Differentiable Room Acoustic Rendering with Multi-View Vision Priors

Supplementary Material

Contents

A. Method Details

A.1. Acoustic Beam Tracing Algorithm

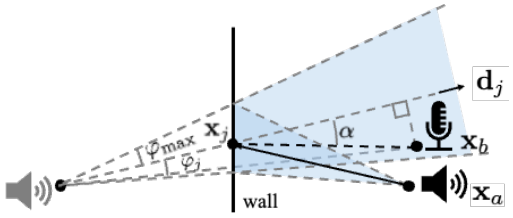

Figure 1. Acoustic beam tracing: in acoustic beam tracing the source and listener are considered as two point, the sound is propagate via a cone-shape beam in space. Acoustic beam tracing handles reflection the same as ray tracing does. The key difference is that acoustic beam tracing enumerate a reflection path if the listener is contatined in the beam volume but not necessarily being hitted by the sampled ray

Given the source location $\mathbf{x}_a$ and listener location $\mathbf{x}_b$, we adopt acoustic beam tracing [2, 4, 7, 14] to sample specular beams in a source-to-listener manner. First we cast N_d beams from the source, using a Fibonacci lattice [3] to approximate uniform coverage of directions. A small apex angle $2\varphi_{\max}$ is selected to ensure the cone-shape beams remain disjoint. Next, each beam's center ray intersects with room geometry to find reflection points (e.g. via Open3D [17]), and after each reflection, we check if the reflected beam can hit the listener. To determine whether a reflected beam at j-th reflection point $\mathbf{x}_j$ (with out-going direction $\mathbf{d}_j$) reaches the listener before hitting another surface, we check if the listener is within the reflected cone (as show in Figure 1). Denote l_j as the distance traveled by reaching $\mathbf{x}_j$, and α_j as the angle between $\mathbf{d}_j$ and the line from $\mathbf{x}_j$ to $\mathbf{x}_b$ and φ_j as the sampled half-apex angle:

$$\varphi_j = \arctan\left(\frac{\|\mathbf{x}_b - \mathbf{x}_j\| \sin\alpha}{\|\mathbf{x}_b - \mathbf{x}_j\| \cos\alpha + l_j}\right). \tag{1}$$

The listener is considered "hit" if α is acute, $\varphi_j < \varphi_{\max}$, and $\mathbf{x}_j$ is visible by $\mathbf{x}_b$. In addition, the time-of-arrival is by:

$$\mathrm{toa}_j = \frac{\|\mathbf{x}_b - \mathbf{x}_j\| \sin\alpha}{v_{\text{sound}} \cdot \sin\varphi_j}. \tag{2}$$

Algorithm 1 summarizes our beam-tracing procedure.

Algorithm 1: Acoustic Beam Tracing

Input: Source $\mathbf{x}_a$, Listener $\mathbf{x}_b$, Geometry $\mathcal{M}$
Output: Specular paths $\{\tilde{\mathbf{x}}_k\}_{k=1}^{N}$
for $i = 1$ *to* N_d **do**
  $\mathbf{x}_{i,0} \leftarrow \mathbf{x}_a$; $l_{i,0} \leftarrow 0$;
  $\mathbf{d}_{i,0} \leftarrow$ `SampleFib`(N_d, i)
end
`ANS` $\leftarrow \{\}$
if *IsVisible*$(\mathbf{x}_a, \mathbf{x}_b)$ **then**
  `ANS.add`$(\varnothing)$ // direct path
end
for $j = 1$ *to* $\mathrm{MAX}_{\text{depth}}$ **do**
  for $i = 1$ *to* N_d **do**
    $[\mathbf{x}_{i,j}, \mathbf{z}] =$ `HitPoint`$(\mathcal{M}, \mathbf{x}_{i,j-1}, \mathbf{d}_{i,j-1})$
    $\mathbf{d}_{i,j} = \mathbf{d}_{i,j-1} - 2\,(\mathbf{z}^\top \mathbf{d}_{i,j-1})\,\mathbf{z}$
    $l_{i,j} = l_{i,j-1} + \|\mathbf{x}_{i,j} - \mathbf{x}_{i,j-1}\|$
    if *BeamHit*$(\mathbf{x}_b, \mathbf{x}_{i,j}, \mathbf{d}_{i,j}, l_{i,j})$ **then**
      `ANS.add`$([\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,j}])$
    end
  end
end
return `ANS`

