Hearing Anything Anywhere
– Supplementary Material –

Mason Long Wang[1*] Ryosuke Sawata[1,2*] Samuel Clarke[1]
Ruohan Gao[1,3] Shangzhe Wu[1] Jiajun Wu[1]
[1]Stanford University [2]Sony AI [3]University of Maryland, College Park
masonlwang.com/hearinganythinganywhere

Contents

A Qualitative Results and Video 1
B RIR Heatmap Visualizations 1
B.1. Broadband RIR Heatmaps 1
B.2. Soundfield Reconstruction 2
C Results on Additional Room Configurations 2
C.1. Description of Additional Subdatasets 2
C.2. Evaluations on Configurations 4
C.3. Quantitative Results on Virtual Room Layout Modifications 4
D Additional Experiments and Ablations 6
D.1. Results on Binaural Rendering 6
D.2. Performance vs Number of Training Points . 6
D.3. Robustness to Inaccurate Geometry. 6
D.4. More Ablations 7
D.5. Modeling the Effects of Transmissions . . . 7
D.6. Comparison to Traditional Acoustic Simulations 8
E Method Details 8
E.1. Details on Source Localization 8
E.2. Minimum-Phase Transform 9
E.3. Specific Loss Formulation 9
E.4. Small Efficiency and Performance Boosts . . 9
E.5. Computational Cost 10
F. Baseline Implementation Details 10
G Data Collection Procedure Details 12
G.1. Estimating the Room Impulse Response (RIR) 12
G.2. Room Geometry Estimation 12
H Guidelines for Microphone Placement. 12

A. Qualitative Results and Video

Please see the supplementary video on the website for an in-depth qualitative analysis and comparative evaluation against baseline models. This video showcases a simulation of a song played in two distinct environments: the Dampened Room and the Hallway. The purpose is to demonstrate the immersive quality and perceptual accuracy of the audio rendered by our model, reflecting the true characteristics of the real scenes. To achieve this, we rendered 100 room impulse responses at various locations, convolved them with the chosen source audio, and smoothly interpolated between these convolved signals. For an optimal experience of these qualitative results, we recommend using earbuds or headphones while viewing the video.

Furthermore, the video features a side-by-side comparison of our binaural audio results with those from baseline models, highlighting the enhanced realism and compelling nature of the audio generated by our model. This comparison underscores the significant qualitative improvements our model offers in creating an immersive auditory experience. In addition, the video provides visualizations explaining our method, and the task setup.

B. RIR Heatmap Visualizations

B.1. Broadband RIR Heatmaps

After our model is trained, we can use it to visualize how the loudness of the rendered acoustic field varies spatially. To do this, we use the model trained on each of the base subdatasets to render RIRs on a dense 2D-grid of listener locations. We visualize of the root mean square (RMS) volume level of the RIRs in Figure 1, on a decibel (logarithmic) color scale. The visualizations shown are similar to those in [19, 27].

We observe several differences in the heatmaps for the different rooms. In the Dampened Room, the surfaces are less reflective, and thus, much of the soundfield's loudness is concentrated in the region in front of the speaker. This

[*] Equal contribution.

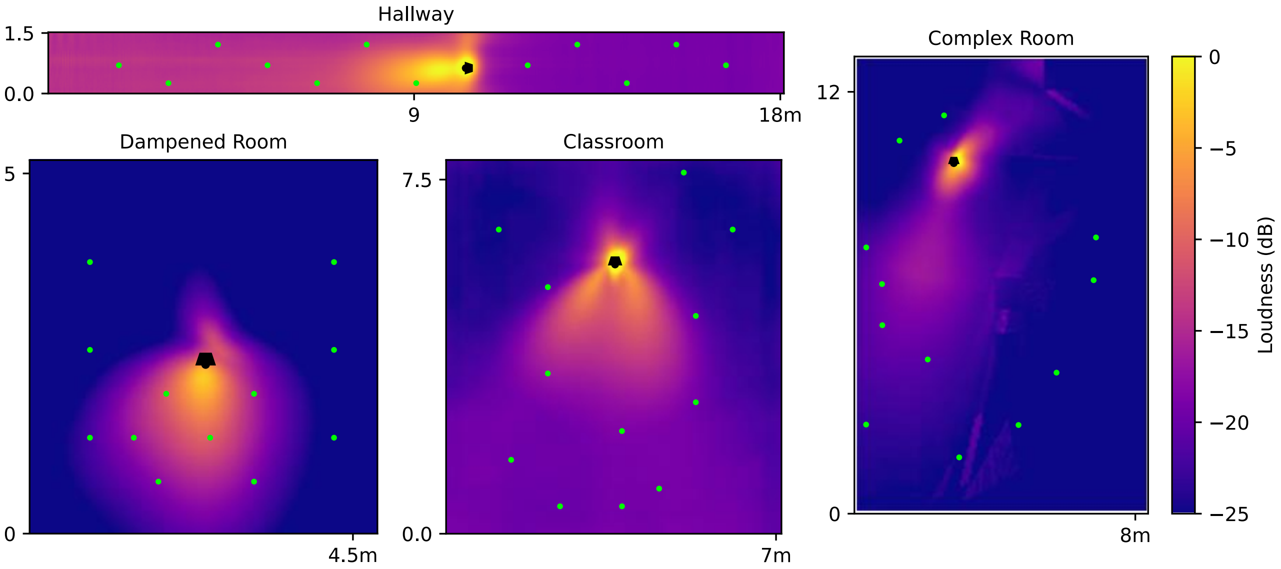


Figure 1. Visualization of RIR loudness maps generated from our model trained in each of the four base subdatasets. We measure loudness by rendering an RIR at a given listener location and measuring its RMS volume level. For each RIR rendered, we fix the height of the listener location to be 1 meter above the floor. The resolution of each xy-grid is approximately 5 centimeters in both the x and y directions. We fix the location and orientation of the speaker (indicated by the black icon) to where it was during RIR measurement. The color scale is in decibels and is consistent between rooms. The green dots indicate the xy locations of the 12 training points, which are projected onto the $z = 1$ plane.

effect is reduced in the Classroom, where the soundfield is more spread out. In the Hallway, which is the most reflective room, the soundfield's volume is even more spread out, and the region behind the speaker is significantly louder than it is in any of the other rooms.

B.2. Soundfield Reconstruction

When observed at a single frequency, the spatial variations in sound pressure for a given sound field often exhibit modal patterns. Reconstructing the pressure levels of a sound field from a sparse set of observations is a problem of longstanding theoretical and practical interest [1, 5, 10, 18]. Using the RIRs measured in the Classroom subdataset, we calculate the sound pressure level at 70 Hz at all locations in our subdataset, plotted in Figure 2a. We also use the predicted RIRs from each method to predict the sound pressure level at 70 Hz at every spatial location. We find that our model learns to predict the modal structure of the RIR sound field without explicitly modeling it, while other baselines fail to do this. Note that our model approximately predicts the locations of the sound field's nodes and anti-nodes (regions of high and low intensity), even without observing training data in those locations.

