best mean result on ten of the fourteen metric–context combinations, improving both perceptual accuracy and retrieval performance over FLAC in seen rooms.

Table 6: Full RIR-synthesis results on seen AcousticRooms environments under sparse ($K_{\mathrm{ctx}} = 1$) and dense ($K_{\mathrm{ctx}} = 8$) acoustic context. Results are reported as mean $\pm$ standard deviation. The best result within each context setting is shown in bold.

Model	K	T60 (%) ↓	C50 (dB) ↓	EDT (ms) ↓	R@1 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	FD ↓
FLAC	1	6.9327 ± 0.0725	0.7256 ± 0.0048	30.7589 ± 0.1845	5.9418 ± 0.1228	17.0275 ± 0.2387	24.0952 ± 0.2596	0.3164 ± 0.0005
ORBITRIR	1	**6.5479 ± 0.0609**	**0.6861 ± 0.0055**	**29.4880 ± 0.1926**	**6.4340 ± 0.1452**	18.1084 ± 0.1574	**25.2501 ± 0.1673**	0.3190 ± 0.0001
Yaw-Augmented FLAC	1	6.605 ± 0.051	0.7158 ± 0.0035	30.176 ± 0.148	6.366 ± 0.135	**18.163 ± 0.177**	24.829 ± 0.172	**0.3111 ± 0.0002**
FLAC	8	5.7340 ± 0.0084	0.6604 ± 0.0016	27.7053 ± 0.0376	6.0930 ± 0.1179	17.4586 ± 0.1358	24.6646 ± 0.1358	0.3195 ± 0.0003
ORBITRIR	8	5.4848 ± 0.0194	**0.6272 ± 0.0011**	**26.6944 ± 0.0669**	**6.6656 ± 0.0629**	**18.6778 ± 0.1168**	**26.0125 ± 0.1638**	0.3217 ± 0.0001
Yaw-Augmented FLAC	8	**5.353 ± 0.015**	0.6555 ± 0.0012	27.207 ± 0.042	6.637 ± 0.091	18.610 ± 0.067	25.572 ± 0.133	**0.3131 ± 0.0003**

Table 7: Compatibility between training weights and evaluation-time conditioning on unseen AcousticRooms environments at the canonical receiver-yaw frame. All checkpoints are evaluated at 40,000 training steps. Results are reported as mean $\pm$ standard deviation. The upper and lower panels report acoustic-parameter and retrieval metrics, respectively; the best result within each context setting is shown in bold.

Acoustic-parameter metrics

Training weights	Evaluation conditioning	K	T60 (%) ↓	C50 (dB) ↓	EDT (ms) ↓
FLAC	Vanilla	1	10.287 ± 0.026	1.0884 ± 0.0088	43.437 ± 0.397
FLAC	FA	1	10.257 ± 0.058	1.0824 ± 0.0060	45.436 ± 0.319
ORBITRIR	FA	1	**9.543 ± 0.054**	**1.0559 ± 0.0040**	**41.754 ± 0.347**
ORBITRIR	Vanilla	1	11.885 ± 0.070	2.1399 ± 0.0045	82.995 ± 0.516
FLAC	Vanilla	8	8.993 ± 0.011	1.0093 ± 0.0035	40.650 ± 0.101
FLAC	FA	8	8.817 ± 0.012	1.0009 ± 0.0019	42.283 ± 0.084
ORBITRIR	FA	8	**8.202 ± 0.017**	**0.9778 ± 0.0015**	**38.793 ± 0.074**
ORBITRIR	Vanilla	8	10.652 ± 0.022	2.0817 ± 0.0041	80.860 ± 0.538

Retrieval metrics

Training weights	Evaluation conditioning	K	R@1 (%) ↑	R@5 (%) ↑	R@10 (%) ↑
FLAC	Vanilla	1	4.990 ± 0.115	15.111 ± 0.129	22.509 ± 0.181
FLAC	FA	1	3.926 ± 0.117	13.192 ± 0.193	20.218 ± 0.224
ORBITRIR	FA	1	**5.166 ± 0.166**	**16.071 ± 0.241**	**23.721 ± 0.150**
ORBITRIR	Vanilla	1	0.622 ± 0.047	2.970 ± 0.097	4.933 ± 0.173
FLAC	Vanilla	8	5.173 ± 0.138	15.430 ± 0.197	23.409 ± 0.056
FLAC	FA	8	4.049 ± 0.147	13.726 ± 0.054	20.909 ± 0.239
ORBITRIR	FA	8	**5.387 ± 0.075**	**16.456 ± 0.038**	**24.198 ± 0.164**
ORBITRIR	Vanilla	8	0.675 ± 0.069	3.093 ± 0.074	5.075 ± 0.139