A.2. Local Variance Derivation

As shown in Figure 2, consider a beam traveling distance l before hitting the surface at $\mathbf{x}$, with half-apex angle φ and local surface normal $\mathbf{z}$. Let θ be the angle between the reflected direction $\mathbf{d}$ and $\mathbf{z}$. In a local coordinate system whose axes are $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{z}\}$, where we requires $\mathbf{t}_1$ aligns with the projection of $\mathbf{d}$ in the tangent surface, the beam's cross-section at distance l is approximately an ellipse with semi-major and semi-minor axes proportional to $l \sin\varphi$, modulated by θ. A simple way to encode this elliptical patch is to

Room	Floor Area	N_{bounce}	N_{basis}	L_{RIR}	$T_{\text{inference}}$	T_{tracing}	T_{res}	T_{train}	N_{params}
HAA-Classroom	$\sim$56m^2	6	2.3K	2.00s	66.0ms	30.4ms	11.2ms	0.59h	26.4M
HAA-Complex	$\sim$106m^2	6	5.1K	2.00s	69.6ms	32.5ms	11.5ms	1.04h	26.4M
HAA-Dampened	$\sim$25m^2	2	1.1K	2.00s	26.8ms	9.58ms	11.5ms	0.36h	26.4M
HAA-Hallway	$\sim$28m^2	6	1.7K	2.00s	67.6ms	29.0ms	11.2ms	1.86h	26.4M
RAF-Furnished	$\sim$44m^2	4	5.9K	0.32s	51.3ms	35.5ms	4.12ms	6.63h	19.5M
RAF-Empty	$\sim$44m^2	4	4.9K	0.32s	54.4ms	35.1ms	4.07ms	10.8h	19.5M

Table 1. Detailed computational breakdown across HAA and RAF scenes. Here, N_{bounce} denotes the number of reflections simulated per beam, N_{basis} is the basis points sampled as described in §A.3, L_{RIR} is the RIR duration, and N_{params} is the total number of trainable parameters in our model. All RIR are sampled at 16kHz. Training times for RAF scenes are reported with 1% training data.

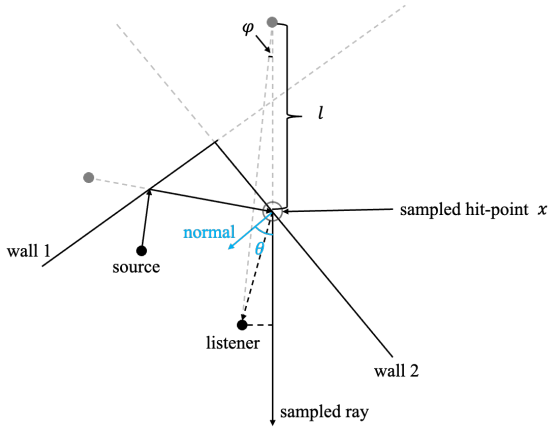

Figure 2. Local covariance derivation: as the traveling space l increases, the region of the contact area expand linearly in terms of radius. In addition, since the half-apex angle is assumed to be small, the contact region is considered an ellipse, which motivates use model the region information with a gaussian distribution.

use a diagonal covariance at local coordinate

$$\Sigma_{\text{local}} = \text{diag}\Big(\sigma_1^2,\ \sigma_2^2,\ 0\Big), \tag{3}$$

where σ_1^2 and σ_2^2 grow with $l \sin\varphi$, adjusted by $\cos\theta$. In the case when φ is small:

$$\sigma_1^2 \approx \big(l \sin\varphi\big)^2 / \cos^2\theta, \quad \sigma_2^2 \approx \big(l \sin\varphi\big)^2 / \cos\theta.$$

These terms capture how the beam's ellipse "stretches" along $\mathbf{t}_1$ and $\mathbf{t}_2$. In world coordinates, the final covariance Σ is simply

$$\Sigma = Q\,\Sigma_{\text{local}}\,Q^\top,$$

where $Q = [\,\mathbf{t}_1\ \mathbf{t}_2\ \mathbf{z}\,]$ rotates from local axes to world axes.