C. Results on Additional Room Configurations

C.1. Description of Additional Subdatasets

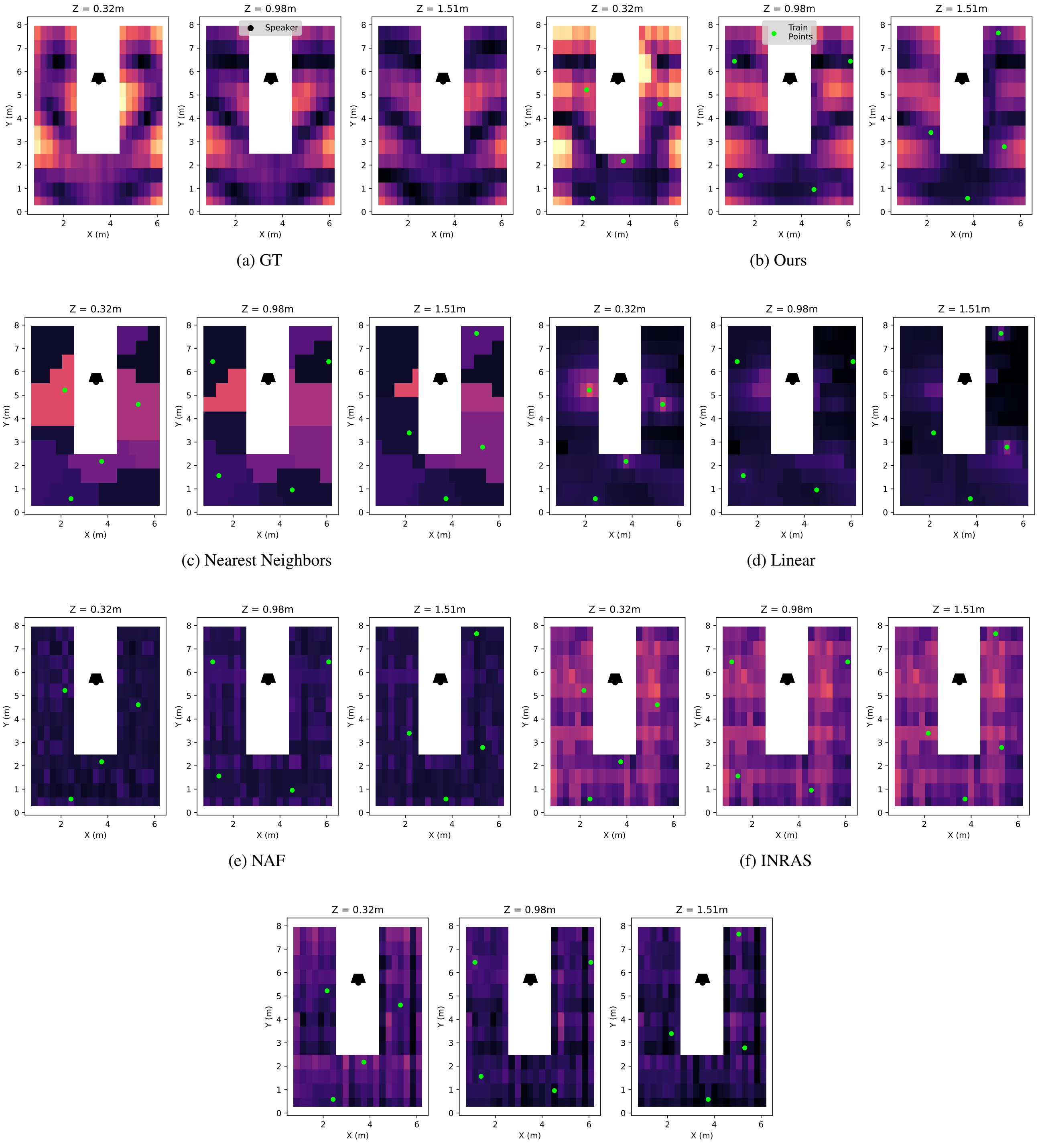
In addition to the base subdatasets collected in each of the four rooms, we collect additional data in different room configurations, where we vary the location of the speaker, the orientation of the speaker, or the presence and number of rectangular whiteboard panels. We collect this additional data for two reasons:

- To test our method's effectiveness on various room layouts, including those where the speaker is occluded.
- To evaluate acoustic interpolation methods on the task of zero-shot generalization to changes in room layouts, by virtually simulating speaker rotation and translation, and panel relocation and insertion.

The locations and orientations of the speakers as well as the positions of the panel(s), are provided as part of the DiffRIR Dataset. Photographs of each additional configuration are shown in Figure 3.

Rotation Subdatasets. In the Dampened Room, Hallway, and Complex Room, we collected 120, 72, and 132 additional datapoints where the speaker was rotated by 225°, 90°, and 90°clockwise, respectively. The location of the speaker and all surfaces otherwise remain the same.

Translation Subdatasets. In the Dampened Room, Hallway, and Complex Room, we collected 120, 72, and 132



(a) GT

(b) Ours

(c) Nearest Neighbors

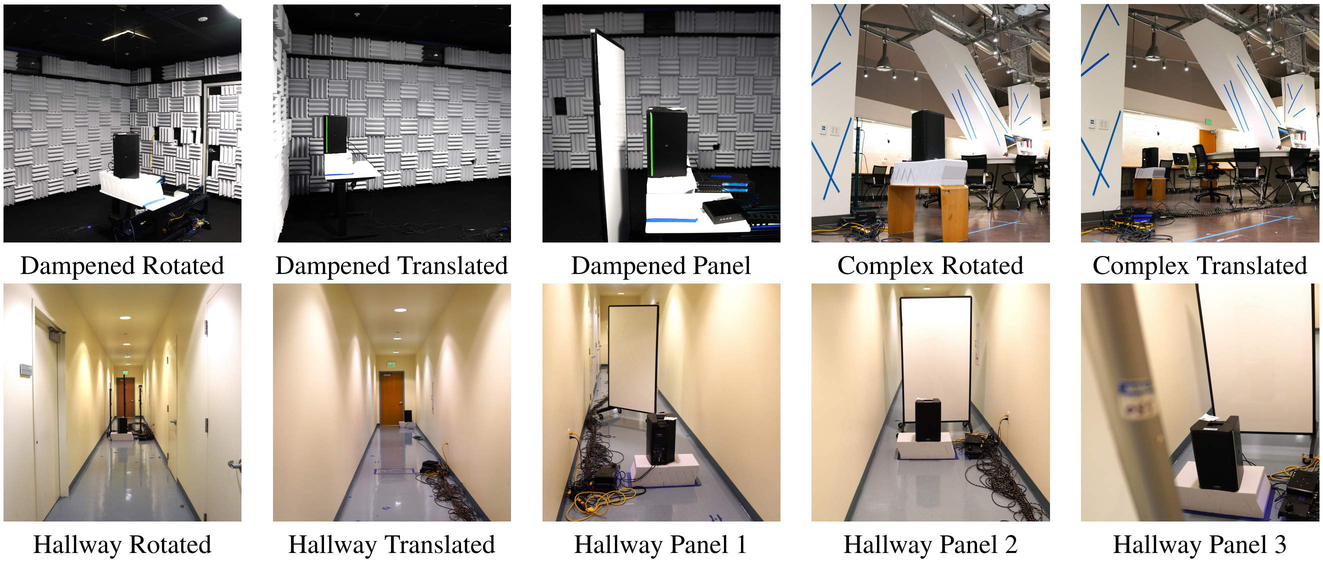
(d) Linear

(e) NAF

(f) INRAS

(g) DeepIR

Figure 2. Visualization of RIR loudness at 70 Hz in the Classroom subdataset. The sound field intensity at a given location is measured by filtering the ground-truth or predicted RIR around 70 Hz using a 2nd order Butterworth filter [4] and measuring the RMS volume level of the filtered signal. Subfigure a) shows the intensity of the 70hz sound field at all locations in the subdataset. Subfigure b) shows predicted intensities at these same locations using our model trained on 12 points. We indicate the spatial locations of these 12 training points with green dots, and the speaker's location and orientation with a black icon. Subfigures c) through g) show the sound field intensity as predicted by each of our baseline models. Note that in subfigure d), the Linear baseline underestimates the soundfield intensity at locations far away from the training locations, since the linear interpolation at these locations is a weighted average of roughly uncorrelated signals whose mean is roughly zero.