C.2 EFFECTS OF TRAINING- AND EVALUATION-TIME CONDITIONING

We further evaluate whether the conditioning construction used at test time must match that used during training. Here, *Vanilla* denotes the original FLAC conditioning path, whereas *FA* denotes the frame-averaged spatial–geometric conditioning used by ORBITRIR. Table 7 evaluates all four combinations of training weights and evaluation conditioning.

Applying FA only at evaluation time does not reproduce the gains of ORBITRIR: although it slightly improves the acoustic-parameter errors of the vanilla checkpoint in some settings, it substantially reduces retrieval performance. *Conversely, evaluating the* ORBITRIR *checkpoint without its frame-averaged conditioning causes severe degradation across all metrics.* These results indicate that the invariant conditioning must be learned jointly with the generative model rather than introduced as a post-hoc inference-time modification.

C.3 TRAINING DYNAMICS OF FEW-SHOT RIR SYNTHESIS

Figure 10 reports perceptual and scene-consistency metrics throughout training for both sparse ($K_{\mathrm{ctx}} = 1$) and dense ($K_{\mathrm{ctx}} = 8$) acoustic context. Checkpoints are evaluated at 2,500-step intervals up to 40,000 steps. For both FLAC and ORBITRIR, the C_{50} and EDT errors decrease rapidly during early training and then improve more gradually, while R@1, R@5, and R@10 increase steadily. The

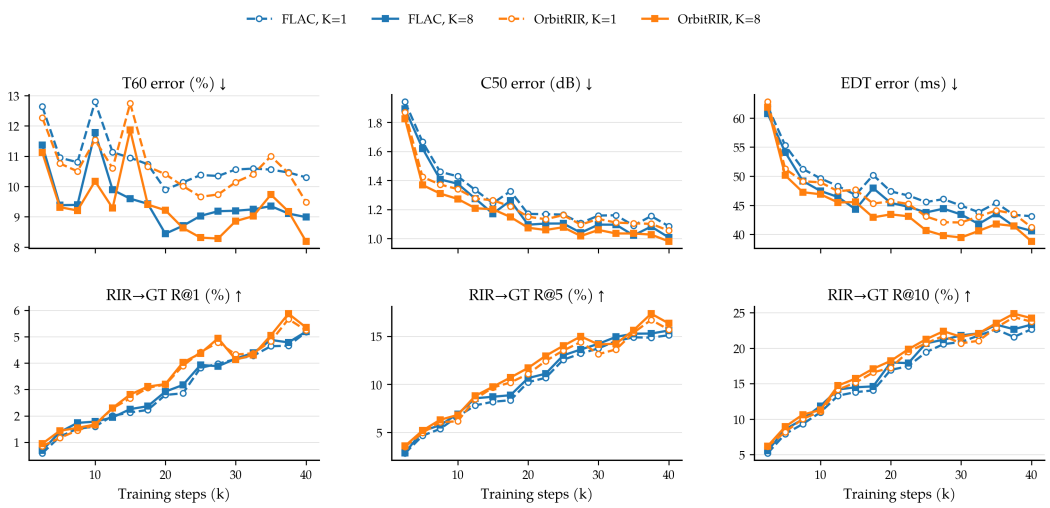

Figure 10: **RIR synthesis performance throughout training.** We compare ORBITRIR and FLAC on unseen AcousticRooms environments at checkpoints from 4k to 40k training steps, under sparse ($K_{\text{ctx}} = 1$) and dense ($K_{\text{ctx}} = 8$) context.

T_{60} error exhibits greater checkpoint-level variation, particularly under sparse context. Dense context generally produces lower acoustic-parameter errors and higher retrieval rates than sparse context.