A.3. Basis Points Sampling

we sample the basis point in two steps, first we densely sample 100,000 points on the room geometry, then, we downsample them with voxel size 0.2m and use the median point (closest to mean point) as the basis samples for vision features, as shown in Figure 3, in this way, we ensures the distances between samples are stable.

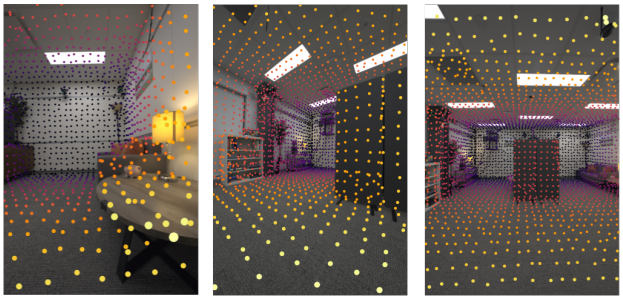

Figure 3. Visualization of surface basis samples for extracting multi-view images features.

A.4. Hyperparameters

Following [15], we use a spherical Gaussian weighting function with a sharpness parameter of 8 for source directional response. We decode the image feature using a 4-layer MLP and sample frequencies from 12 to 7800 Hz with 16 logarithmically spaced samples, linearly interpolating the frequency response.

A.5. Optimization

We optimize the network using the Adam optimizer with a fixed learning rate of 5×10^{-4} (and 1×10^{-4} for the residual component). Our loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{MAG}} + \lambda_{\text{pink}}\mathcal{L}_{\text{pink}} + \lambda_{\text{decay}}\mathcal{L}_{\text{decay}}, \tag{4}$$

where $\mathcal{L}_{\text{MAG}}$ is a multi-scale log L1 loss, $\mathcal{L}_{\text{pink}}$ is the pink noise supervision loss, and $\mathcal{L}_{\text{decay}}$ is the decay loss proposed by [9, 11]. We adopt a progressive training strategy, starting with a reflection order $N = 1$ and increasing by 1 every 100 epochs until $N = 6$. During training, we sample 16,384 points from Fibonacci lattices for beam tracing, reducing this to 8,192 points per RIR during inference. Training is performed with a batch size of 128.

A.6. Computational Cost

Training/Inference Time and Model Size. We measure our model's size and training/inference time against baseline methods, as shown in Table 2. On the HAA Classroom

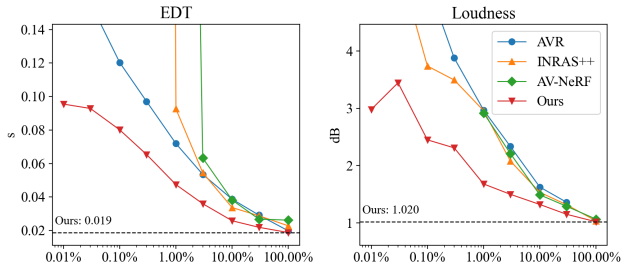

Figure 4. Performance comparison across training scales (from 0.01% to 100% of training data). In addition to the metrics reported in the main paper, our model consistently outperforms the baselines in terms of both EDT and Loudness.

dataset base setting, inference on a 2.0s, 16kHz RIR using a single RTXA6000 takes 66ms. Our approach achieves the fastest inference among existing physics-based methods (*i.e.*, DiffRIR and AVR).