Dampened Rotated
Dampened Translated
Dampened Panel
Complex Rotated
Complex Translated
Hallway Rotated
Hallway Translated
Hallway Panel 1
Hallway Panel 2
Hallway Panel 3

Figure 3. Photographs of all additional configurations in the DiffRIR Dataset. Note that the Hallway Panel 1 photo is taken from behind the speaker.

additional datapoints where the speaker was translated to another part of the room, but the orientation of the speaker is was kept the same. In the Dampened Room, we move the speaker such that it is near one corner of the room and facing a wall. In the Hallway, we move the speaker to the far end of the Hallway, such that the speaker faces the entire length of the Hallway. The Complex Room is roughly divided into two halves by the table and pillars in the middle of the room. In the Complex Room, we collect additional datapoints where the speaker is translated from the one half to the other.

Panel Subdatasets. In the Dampened Room and Hallway, we place 1-2 whiteboard panels in the room. In the Dampened Panel subdataset, we place the panel directly in front of the speaker. In the Hallway subdataset, there are three panel configurations. In Hallway Panel 1, we place one whiteboard panel in front of the speaker at a slanted angle. In Hallway Panel 2, we place one whiteboard panel directly behind the speaker. In Hallway Panel 3, we place whiteboard panels both in front of and behind the speaker.

C.2. Evaluations on Configurations

We evaluate our model on each of these configurations independently in Table 1, training and testing on the same subdataset. For each configuration, we select 12 training points from each of the subdatasets, and evaluate our rendered RIRs on a test set of held-out data.

	RIR		Music	
Room/Configuration	Mag	ENV	Mag	ENV
Dampened	1.21	0.56	1.59	1.19
w/ Rotated Speaker	1.14	0.44	1.49	1.36
w/ Translated Speaker	0.68	0.39	0.91	1.18
w/ Panel	1.23	0.60	1.62	1.47
Hallway	9.13	2.95	2.59	1.25
w/ Rotated Speaker	8.40	2.86	2.58	1.27
w/ Translated Speaker	8.91	3.02	2.84	1.25
Panel Config. 1	8.47	2.99	2.58	1.32
Panel Config. 2	8.52	3.61	2.63	1.36
Panel Config. 3	8.39	2.94	2.67	1.35
Complex	4.86	0.92	2.25	1.41
w/ Rotated Speaker	4.33	0.83	2.13	1.41
w/ Translated Speaker	4.38	1.19	2.22	1.44

Table 1. DiffRIR's performance on additional configurations in the DiffRIR Dataset, on the task of predicting monaural RIRs and music at an unseen point. Lower is better for all metrics. Errors for RIRs are multiplied by 10. Each DiffRIR model is trained on 12 points.

C.3. Quantitative Results on Virtual Room Layout Modifications

Since our model learns physically interpretable parameters for the speaker's directivity, we expect to be able to virtually simulate rotations or translations of the speaker that are unobserved in the training data. We simulate rotating the speaker by rotating the speaker's learned directivity map, and translation by moving the speaker's estimated location

	RIR		Music	
Room/Configuration	Mag	ENV	Mag	ENV
Dampened w/ Rotation				
Trained on Rot. Data	**1.14**	**0.44**	**1.49**	**1.36**
Trained on Base w/ Virt. Rot.	1.39	0.51	1.88	1.48
Hallway w/ Rotation				
Trained on Rot. Data	**8.40**	**2.86**	**2.58**	**1.27**
Trained on Base w/ Virt. Rot.	9.83	3.22	2.88	2.50
Complex w/ Rotation				
Trained on Rot. Data	**4.33**	**0.83**	**2.13**	**1.41**
Trained on Base w/ Virt. Rot.	4.84	0.89	2.27	1.59

Table 2. Results on Virtual Speaker Rotation. Evaluations are done on the test set of the rotated subdataset.

	RIR		Music	
Room/Configuration	Mag	ENV	Mag	ENV
Damp. → Hall. 1.				
Hall. 1 Model	**8.47**	**2.99**	**2.58**	**1.32**
Virtual Insertion	9.32	2.96	2.69	1.33
Damp. → Hall. 2.				
Hall. 2 Model	**8.52**	**3.61**	**2.63**	**1.36**
Virtual Insertion	9.31	3.45	2.62	1.38
Hall. 1.→ Damp.				
Damp. Panel Model	**1.23**	**0.600**	**1.62**	**1.47**
Virtual Insertion	1.84	0.660	3.70	1.56
Hall. 2.→ Damp.				
Damp. Panel Model	**1.23**	**0.600**	**1.62**	**1.47**
Virtual Insertion	1.84	0.660	3.70	1.56

Table 3. Results on Virtual Panel Insertion. 'Damp.→Hall 1.' means that take the DiffRIR model from the Hallway Base subdataset (no panel). Then, we virtually insert a panel to simulate the Hallway Panel 1 subdataset, by borrowing the reflection coefficients of the panel from the DiffRIR model trained on the Dampened w/ Panel subdataset. We then evaluate the virtual insertion on the recordings from the Hallway Panel 1 subdataset. As a baseline, we compare to a model that is trained on the same panel subdataset that it is tested on.

during path-tracing.

These predicted changes in the speaker's location or orientation can be evaluated against real data, since the DiffRIR Dataset contains additional configurations that modify the base subdataset in each room by moving or rotating the speaker.

The quantitative results in Tab. 2, 3, 4, and 5 show the usefulness of the DiffRIR Dataset in benchmarking the performance of methods of virtual room layout modification. Future work can use the DiffRIR Dataset to improve the performance of these tasks.

	RIR		Music	
Room/Configuration	Mag	ENV	Mag	ENV
Hall. 1.→ Hall. 2.				
Baseline	**8.52**	**3.61**	**2.63**	**1.36**
Virtual Panel Relocation	8.91	3.59	2.71	1.39
Hall. 2.→ Hall 1.				
Baseline	**8.47**	**2.99**	**2.58**	**1.32**
Virtual Panel Relocation	8.89	3.13	2.72	1.39

Table 4. Results on Virtual Panel Relocation. 'Hall 1.→ Hall 2.' means that take the DiffRIR model from the Hallway Panel 1 subdataset (no panel). Then, we virtually move this panel to its location in the Hallway Panel 2 subdataset. We then evaluate on the recordings from the Hallway Panel 2 subdataset.

	RIR		Music	
Room/Configuration	Mag	ENV	Mag	ENV
Dampened w/ Translation				
Trained on Trans. Data	**0.68**	**0.39**	**0.91**	**1.18**
Trained on Base w/ Virt. Trans.	1.22	0.53	1.26	1.61
Hallway w/ Translation				
Trained on Trans. Data	**8.91**	**3.02**	**2.84**	**1.25**
Trained on Base w/ Virt. Trans.	9.28	3.05	**2.84**	1.28
Complex w/ Translation				
Trained on Trans. Data	**4.38**	**1.19**	**2.22**	**1.44**
Trained on Base w/ Virt. Trans.	4.79	**1.19**	2.24	1.54

Table 5. Results on Virtual Speaker Translation. Evaluations are done on the test set of the translated subdataset.