ORBITRIR follows a similar convergence trajectory to FLAC but achieves consistently stronger late-training performance. From approximately 20,000 steps onward, it generally yields lower C_{50} and EDT errors and higher retrieval rates under both context settings. Although the T_{60} trajectories are less monotonic, ORBITRIR obtains lower final errors for both $K_{\text{ctx}} = 1$ and $K_{\text{ctx}} = 8$. At 40,000 steps, it improves all six reported metrics over FLAC, consistent with Table 1. Importantly, these improvements persist across multiple late-training checkpoints rather than arising from a single favorable checkpoint, suggesting that frame-averaged conditioning provides a sustained advantage throughout optimization.

D ADDITIONAL LOCALIZATION RESULTS

D.1 QUALITATIVE 3D LOCALIZATION COMPARISONS

Figures 11–16 provide qualitative comparisons of predicted source locations in unseen AcousticRooms environments. All methods are evaluated using the same observed RIR, receiver location, and query-specific candidate grid. In each rendering, the black triangle denotes the receiver, the red cross denotes the ground-truth source, and colored circles denote method predictions. A line connecting a prediction to the ground truth visualizes the corresponding localization error.

In the shared `Cafe_idx_1` example (Fig. 11), ORBITRIR and FLAC select candidates near the ground-truth source, whereas Yaw-Augmented FLAC and especially FEW-SHOTRIR (AR-adapted) produce more distant predictions. Across the six-room comparison, ORBITRIR remains close to the target in all displayed layouts. By contrast, FLAC exhibits larger errors in the bathroom and bedroom examples, Yaw-Augmented FLAC in the bathroom, FEW-SHOTRIR (AR-adapted) in the apartment and restaurant, and FEM-Based Localization in the apartment, bedroom, and meeting-room examples. These cases illustrate the behavior captured by Table 2: ORBITRIR more consistently selects a nearby candidate rather than a geometrically distant region, contributing to its stronger SR@0.5 m and resolution-aware success rate.