Method	$T_{\text{inference}}$	T_{training}	Size
DiffRIR	376 ms	3.55 h	34.4 K
AVR	100 ms	0.17 h	45.7 M
Ours	66 ms	0.59 h	26.4 M

Table 2. Training time, inference time, and model size comparison on HAA Classroom (2.0s IR, 16kHz, same RTX A6000)

Scene-Level Breakdown. Table 1 decomposes the computation cost across additional HAA and RAF scenes. For each scene, we report the total inference time $T_{\text{inference}}$, which includes beam-tracing (T_{tracing}), residual rendering (T_{res}), and early-stage RIR rendering[1]. Inference time remains below 70ms for all HAA rooms and below 55ms for the RAF dataset, demonstrating that our model naturally extends with scene complexity. As an additional sanity check, we also test our framework on a much larger scene—Gibson Hennepin [16] ($\sim$600m^2; 69k points; 6 bounces), our model costs 108ms per 8s, 16kHz RIR render.

B. Additional Results

B.1. Waveform Comparison

Figure 5 shows wave visualizations on the Hearing Anything Anywhere dataset. All models were trained on only 12 data points. Our model significantly outperforms the baselines in preserving the wave structure, producing a wave front that closely matches the ground truth in terms of peak locations and magnitudes. Note that quantitative metrics do not always capture these perceptual differences; some methods may achieve low error values despite generating distorted wave patterns. This comparison highlights the superior capability of our approach in modeling acoustic dynamics in few-shot settings.

Figure 6 presents wave visualizations on the Real Acoustic Field dataset. Here, we compare three baseline models trained on 1% of the data with our model trained on both 1% and 0.1% of the data. Our results demonstrate that, in terms of wave structure, our model achieves better peak alignment and peak magnitude than the baselines—even when our model is trained on only 0.1% of the data. When trained on 1% of the data, our method further outperforms the baselines.

B.2. Multi-scale Performance Comparison

Loudness and EDT Errors. Figure 4 extend the multi-scale performance comparison in main paper by evluating on two more metrics, i.e., Loudness and EDT. The result shows that our model performs consistantly better than baselines in all training data scale, which is aligned with our observation in the main paper.

Initial Drop in T60 and Loudness Errors. We discovered one of 9 RIRs in the 0.0003% subset was invalid due to speaker failure, resulting in an almost silent recording.Excluding it, the T60 and Loudness errorss (15.7% and 2.74dB, respectively) restore the expected monotonic decrease with larger dataset size. Only 0.27% of RAF data were similarly affected; all other conclusions remain valid.

B.3. Full Metrics on the HAA Dataset

Table 4 present the complete evaluation metrics on the HAA dataset, including Loudness, C50, EDT, and T60 across four scenes. Our results show that our method outperforms state-of-the-art baselines across almost all metrics, confirming the trends observed in the main paper. The only exception is the C50 metric and EDT metric in the *Hallway* scene, where AV-NeRF performs particularly well, likely due to its effective use of depth information in this constrained geometry. These comprehensive results validate the robustness and effectiveness of our model in diverse real-world acoustic environments.

Variant	C50	EDT	T60
65 views	1.98	80.1	15.2
20 views	2.01	80.9	15.7
10 views	2.13	97.9	15.2
5 views	2.12	97.2	15.3
ResNet18	1.96	89.4	15.3

Table 3. Ablation study on vision features. “65 views” denotes using 65 images for training; “20 views”, “10 views”, and “5 views” denote reduced image sets. “ResNet18” indicates replacing the DINO-V2 encoder with ResNet18.

[1]We do not include it in the comparison, as isolating it would require caching the full beam trace for every source-listener pair, which is prohibitively memory intensive.