Virtual Speaker Rotation. As an experiment, we take the DiffRIR model trained on each base subdataset with a corresponding rotated subdataset, virtually rotate the speaker by rotating the learned directivity heatmap, and predict RIRs and music at locations in each of the corresponding rotated subdatasets. We evaluate these predictions against ground-truth RIRs and music recordings from the rotated subdatasets. In addition, we compare our virtual rotation with the performance of the DiffRIR model both trained and tested on the rotated subdatasets. The results are shown in Table 2. Although the model both trained and tested on the rotated subdatasets outperforms our virtually-rotated model, the results are quite close in the Dampened and Complex Rooms. The results in the Hallway are worse, perhaps because the Hallway's narrow nature means that the set of direct paths from the speaker to the training locations cover a narrow range of outgoing angles.

Virtual Speaker Translation. We perform a similar experiment with virtual speaker translation, evaluating against ground-truth recordings from the corresponding subdataset. The results are shown in Table 5.

Virtual Panel Relocation. We would like to see if we can learn the reflective characteristics of a surface in one room, then 'virtually move' the surface to another location in the same room. In the Hallway, we collect two subdatasets (Hallway Panel 1 and Hallway Panel 2 in Figure 3), where the room layouts are identical except for the location and orientation of a single whiteboard panel. In our experiments, we train on the first panel configuration, then move the location of the whiteboard panel to that of the second configuration before performing inference. We then evaluate our predicted audio against ground-truth audio from the second configuration. Results are shown in Table 4. The baseline shown is one where we train on the same subdataset that we evaluate on.

Virtual Panel Insertion. We would like to see if we can learn the reflective characteristics of a surface in one room, then 'virtually insert' the surface into another room. Three of our base subdatasets also include a version with a single inserted whiteboard panel. In each of our four experiments, we take the base subdataset (e.g., the Dampened Base subdataset), and the coefficients learned for the whiteboard panel from another room (e.g., the Hallway Panel Config. 1 subdataset). We then virtually insert the whiteboard panel into the base subdataset, and evaluate the virtual insertion against the version of the base dataset with a panel in it (e.g., the Dampened Panel subdataset). Results are shown in Table 3. The baseline shown is one where we train on the same subdataset that we evaluate on.

D. Additional Experiments and Ablations

D.1. Results on Binaural Rendering

We evaluate our method on the task of rendering a binaural RIR at an unseen location. We collect binaural RIRs at several locations in all rooms using our 3Dio binaural microphone, and compare these to predicted RIRs that we binauralize from single-channel audio as described in the Methods section.

We compare our binauralized audio with the ground-truth audio using the left-right energy ratio error between the ground-truth and predicted recordings, which is used in [7]. To compute the left-right energy ratio, we compute compute the ratio of total energy between the left and right channels of the RIR or music recordings. We then compute the mean squared error between the left-right energy ratio of the predicted and ground-truth RIRs or music. Results are shown in Table 6.

Since the baselines do not have a way of generating binaural RIRs from monaural ones, we binauralize these baselines by rendering two monaural RIRs at the locations of the left and right ears of the 3Dio microphone, and combining them into left and right channels.

Our method outperforms our baselines across most metrics. Note that it is difficult to compare a binaural RIR recorded from our binaural microphone with binauralized audio originally recorded from a different microphone. Our rendered binaural audio will have characteristics of the monaural microphone and the microphones used in the SADIE dataset [2] used to record the HRIRs that we convolve our monaural recordings with. The binaural recordings in our dataset will h ave different characteristics, since they are recorded using a different microphone with different spectral characteristics and directionality. Because of this, we include qualitative binauralization results in the supplementary video.

D.2. Performance vs Number of Training Points

We conduct an ablation study with varying numbers of training points N on each subdataset and compare against the baselines. As shown in Figure 4, the performance increases with N, and our model consistently outperforms the baselines when $N \geq 2$. Note that in all rooms, our model trained on only 6 locations outperforms all baselines trained on 100.

Note that our model's hyperparameters are optimized for performance in data-limited scenarios. When the number of training points is higher, it is possible that increasing the number of parameters (for instance, increasing the resolution of the heatmap or the number of reflection coefficients) leads to even better performance.

D.3. Robustness to Inaccurate Geometry.

Our method requires measuring the room's geometry. In our dataset, we do this using a tape measure or laser distance measure, which both provide sufficiently accurate measurements. In order to explore the effect of inaccurate geometric measurements, we conduct an additional experiment to measure the performance after adding random artificial distortions to the surfaces. In the Classroom, we select 8 random directions to move each of the 11 vertices defining the walls, ceiling, floors and the corners of the tables that are exposed. We move each vertex by 0-2 meters in its corresponding random direction. Results are shown in Figure 5. Observe that unless we distort *all* vertices in the room by over 1.5 meters, our model outperforms the best baseline (Nearest Neighbors). We conclude that our method is robust to geometric distortion.

Geometric distortion can affect our model's rendering in one of three ways: It can change the distance of reflection paths, which affects its time-of-arrival and amplitude; it can eliminate reflection paths, or it can add new reflection paths. Since our model is optimized against a frequency-domain loss whose smallest window size is 256 samples (or 1.8 meters at the speed of sound), our model is robust to perturbations in times-of-arrival.

	Classroom		Dampened Room		Hallway		Complex Room	
	RIR	Music	RIR	Music	RIR	Music	RIR	Music
NN	1.27	0.516	5.64	2.57	0.062	0.034	0.345	0.166
Linear	1.29	0.531	5.48	2.09	**0.045**	**0.008**	0.335	**0.157**
DeepIR	1.10	0.529	6.20	5.90	0.048	0.036	0.350	0.397
NAF	1.93	0.743	5.93	2.37	0.108	0.012	0.320	0.176
INRAS	1.25	0.383	5.86	4.35	1.60	4.41	0.332	0.183
DiffRIR (ours)	**0.43**	**0.091**	**2.94**	**0.316**	0.097	0.012	**0.287**	0.288

Table 6. Experimental results from the task of predicting binaural RIRs and music at an unseen point from a model trained on 12 monoaural RIRs. We use the left-right energy ratio error metric [7]. Lower is better. All errors are multiplied by 10.

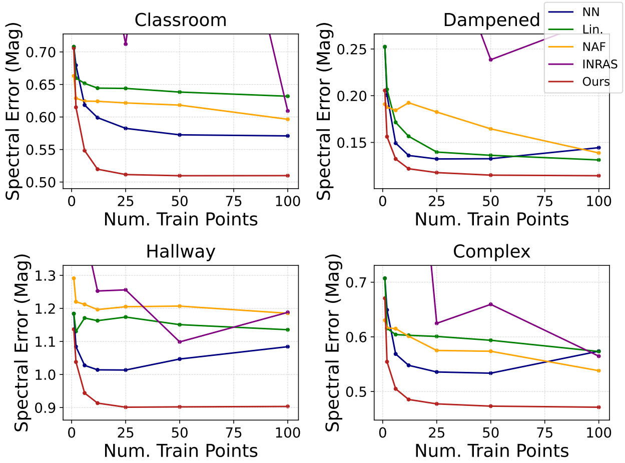


Figure 4. Evaluations of our method and baselines with different numbers of training points. We use the Multiscale Log-Spectral L1 Loss (Mag), and train with $N \in \{1, 2, 6, 12, 25, 100\}$. All training locations are selected as nested subsets of one another, and we evaluate on a fixed test set. Note that the DeepIR baseline's error was too large to fit into the range of the plot.