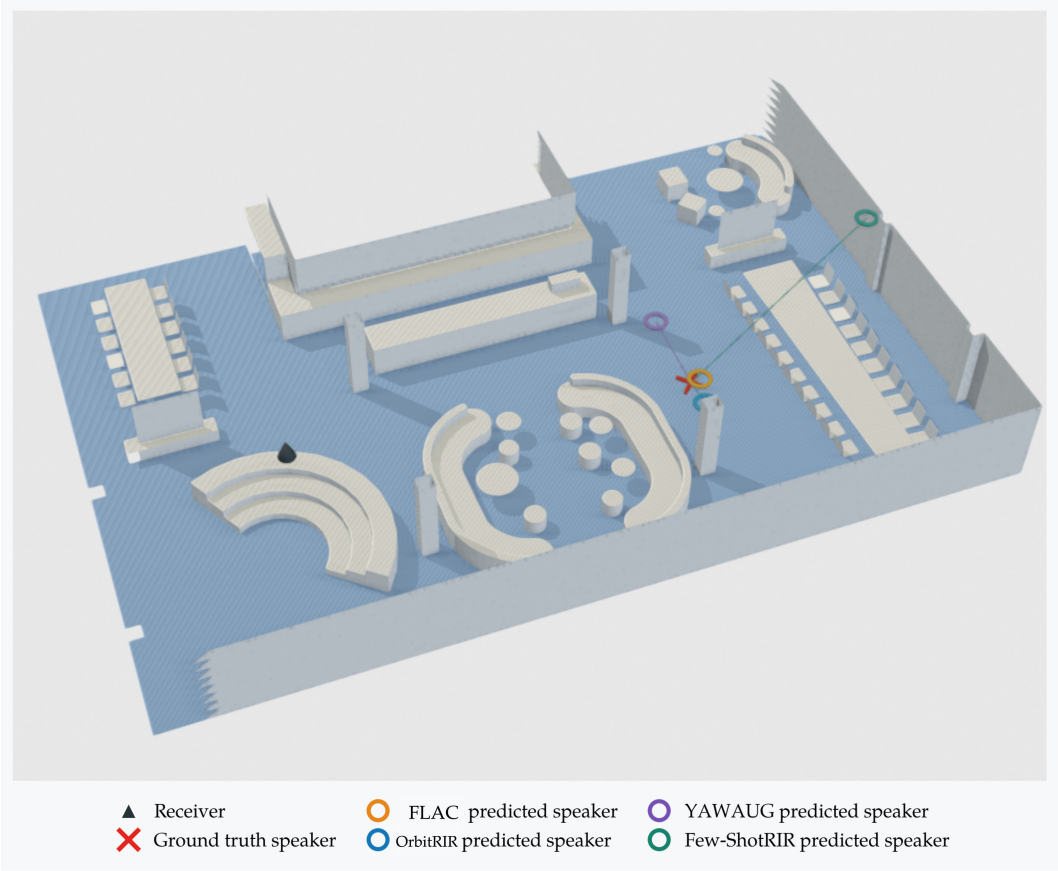

Figure 11: **Four-method comparison on a shared localization query.** Predictions from FLAC, ORBITRIR, Yaw-Augmented FLAC, and FEW-SHOTRIR (AR-adapted) are shown for the same query in `Cafe_idx_1`. The four methods receive identical acoustic observations and candidate locations, illustrating how their inferred source positions differ under the same room geometry.

D.2 FEW-SHOTRIR (AR-ADAPTED) LOCALIZATION: DIRECT REGRESSION VERSUS ANALYSIS-BY-SYNTHESIS

We compare two ways of using RIRs for localization: a localization-specific direct regressor adapted from FEW-SHOTRIR (Majumder et al., 2022), and our analysis-by-synthesis use of the AR-adapted FEW-SHOTRIR forward model (FEW-SHOTRIR (AR-adapted)).

Direct coordinate regression. Following the downstream localization architecture of FEW-SHOTRIR, a ResNet-18 encodes the log-magnitude spectrogram of an RIR, and a linear prediction head directly regresses the source coordinates relative to the receiver. We adapt the original two-dimensional output to three-dimensional AcousticRooms coordinates and train the network with localization-specific coordinate supervision. At inference, the observed RIR is mapped to a continuous source-location estimate in a single forward pass, without constructing or searching a candidate set.

The original FEW-SHOTRIR experiment used this localization network to measure whether query-conditioned predicted RIRs retained source-position information. Because our setting instead begins with an observed RIR from an unknown source, we treat this baseline as a FEW-SHOTRIR-*style direct-regression adaptation*, rather than an exact reproduction of the original downstream protocol.

Analysis-by-synthesis. Our analysis-by-synthesis variant uses the AR-adapted FEW-SHOTRIR model only as a forward RIR predictor (the exactly the same method named FEW-SHOTRIR (AR-adapted)). Its parameters remain frozen during localization, and no localization-specific network is

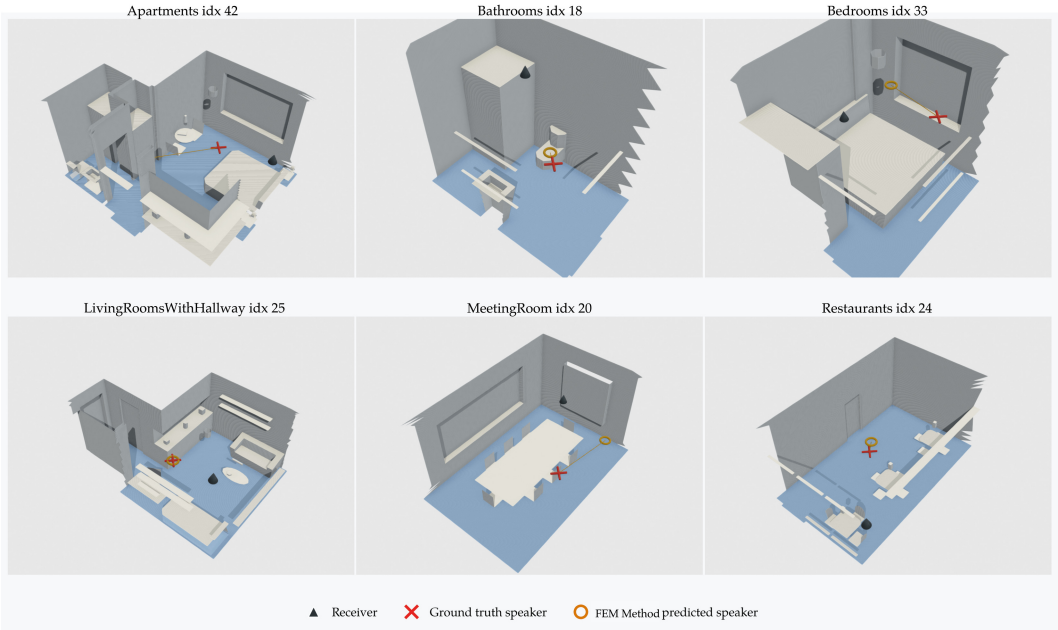

Figure 12: **Qualitative results for FEM-Based Localization.** Predicted source locations are shown for one query from each of six unseen rooms with different layouts. Orange circles indicate the FEM-based predictions; black triangles and red crosses indicate the receivers and ground-truth sources, respectively.