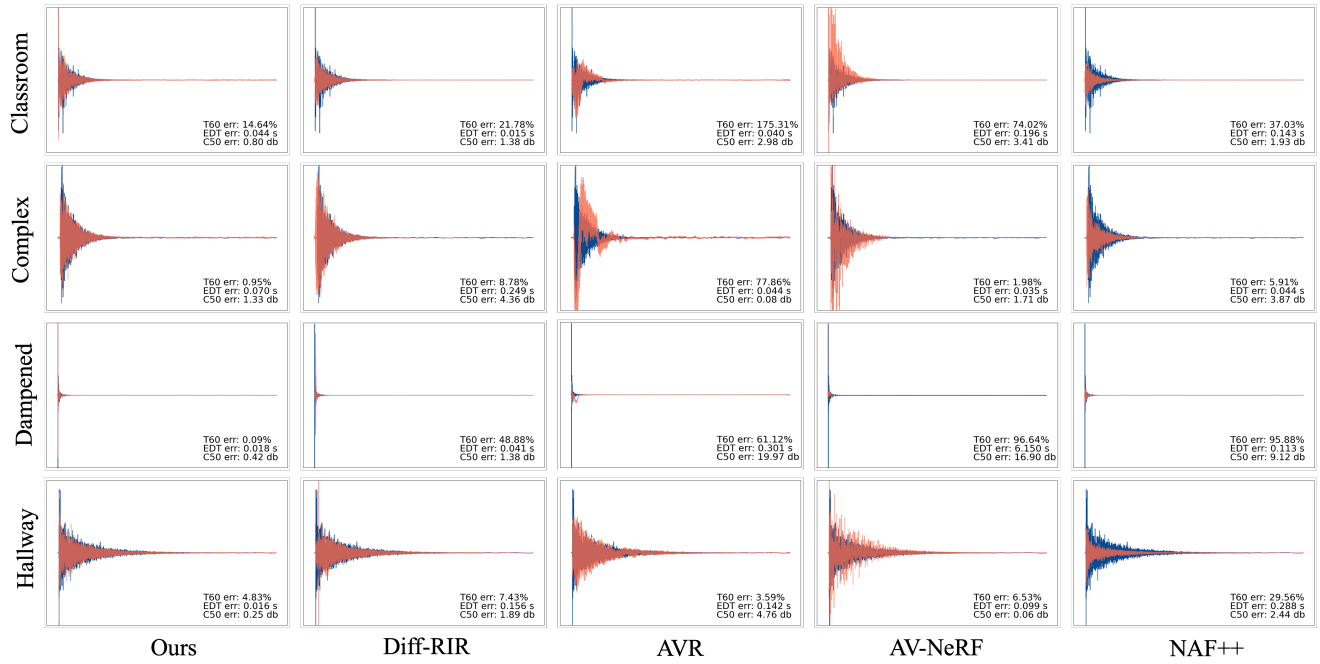

Figure 5. Wave visualization on the Hearing Anything Anywhere dataset [15]. All models are trained on 12 data points. Our model significantly outperforms all baselines in preserving the wave structure—producing the most faithful wave front with accurate peak locations and magnitudes. Note that quantitative metrics do not always capture these perceptual details; some methods may have low error values despite producing distorted wave patterns.

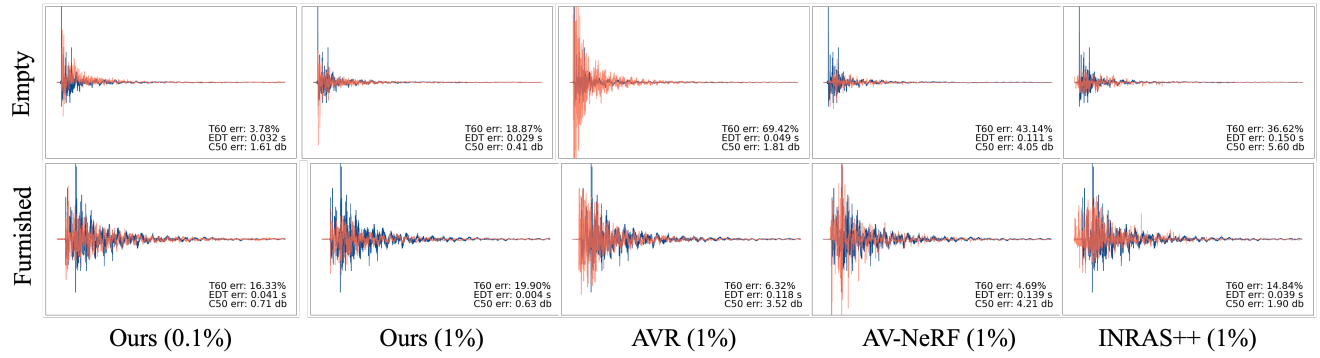

Figure 6. Wave visualization on the Real Acoustic Field dataset [1]. We show results from three baseline models trained on 1% of the data alongside our model trained on 1% and 0.1% of the data. Our model exhibits better peak alignment and magnitude than baseline methods—even when trained on only 0.1% of the data—and significantly outperforms all baselines when using the same amount of training data.