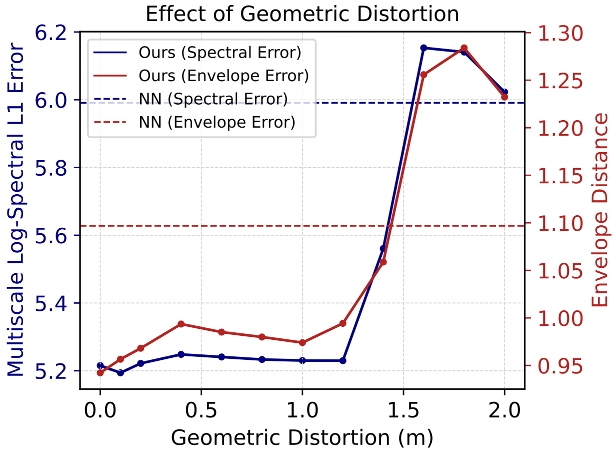


Figure 5. Effect of geometric distortion on RIR prediction performance in the Classroom subdataset The blue line shows our model's performance according to the Multiscale Log-Spectral L1 metric, and the red line shows our model's perfomance according to the envelope distance metric. The red and blue dashed lines indicate the performance of the nearest-neighbors baseline according to the multiscale log-spectral L1 metric and the envelope distance metric, respectively.

D.4. More Ablations

In the Methods section, Section E.4, and Section E.3, we discuss several minor components of our model (axial boosting, time-of-arrival perturbation, hop size 1 loss, etc.) that provide a boost to our model's performance and/or robustness. Results with each of these components ablated are in Table 7. Our model performs the best on a plurality of evaluations, proving that these performance boosts are good on balance. However, we should also observe that even in evaluations where our model does not perform the best, it is never worse than the best performing ablation by a significant margin. We cannot say the same for the Interpolation Spline ablation, which also performs the best in the same number of evaluations (six), but significantly underperforms our model in several settings.

D.5. Modeling the Effects of Transmissions

Our model assumes that sound energy encountering a surface is either reflected or absorbed by the surface. This is for the sake of simplicity. We also conduct an experiment in which we consider surface transmission as well. This means that we modify our tracing algorithm to consider reflection paths that pass through surfaces, and assume that a proportion of the sound energy at each frequency can be *transmitted* through these surfaces in a frequency-dependent manner. Our modified training procedure then fits *surface transmission coefficients* in a manner identical to the way it fits *surface reflection coefficients*. Table 8 contains quantitative results from a model that models both transmission and reflection, and shows that in our settings, modeling transmission is not necessary. However, in other rooms with surfaces of different materials, modeling transmission may be

	Classroom				Dampened Room				Hallway				Complex Room			
	RIR		Music		RIR		Music		RIR		Music		RIR		Music	
	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV
DiffRIR	5.22	**0.942**	2.71	**1.36**	**1.21**	**0.555**	1.59	1.19	**9.13**	2.95	2.59	1.25	**4.86**	0.917	2.25	**1.41**
w/o Time-of-Arrival Perturbation	**5.19**	0.962	**2.70**	1.43	1.23	0.582	1.61	1.36	**9.13**	2.93	2.60	1.27	**4.86**	**0.913**	2.23	1.42
w/o Axial Boosting	**5.19**	0.969	2.71	1.43	1.22	**0.555**	1.59	1.20	9.14	2.95	2.59	1.30	**4.86**	0.934	2.25	1.44
w/o Hop Size 1 Loss	5.26	0.988	2.74	1.41	1.25	0.559	1.67	**1.16**	9.22	2.98	2.60	**1.24**	4.90	0.962	2.27	1.42
w/o Interpolation Spline	5.60	0.973	2.72	1.41	1.63	0.565	**1.53**	**1.16**	9.47	**2.92**	**2.56**	**1.24**	5.24	0.920	**2.21**	1.42

Table 7. Ablation results from the task of predicting monaural RIRs and music at an unseen point. In the Interpolation Spline ablation, the *Residual Component* and the contributions from explicitly computed reflection paths are simply added together, instead of being blended using the learned temporal spline γ. Lower is better for all metrics. Errors for RIRs are multiplied by 10.

	RIR		Music	
Room/Configuration	Mag	ENV	Mag	ENV
Classroom				
DiffRIR (ours)	**5.22**	**0.942**	**2.71**	**1.36**
w/ Transmission	5.23	0.951	2.72	**1.36**
Dampened Room w/ Panel				
DiffRIR (ours)	**1.23**	**0.604**	**1.62**	1.47
w/ Transmission	**1.23**	**0.604**	**1.62**	**1.45**
Hallway w/ Panels				
DiffRIR (ours)	8.39	2.94	2.67	1.35
w/ Transmission	**8.38**	**2.92**	**2.64**	**1.34**
Complex Room				
DiffRIR (ours)	**4.86**	0.917	2.25	1.41
w/ Transmission	**4.86**	**0.915**	**2.24**	**1.38**

Table 8. Evaluations of DiffRIR vs DiffRIR with Transmission modeling. Lower is better for all metrics, and RIR errors are multiplied by 10.

	RIR		Music	
Room/Configuration	Mag	ENV	Mag	ENV
Classroom				
DiffRIR (ours)	**5.22**	**0.942**	**2.71**	**1.36**
Pyroomacoustics	18.64	3.67	3.26	1.68
Dampened Room				
DiffRIR	**1.21**	**0.555**	**1.59**	**1.19**
Pyroomacoustics	2.14	0.798	2.17	1.96
Hallway				
DiffRIR	**9.13**	**2.95**	**2.59**	**1.25**
Pyroomacoustics	32.01	4.03	3.39	1.70

Table 9. Comparison of our model against Pyroomacoustics. Lower is better for all metrics, and RIR errors are multiplied by 10.

important.

D.6. Comparison to Traditional Acoustic Simulations

We compare our method to a widely-used image-source audio simulator, Pyroomacoustics [25]. For each room in our dataset, we simulate RIRs by providing the dimensions and the speaker location, and selecting the closest material coefficients for each surface from its pre-defined database (e.g., drywall, ceiling tiles, carpet). Table 9 reports the accuracy of the simulated RIRs compared to the ground truth.

E. Method Details

E.1. Details on Source Localization

Our method does not require a ground-truth source location measurement. Instead, we use a simple time-of-arrival technique to estimate the sound source's location to a degree of accuracy sufficient for the subsequent steps of the method. For each Room Impulse Response (RIR) in the training set, we determine its first peak, which is proportional to the distance of the direct path between the microphone and source locations. We locate the first peak of the RIR by measuring when the RIR first exceeds a quarter of its absolute maximum. We then determine the distance from the source to the target microphone by multiplying by the speed of sound, assumed to be 343 m/s.