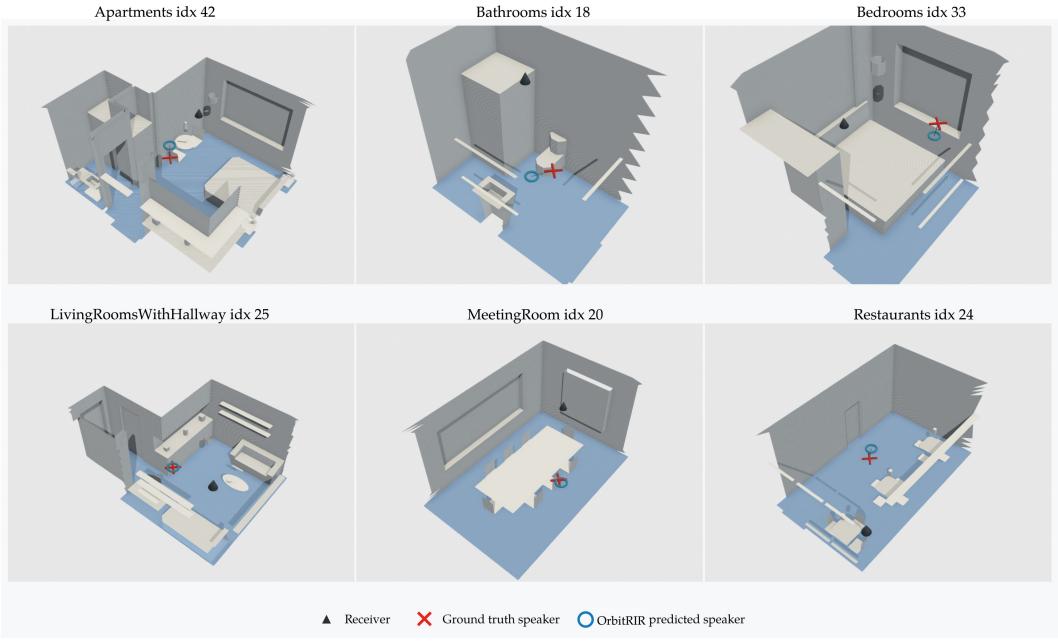

Figure 13: **Qualitative results for ORBITRIR.** Blue circles show the source locations predicted by ORBITRIR across the shared set of six unseen-room queries. The predictions remain close to the ground-truth sources across diverse room layouts, illustrating the spatial inference enabled by the frame-averaged acoustic generator.

trained. For each mesh-valid candidate source position, the model predicts a candidate-conditioned RIR. We compare this prediction with the observed RIR in the frozen AGREE audio-embedding space and select the candidate with the highest similarity. Unlike direct regression, this procedure