B.4. Ablations on Vision Features

We investigate the impact of vision features by varying two aspects: the number of multi-view images used for training and the choice of the pretrained encoder. Both experiments are conducted on the RAF Furnished scene using only 0.1% of the training data.

Table 3 shows our vision feature saturation experiment, we initially use 65 images to cover the entire scene, then reduce the number to 20, 10, and 5 views (see top four rows of Table 3). Reducing from 65 to 20 views incurs less than a 1% drop in C50 and EDT, but further reduction from 20 to 10 views causes a marked performance decline, indicating that adequate view redundancy is essential for effective visual guidance. Performance remains stable when further reduced from 10 to 5 views, suggesting that with only 10 views the model nearly abandons visual feature learning and relies primarily on acoustic cues.

We also replace the DINO-v2 [12] encoder with ResNet18 [5], which results in a noticeable drop in EDT, demonstrating that DINO-V2 is better suited for our model.

Method	Classroom				Complex Room			
	Loudness (dB) ↓	C50 (dB) ↓	EDT (ms) ↓	T60 (%) ↓	Loudness (dB) ↓	C50 (dB) ↓	EDT (ms) ↓	T60 (%) ↓
NAF++ [10]	8.27	1.62	162.3	134.0	4.43	2.25	203.5	44.8
INRAS++ [13]	1.31	1.86	110.0	60.9	1.65	2.26	150.7	29.5
AV-NeRF[8]	1.51	1.43	77.8	50.0	2.01	1.88	107.9	36.6
AVR [6]	3.26	4.18	135.6	44.3	6.47	2.55	138.3	36.7
Diff-RIR [15]	2.24	2.42	139.7	39.7	1.75	2.23	129.5	18.5
Ours	0.99	1.02	55.5	24.3	0.98	1.44	86.5	10.8

Method	Dampened Room				Hallway			
	Loudness (dB) ↓	C50 (dB) ↓	EDT (ms) ↓	T60 (%) ↓	Loudness (dB) ↓	C50 (dB) ↓	EDT (ms) ↓	T60 (%) ↓
NAF++ [10]	3.88	4.24	360.0	306.9	8.71	1.36	148.3	21.4
INRAS++ [13]	3.45	3.28	187.1	382.9	1.55	1.87	157.9	7.4
AV-NeRF [8]	2.40	3.05	242.1	107.9	1.26	1.03	89.9	9.5
AVR [6]	6.65	11.11	305.3	81.4	2.48	2.69	195.4	7.0
Diff-RIR [15]	1.87	1.56	153.0	44.9	1.32	3.13	188.1	6.8
Ours	1.11	1.45	139.0	31.9	0.85	1.15	96.5	6.3

Table 4. Result on the HAA [15] dataset, 2.0s, 16K sample rate

Notably, all vision ablations have minimal impact on T60, indicating that vision features primarily contribute to modeling early reflection rather than late reverberation.

B.5. Detailed Analysis on Model Components

Table 5 summarizes the impact of ablating individual components of our model.

Differentiable Renderer. Replacing the learned residual field with a position-independent on (*Uni. Residual*) increases EDT error by more than 30%, and removing the residual entirely (*w/o Residual*) raises C50, EDT, and T60 errors by over 70%. Substituting our beam-tracing method with conventional ray tracing (*Ray-Tracing*) worsens all three metrics by more than 40%. Using plain Fourier features instead of integrated positional encoding (*w/o IPE*) raises EDT by 26.3%.