We use a gradient descent optimization method to fit the optimal source location. We initialize the source location to the origin, which is at a corner of the room. We perform an optimization process that updates the optimal source location's position at each step. At each iteration of the optimization process, we compute the estimated times-of-arrival for each of the microphone locations, based on the current estimate for the source location. We then calculate the L1 loss between the estimated times-of-arrival and the times-of-arrival as measured by locating the first peak of the ground-truth RIR as described in the previous paragraph. We perform a gradient update on the estimated source location to minimize this L1 loss. We optimize for 1000 steps, and use the final estimate for the source location as our estimated source location.

In all of the base configurations, our method is able to predict a source location that is inside the location of our QSC loudspeaker. We used the estimated location in all configurations except for the Complex Rotation and Complex Translation configurations, where our localization method failed.

E.2. Minimum-Phase Transform

Our model learns the frequency-domain response curve for each of the surfaces in the room and for each outgoing direction from the source, allowing us to determine how the frequency profile of sound traveling along that reflection path is altered. However, this frequency-domain response is not enough to determine the reflection path's time-domain contribution to the RIR, because it contains magnitude information, but no phase information. In order to invert our reflection path's frequency profile into a time-domain signal, we need to provide phase values at each frequency, so we can perform the inverse-Fourier transform.

In our analysis, we adopt the minimum-phase assumption to calculate phase values for acoustic reflections, a method widely recognized and justified within acoustic research [15, 16]. This assumption posits that for each frequency, the phase delay introduced by the reflection is minimal, implying that the time delay contributed by the path of reflection at any given frequency is as short as possible. From a physical standpoint, this is akin to assuming that sound is reflected off surfaces with negligible delay, thereby behaving as if the reflections are 'instantaneous' while still preserving the unique frequency-dependent characteristics of the reflection. We compute the phase values using the method described in [26].

E.3. Specific Loss Formulation

Loss Formulation and Equations. We define the loss for a given short-time Fourier transform (STFT) window size s_w and hop size h in Equation (6). This is the sum of the L1 distance between the magnitude-spectrograms of the ground-truth and synthesized RIRs and the log-magnitude spectrograms of the ground-truth and synthesized RIRs.

In the formula, W and $\hat{W}$ indicate the ground-truth and predicted RIRs, respectively. h indicates the hop length, s_w indicates the STFT window size, and S is the short-time Fourier transform, or spectrogram, whose arguments are the time-domain signal to transform, the window size, and the hop length, respectively. H indicates the hop ratio, or the hop length divided by the window size. We set $H = 0.25$.

Equation (7) provides the total loss, which sums across multiple window sizes, and adds a loss term that uses a hop size of 1.

Hop Size 1 Loss. We use a spectral loss term with hop size 1 to ensure that the early part of the RIR has accurate time-domain characteristics, since the hop length of 1 allows for high-resolution in the time domain. We take inspiration from [8] for this term, and discover it improves performance, as seen in Table 7.

Modifications from Related Work. Our multi-scale spectral plus log-spectral loss is identical to those used in [9] and [13], with two exceptions: First, is the introduction of the loss term with hop size one. Second, the minimum window size in our loss is 256, instead of 32 or 64. This is because there will be error in the time-of-arrival of certain reflection paths, due to geometric measurement error (which increases with reflection order) or errors in the speed of sound approximation. This means that the placement in time of a reflection path's contribution to our synthesized RIR may be off from its placement in the ground-truth RIR by some amount. Using larger window sizes compensates for this error, since larger windows are more likely to contain both the reflection path's contribution to our synthesized RIR and its contribution to the ground-truth RIR.

E.4. Small Efficiency and Performance Boosts

Efficiency Boosts. Since each rendered RIR combines hundreds of reflection paths, we compute all the reflection path contributions in parallel to minimize runtime. In addition, all reflection paths for the training points are precomputed before training starts.

Time-of-Arrival Perturbation. Since our measurements of each room are not necessarily precise, to make our model more robust, especially with an extremely limited number of measurements, we would like to perturb the surfaces during training. However, reflection paths for all training locations are precomputed before the training process begins. Perturbing each surface would require retracing at each iteration, which is computationally inefficient. As a proxy to this, we perturb the time of arrival of all paths by adding Gaussian noise to it, with a standard deviation of 7 samples. We found that this improved the interpretability of the estimated parameters and led to minor perfomance boosts, as shown in Table 7.

Regularization via Convolution with Pink Noise. Since RIRs are often used as a means to simulate sounds in an acoustic environment, we would like to not only ensure that our rendered RIRs are accurate, but also that the sounds we simulate via convolution with the RIR are accurate. Minimizing the spectral loss between ground-truth and predicted RIRs does not always accomplish this, since convolving the RIRs with other waveforms results in significant changes in the spectrograms.

Pink Noise is a special type of noise whose power spectral density is inversely proportional to frequency. It is ubiquitious in nature [3], and is often used as a test signal to calibrate sound systems and loudspeakers, since its frequency profile is similar to that of music [11] and other sounds the speaker might play.

To encourage our model to maintain accuracy post-

$$L_{s_w,h}(W,\hat{W}) = |S(W,s_w,h) - S(\hat{W},s_w,h)| + |\log S(W,s_w,h) - \log S(\hat{W},s_w,h)| \tag{6}$$

$$L(W,\hat{W}) = \left[\sum_{s_w \in (512,\ldots 4096)} L_{s_w,Hs_w}(W,\hat{W}) \right] + L_{256,1}(W,\hat{W}) \tag{7}$$

convolution, we implement a regularization strategy using pink noise. For the latter half of training iterations, we convolve both our predicted and the ground-truth RIRs with five seconds of randomly generated pink noise, compute the loss between them, and add it to the loss computed between RIRs at each iteration. Convolving RIRs with pink noise simulates the speaker playing of a pink noise test signal. It is equivalent to reshaping the RIR's spectrum according to the profile of pink noise, and applying a random phase shift at each frequency.

Table 10 shows that this form of regularization results in improvements in both RIR prediction and music prediction. Such forms of regularization should be the study of future work and theoretical study.

With the goal of improving rendered music in mind, we also tried a similar form of regularization, where we convolve both our ground-truth and predicted RIRs with five seconds of music randomly sampled from the FMA dataset [12] at each iteration after training is halfway done. Convolution with the music files simulates the speaker playing them. Results for this form of regularization are also shown in Table 10, although we prefer the performance and simplicity of pink noise regularization.

E.5. Computational Cost

Training and Path-Tracing Time. In all of our experiments, we trained our model for 1000 epochs. In Table 11, we report the amount of time it took for our model to train on each of the base room configurations. Note that since the Complex Room is only traced up to order 4, there are substantially fewer valid reflection paths, and thus training is faster. In all other rooms, we trace up to order 5. Tracing is slower in rooms with more surfaces.

Main Contributions to Training Time. We also measured the different steps in the training process to see which ones took the longest. Each training location is associated with hundreds of reflection paths that must be added together to form the the RIR. While rendering these contributions is done in parallel, compiling them requires placing them in at the right locations in time and is done sequentially. In practice, 37.7% of the time during the 1000 epochs is spent on this compilation, 61.9% on the backwards passes, and 0.4% on everything else.

F. Baseline Implementation Details

Linear. The Linear baseline computes a RIR at a given test location by taking a linear combination of the four nearset points in the training data. The weights on each of these four training points are inversely proportional to the distance to the test location. We also experimented with taking a weighted combination of all the training data, where the weights are inversely proportional to distance. This alternative linear baseline performs quite poorly, with error increasing with the number of training points. This is because the training RIRs are roughly uncorrelated with mean zero, so the average of N RIRs tends towards zero as N increases.