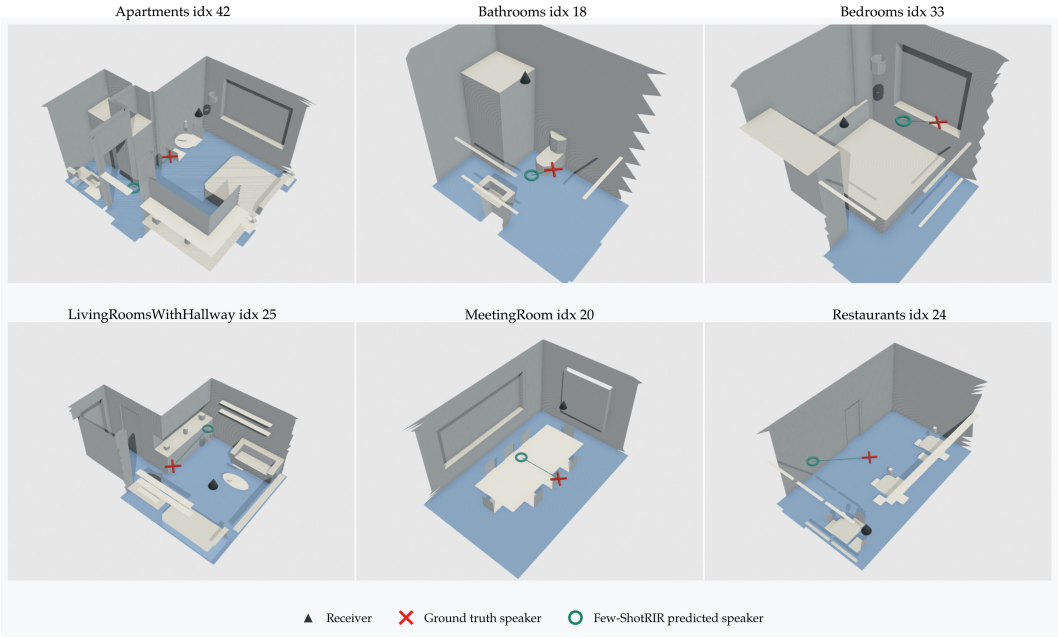

Figure 14: **Qualitative results for FEW-SHOTRIR (AR-adapted) .** Predicted source locations are shown for the same six unseen-room queries used throughout this comparison. Green circles indicate the FEW-SHOTRIR (AR-adapted) predictions, with line segments visualizing their distances from the ground-truth sources.

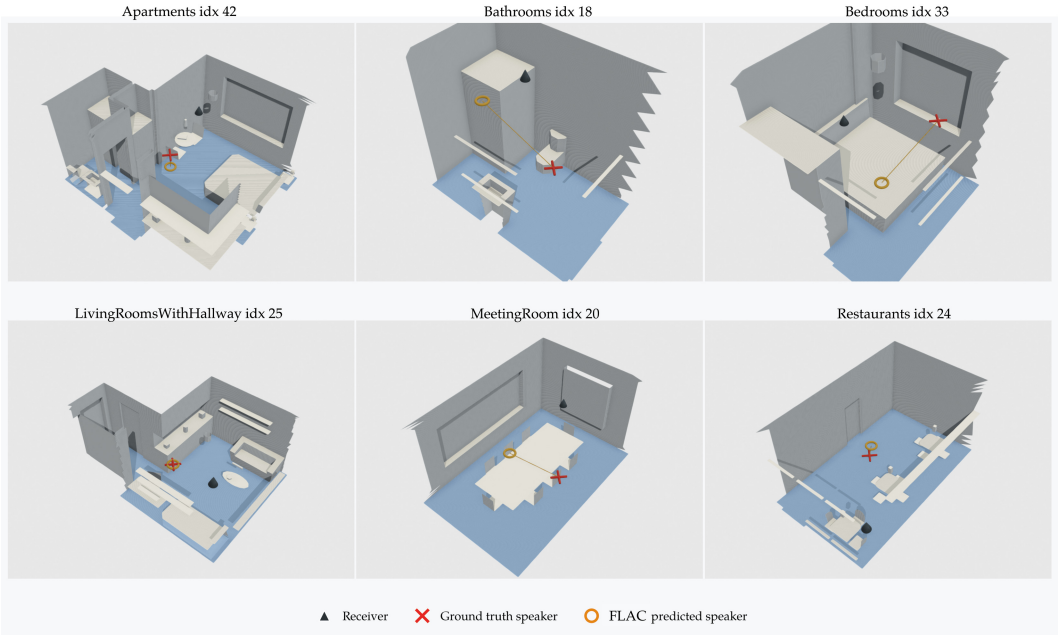

Figure 15: **Qualitative results for FLAC.** Orange circles show the source locations predicted by FLAC across the shared set of six unseen-room queries. The examples illustrate how localization errors vary across room geometries despite using the same candidate-grid and scoring protocol.

conditions every localization hypothesis on the geometry and sparse acoustic context of the test room, although its output resolution is limited by the candidate grid.