Variant	C50	EDT	T60
Ours (full)	1.98	80.1	15.2
Uni. Residual	2.11	106.4	13.9
w/o Residual	3.82	142.8	49.0
w/o Vision	2.13	98.6	14.3
Ray-Tracing	4.27	146.9	21.9
w/o IPE	2.10	101.2	15.0

Table 5. Ablation study results. See text for details.

Vision Encoder. Replacing vision encoder by zero vectors (*w/o Vision*) degrades EDT error by roughly 23%, confirming the importance of importance of visual information for accurate acoustic estimation.

These results confirm that each design choice contributes substantially to the overall performance.

B.6. Failure Cases

In Figure 7, we show a failure case where the source and listener are close. Accurately predicting the first-arrival spike is challenging due to its narrow ROI, which limits gradient flow. Nonetheless, our method still outperforms the baseline.

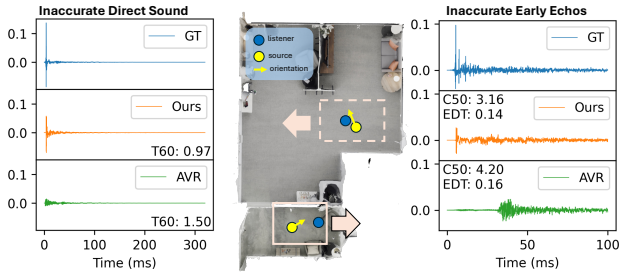

Figure 7. We visualize two failure cases in our model on RAF-Furnished Room with 1% training data.

References

[1] Ziyang Chen, Israel D Gebru, Christian Richardt, Anurag Kumar, William Laney, Andrew Owens, and Alexander Richard. Real acoustic fields: An audio-visual room acoustics dataset and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21886–21896, 2024. 4

[2] Thomas Funkhouser, Ingrid Carlbom, Gary Elko, Gopal Pingali, Mohan Sondhi, and Jim West. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 21–32, New York, NY, USA, 1998. Association for Computing Machinery. 1

[3] Douglas P. Hardin, Timothy Michaels, and Edward B. Saff. A comparison of popular point configurations on $\mathbb{S}^2$. *Dolomites Research Notes on Approximation*, 9, 2016. 1

[4] John Kenneth Haviland and Balakrishna D. Thanedar. Monte carlo applications to acoustical field solutions. *The Journal of the Acoustical Society of America*, 54(6):1442–1448, 12 1973. 1

[5] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 4

[6] Zitong Lan, Chenhao Zheng, Zhiwei Zheng, and Mingmin Zhao. Acoustic volume rendering for neural impulse response fields. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 5

[7] Christian Lauterbach, Anish Chandak, and Dinesh Manocha. Interactive sound rendering in complex and dynamic scenes using frustum tracing. *IEEE Transactions on Visualization and Computer Graphics*, 13:1672–1679, 2007. 1

[8] Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Av-nerf: Learning neural fields for real-world audio-visual scene synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 5

[9] Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Neural acoustic context field: Rendering realistic room impulse response with neural fields. *ArXiv*, abs/2309.15977, 2023. 2

[10] Andrew Luo, Yilun Du, Michael Tarr, Josh Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *Advances in Neural Information Processing Systems*, 35:3165–3177, 2022. 5

[11] Sagnik Majumder, Changan Chen, Ziad Al-Halah, and Kristen Grauman. Few-shot audio-visual learning of environment acoustics. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2

[12] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 4

[13] Kun Su, Mingfei Chen, and Eli Shlizerman. INRAS: Implicit neural representation for audio scenes. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 5

[14] Dirk van Maercke and Jacques Martin. The prediction of echograms and impulse responses within the epidaure software. *Applied Acoustics*, 38(2):93–114, 1993. 1

[15] Mason Wang, Ryosuke Sawata, Samuel Clarke, Ruohan Gao, Shangzhe Wu, and Jiajun Wu. Hearing anything anywhere. In *CVPR*, 2024. 2, 4, 5

[16] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 3

[17] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 1