Neural Acoustic Fields (NAF) [19]. To compare our method to NAF, we utilized NAF's official code,[1] as open-sourced by authors. However, in order to apply NAF to our dataset and experimental settings, we modified this code in some minor ways. Specifically, the original NAF was designed to estimate arbitrary stereo RIRs constrained to lie on a 2D horizontal plane within a 3D room, i.e., it did not consider a z-axis and thus does not output RIRs at arbitrary heights. Therefore, we added the height on the z-axis as an input, embedding it by using the same positional encoding [21, 23] as the authors' code. The corresponding elements of the network architecture, e.g., the number of units in the input layer, were also modified. The architecture we used for NAF in our experiments consisted of 8 linear layers with Leaky-ReLU activations [20]. Note that we only changed the number of the number of units in the input layer, from 126 to 168, due to the aforementioned addition of z-axis features. In addition, the NAF we used in our experiments was designed to output only magnitude-spectrograms, i.e., without any phase information, because the official code also does not have the phase-related loss and corresponding phase output. We utilized the Griffin-Lim algorithm [14] to estimate the phase of each magnitude spectrogram and render the time-domain RIRs. For training, we followed the same process in their official code and used the model's weights after the final training epoch for inference and evaluation. Finally, we used a 48000 Hz sample rate rather than the original 22050 Hz. All other settings, such as the optimizer, number of epochs, learning rate, etc., are the same as their official implementation.

[1] https://github.com/aluo-x/Learning_Neural_Acoustic_Fields

	Classroom				Dampened Room				Hallway				Complex Room			
	RIR		Music		RIR		Music		RIR		Music		RIR		Music	
	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV	Mag	ENV
Pink Reg	5.22	**0.942**	**2.71**	**1.36**	**1.21**	**0.555**	**1.59**	**1.19**	**9.13**	**2.95**	**2.59**	**1.25**	4.86	0.917	**2.25**	**1.41**
No Reg.	5.22	0.973	2.76	1.47	1.23	0.579	1.62	1.33	9.17	2.99	2.71	1.35	**4.84**	0.908	2.26	1.45
Music Reg.	**5.20**	0.952	2.72	1.40	1.22	0.569	1.62	1.27	9.14	2.96	2.65	1.31	4.84	**0.903**	2.25	1.42

Table 10. Comparison of our model trained with no regularization, regularizing by convolving with pink noise, and regularization by convolving with music, on the tasks of monoaural RIR and music prediction. Lower is better for all metrics, and RIR errors are multiplied by 10.

Room	Training Time (Hours)	Inference Time (s)	N. Surfaces	Tracing Time (s)	Avg N. Reflection Paths
Classroom	9.61	0.90	9	4.3	874
Dampened	5.75	0.56	6	0.83	675
Hallway	8.97	0.90	6	1.5	853
Complex	2.82	0.37	33	47	439

Table 11. In all of our experiments, we train our model for 1000 epochs and report the training time that this takes in each room, in the base configuration. In addition, we report the inference time, or the time it takes our model to render a single RIR. Before training begins, we precompute the reflection paths that go between the source and listener locations, up to a certain maximum reflection order, so we also report this tracing time to trace reflection paths, per listener location of each room and its corresponding subdataset. We also report the number of valid reflection paths found by the tracing algorithm, as an average across all points in the subdataset.

Deep Impulse Responses (DeepIR) [24]. Unlike NAF, the authors of DeepIR have not open-sourced an official codebase. Therefore, we implemented DeepIR ourselves, based on the details in their paper. Specifically, we built a simple multi-layer perceptron (MLP) consisting of 6 linear layers, each followed by leaky-ReLU activations. The input feature vector consists of (x, y, z, t), which are the desired spatial coordinates and the time index, respectively. Similar to NAF, we applied positional encoding to all inputs before passing them into the MLP. Hence, the number of units in the input layer is d_{emb}, whereas all other layers have 512 units. DeepIR directly outputs the t^{th} time sample of the RIR to produce an estimate $\hat{\text{IR}}$ of the full RIR. We then convolve this with the arbitrary dry source audio x, to produce an estimate $\hat{y}$ of the sound of the arbitrary audio being recorded from the specified source and listener location in the room, i.e., $\hat{y} = x * \hat{\text{IR}}$. We optimized $\hat{y}$ according to an L2 loss comparing the log-magnitude spectrogram with that of the corresponding ground-truth audio y_{gt}. We omitted the noise model, since our dataset did not include artificially added noise, and the noise in our recordings was minimal. We set other hyperparameters for DeepIR such as the optimizer, learning rate, the number of epochs, etc., to the same values as NAF.

Implicit Neural Representation for Audio Scenes (INRAS) [27]. The authors of the INRAS baseline released their code in the Supplementary Materials of their submission.[2] We use their code with some minor modifications. The framework is originally trained and tested on data from the SoundSpaces dataset [6], which provides simulated binaural recordings within virtual environments. The architecture is built around consuming this data, where each simulated recording represents a stereo, binaural recording with the head positioned at one of the four cardinal angles. Our training sets use exclusively monaural recordings from omnidirectional microphones. Thus, in order to make our changes to the network as minimal as possible, we duplicated our mono-channel recordings to stereo-channel recordings and assumed them to all be at the 0° angle. We then took only the left channel of the stereo output as the framework's estimate of the monaural RIR. Since INRAS consumes environment meshes, we provide it with a 3D scan of each room. Otherwise, we used mostly the same hyperparameters as the original, with the exception that we increased the sample rate from 22050 to 48000 Hz. Since our training set of 12 recordings per subdataset was approximately four orders of magnitude smaller than the datasets on which the authors had trained, we increased the initial learning rate from to 0.001 instead of 0.0005, slowed the learning rate's exponential decay schedule to decay rate $\gamma = 0.1$ over 3000 epochs rather than 50, and trained for 5000 epochs rather than 100. We evaluated the model against a validation set every 100 epochs. For our test evaluations of INRAS, we used the weights and consequent outputs of the model with the best performance across all such validation

[2] https://openreview.net/forum?id=7KBzV5IL7W

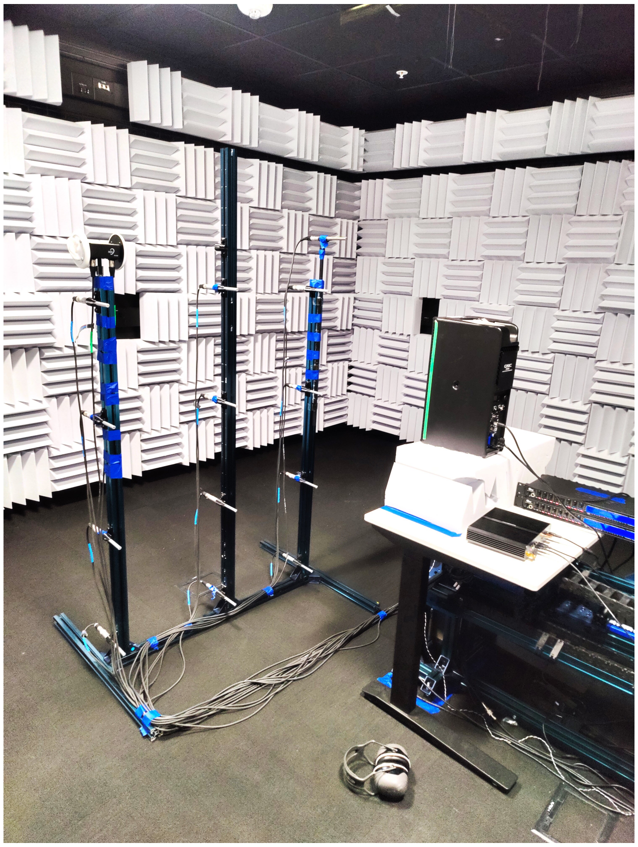


Figure 6. A photo of the data collection procedure in the Dampened Room. The custom microphone frame holds 12 microphones, as well as a 3Dio FS XLR binaural microphone.

evaluations.