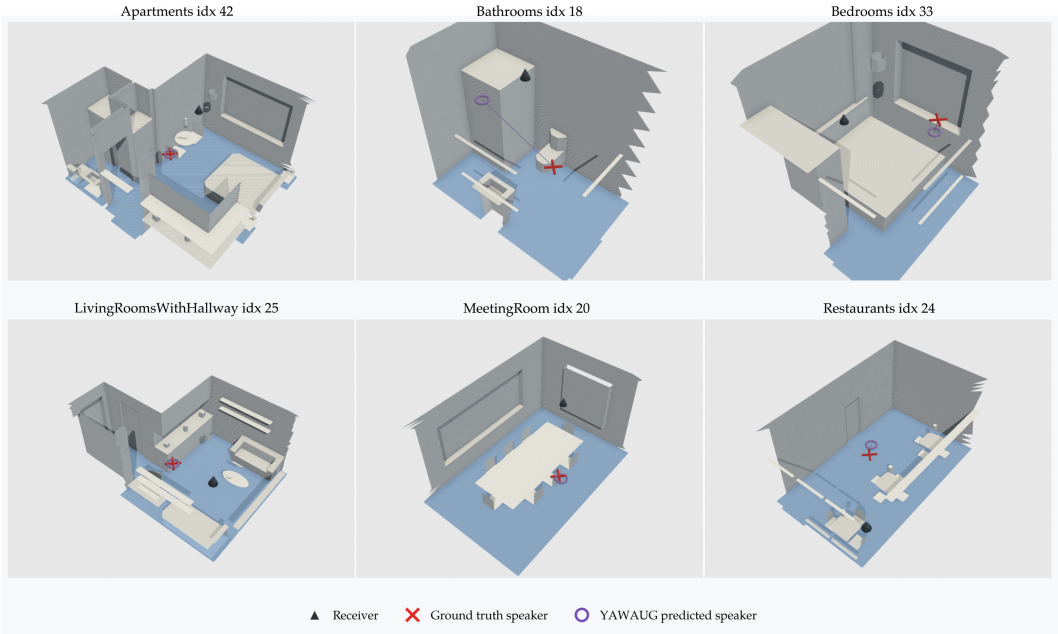

Figure 16: **Qualitative results for Yaw-Augmented FLAC.** Purple circles show the source locations predicted by the yaw-augmentation baseline across the same six unseen-room queries. Comparison with Fig. 13 illustrates the difference between learning yaw robustness through augmentation and enforcing the corresponding symmetry through Frame Averaging.

Table 8: Comparison of direct regression and analysis-by-synthesis using FEW-SHOTRIR on the same 64-query unseen-room evaluation. Errors are reported in meters and success rates in percentages. Best results are shown in bold.

Localization strategy	Median error ↓	SR@0.5 m ↑	SR@1 m ↑	Resolution-aware SR@0.5 m ↑
Direct regression (ResNet-18)	1.934	**2.344%**	17.188%	2.344%
Analysis-by-synthesis (FEW-SHOTRIR (AR-adapted))	**1.595**	1.600%	**26.600%**	**15.600%**

Comparison. As shown in Table 8, analysis-by-synthesis reduces the median localization error by 17.5% and improves SR@1 m and resolution-aware SR@0.5 m by 9.412 and 13.256 percentage points, respectively. Direct regression is slightly better under the raw SR@0.5 m metric, by 0.744 percentage points. Although direct regression has the advantage of predicting arbitrary continuous coordinates, analysis-by-synthesis achieves a lower median error and substantially higher SR@1 m and resolution-aware SR@0.5 m despite being restricted to a discrete candidate grid. The direct method is only marginally better on raw SR@0.5 m. This pattern suggests that the advantage of analysis-by-synthesis arises from room-conditioned acoustic hypothesis testing rather than from a finer output space.

This overall advantage likely arises from how the two methods formulate the inverse problem. Direct regression must learn a single mapping from RIR spectra to coordinates that transfers across different room geometries. Analysis-by-synthesis instead evaluates room-conditioned forward hypotheses and compares them within the current scene, thereby using the test room's geometry, acoustic context, and valid candidate locations explicitly. The results suggest that candidate-conditioned forward prediction provides a more robust basis for localization in unseen rooms than a separately learned coordinate regressor.

Efficiency trade-off. Direct regression predicts the source coordinates from an observed RIR in a single network evaluation, independently of candidate-grid size. Its mean, median, and P90 per-query latencies are 14.438, 14.425, and 14.556 ms, respectively. In contrast, analysis-by-synthesis with the AR-adapted FEW-SHOTRIR (the same method as FEW-SHOTRIR (AR-adapted)) synthesizes and

Table 9: Comparison of FEM-based localization scoring methods. Best results are shown in bold.

Method	Median Localization Error [m] ↓	SR@0.5 m ↑	SR@1.0 m ↑	Resolution-Aware SR@0.5 m ↑
FEM–OMP	**1.975**	**25.781%**	**49.219%**	**36.719%**
FEM–AGREE	2.301	21.875%	44.531%	34.375%

scores an RIR at every candidate location, resulting in mean, median, and P90 per-query latencies of 1.263, 0.400, and 6.457 s, respectively (Table 3). Direct regression is therefore approximately $87.5\times$, $27.7\times$, and $443.6\times$ faster under the three statistics. Although the per-candidate cost of analysis-by-synthesis remains low—1.745 ms on average—its total latency accumulates over the candidate grid. This comparison exposes a clear trade-off: direct regression provides substantially faster inference but requires localization-specific supervision, whereas analysis-by-synthesis achieves stronger overall localization fidelity without training an inverse localization model, at the cost of candidate-wise evaluation.

D.3 Ablation Study of the Scoring Rule in the FEM-Based Method

We isolate the effect of the candidate-scoring rule while keeping the FEM forward model and candidate set fixed. Let $\mathbf{y} \in \mathbb{C}^F$ denote the observed RIR evaluated at the F frequency bins, and let $\mathbf{h}_i \in \mathbb{C}^F$ denote the FEM transfer function for candidate location i.

FEM–OMP scoring. FEM–OMP adapts the single-microphone, wideband pulse-source formulation of Room Helps (Dokmanić & Vetterli, 2012) to the case of one active source. For each candidate location i, the FEM solver predicts the complex acoustic response at the receiver over the $F = 102$ frequency bins in the 80–300 Hz band. We denote this response by $\mathbf{h}_i \in \mathbb{C}^F$. Each $\mathbf{h}_i$ is called an *FEM atom* because it forms one entry of the candidate-response dictionary used by OMP. The observed RIR is represented at the same frequencies by $\mathbf{y} \in \mathbb{C}^F$.

Each candidate is scored using the normalized energy of the projection of the observed response onto its FEM atom:

$$s_i^{\mathrm{OMP}} = \frac{\left|\mathbf{h}_i^H \mathbf{y}\right|^2}{\|\mathbf{h}_i\|_2^2 \|\mathbf{y}\|_2^2}, \qquad \hat{i}_{\mathrm{OMP}} = \arg\max_i s_i^{\mathrm{OMP}}. \tag{16}$$

Here, $\mathbf{h}_i^H$ denotes the conjugate transpose of $\mathbf{h}_i$, $\|\cdot\|_2$ is the Euclidean norm, s_i^{OMP} is the fraction of observed energy explained by candidate i, and $\hat{i}_{\mathrm{OMP}}$ is the selected candidate index. The score preserves relative phase information and is invariant to a global scalar gain. This scorer is deterministic.

FEM–AGREE scoring. The scoring process is the same as described in Section 4.2. For each candidate location $x^{(m)}$, we reconstruct one deterministic band-limited RIR $\widehat{h}_m$ from its FEM response over the 80–300 Hz frequency band. The observed RIR is restricted to the same frequency bins, and the observed and candidate RIRs are independently peak-normalized. The frozen AGREE audio encoder evaluates each candidate once:

$$a_m = \cos\left(E_a(h_{\mathrm{obs}}), E_a(\widehat{h}_m)\right), \qquad \widehat{x}_s = x^{(\arg\max_m a_m)}. \tag{17}$$

Thus, the method selects the candidate whose FEM-simulated RIR is most similar to the observed RIR in the AGREE embedding space. No localization-specific model parameters are learned.

Evaluation protocol. We evaluate the two FEM scoring methods on the same 128 queries from the 16 mesh-available unseen rooms, with eight queries per room. Only the scoring rule and its required representation differ between the two variants. We report median localization error, SR@0.5 m, SR@1.0 m, and resolution-aware SR@0.5 m. The resolution-aware metric counts a prediction as successful when its localization error exceeds that of the nearest available candidate by at most 0.5 m.

Results. FEM–OMP consistently outperforms FEM–AGREE, reducing the median localization error from 2.301 m to 1.975 m and improving SR@0.5 m from 21.875% to 25.781%, SR@1.0 m from 44.531% to 49.219%, and resolution-aware SR@0.5 m from 34.375% to 36.719%. These results indicate that direct complex-frequency scoring is more effective than AGREE-based scoring for FEM localization.