G. Data Collection Procedure Details

We use a custom-built microphone frame designed to accommodate 12 Dayton Audio EMM6 measurement microphones, as well as one 3Dio FS XLR binaural microphone, all of which were rigidly mounted at precisely measured positions on the frame. Figure 6 shows a photo of the microphone frame used to collect the data. We set the origin of each room such that there is one wall representing $x = 0$ and one wall representing $y = 0$. Before each recording, we positioned the frame within the room and measured the distance from the edge of the frame to each of the origin walls using a tape measure or a Bosch GLM20 laser distance measure, which have 1 and 3 millimeter measurement resolutions, respectively. We use the measured position of the frame's corner as well as the pre-measured offset of each microphone from the frame's corner in order to annotate each microphone's position in the room to sub-centimeter precision for our dataset.

G.1. Estimating the Room Impulse Response (RIR)

In order to measure each RIR, we played a logarithmic sine sweep through the speaker. The sweep spanned from 20 Hz to 24 kHz for 10 seconds, followed by 4 seconds of silence. This sine sweep was recorded from each of the microphones simultaneously at each gantry position. While sending the sine sweep signal from the audio interface to the speaker, we also recorded loopback signal by wiring the audio interface's output to one of its inputs. We used this loopback signal to estimate and correct for the latency in the system.

To compute the RIR $r[t]$, we take

$$r[t] = IFFT\left(\frac{FFT(a[t])}{FFT(l[t])}\right),$$

where FFT and $IFFT$ are the Fast-Fourier Transform and its inverse respectively, $a[t]$ is the digital recording of the sine sweep, and $l[t]$ is the digital loopback signal. Note that we deconvolve the loopback signal from the recording, instead of deconvolving the source signal sent to the speaker from the recording. We assume that the loopback signal is the same as the source signal, but delayed in time by the latency of the system. Deconvolving from a delayed copy of the source signal instead of directly from the source signal thus corrects for the delay in the system. We remove the last 0.1 seconds of the 14-second RIR to eliminate anti-causal artifacts.

In addition, to account for differences in microphone sensitivity, we adjust the volume of each sweep recording according to the sensitivity of the microphone used to record it. Specifically, we look up each EMM6's microphone's response at 1000 Hz in dB from its calibration sheet, and reduce the overall volume of its recordings by the same amount.

G.2. Room Geometry Estimation

As the wavelengths of audible sound typically range from 2 cm - 17 m [22], the prominent sound waves are likely to bypass or diffract around smaller surfaces. Hence, we only focus on modeling salient surfaces (e.g., walls, pillars, table tops), which are often characterized by planes, and simply trace the reflection paths using image source methods. For the rooms we captured in our dataset, we also measured the walls and surfaces and manually created planar mesh-based reconstructions of them. With the recent progress in visual 3D scene reconstructions [21], our geometric estimation can also easily be replaced by automatic algorithms or even mature customer tools such as Polycam.

H. Guidelines for Microphone Placement.

To maximize efficiency, we found it empirically beneficial to spread our RIR locations in all three dimensions. This allows us to 1) cover a variety of angles around the speaker,

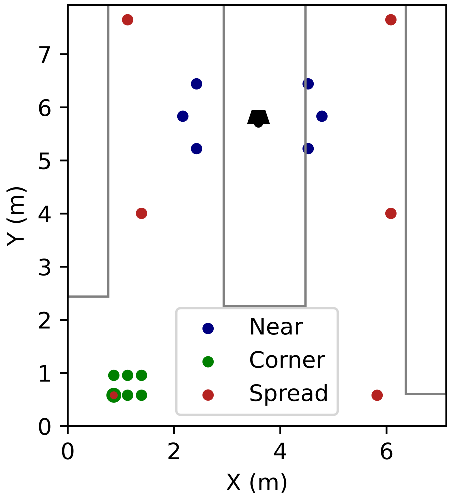


Figure 7. The distributions of three different sets of training points in the Classroom subdataset. The grey lines indicate the locations of tables in the subdataset.

	RIR		Music	
Training Point Configuration	Mag	ENV	Mag	ENV
Near	5.89	1.14	3.25	1.61
Spread	**5.39**	**0.976**	**2.80**	**1.36**
Corner	5.88	1.07	3.12	1.41

Table 12. Evaluations of DIFFRIR on different datasets of size 6, with varying spatial distributions. All microphone locations are selected from $Z = 0.98$, and all locations used for testing and evaluation are also selected from $Z = 0.98$. Lower is better for all metrics, and RIR errors are multiplied by 10.

which likely leads to better speaker directivity estimates, 2) disentangle the effects of individual reflections, and 3) better estimate the diffuse sound field, which is approximated in our model as spatially uniform.

To study this effect, we conducted an experiment in the Classroom subdataset. We select three different sets of training locations (shown in Figure 7), each of which contain 6 RIR recordings from 6 different locations. For simplicity, these training locations were selected in the 2D plane defined by $Z = 0.98$. We evaluated DIFFRIR trained on each of these sets of training locations on a test set comprised of other points selected in the $Z = 0.98$ plane.

Our best performance across all metrics is achieved in the 'Spread' configuration of training points, confirming our intuition. Interestingly enough, the 'Near' Configuration performed the worst. We believe this could be due to the model overfitting to the near-field of the speaker [17], which can be substantially different than the sound field at other locations in the room.

References

[1] Thibaut Ajdler, Luciano Sbaiz, and Martin Vetterli. The plenacoustic function and its sampling. *IEEE TIP*, 54(10): 3790–3804, 2006. 2

[2] Cal Armstrong, Lewis Thresh, Damian Murphy, and Gavin Kearney. A perceptual evaluation of individual and non-individual hrtfs: A case study of the sadie ii database. *Applied Sciences*, 8(11):2029, 2018. 6

[3] Per Bak, Chao Tang, and Kurt Wiesenfeld. An explanation of 1/f noise. *Physical Review Letters*, 59:381–384, 1987. 9

[4] S. Butterworth. On the Theory of Filter Amplifiers. *Experimental Wireless & the Wireless Engineer*, 7:536–541, 1930. 3

[5] Diego Caviedes-Nozal, Nicolai A.B. Riis, Franz M. Heuchel, Jonas Brunskog, Peter Gerstoft, and Efren Fernandez-Grande. Gaussian processes for sound field reconstruction. *Journal of the Acoustical Society of America*, 149(2):1107–1119, 2021. Funding Information: The authors would like to thank Manuel Hahmann for the fruitful discussions. This work is part of the MONICA project and has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 732350. It is partly supported by the VILLUM foundation (Grant No. 19179, "Large scale acoustic holography"). Publisher Copyright: © 2021 Acoustical Society of America. 2

[6] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020. 11

[7] Changan Chen, Alexander Richard, Roman Shapovalov, Vamsi Krishna Ithapu, Natalia Neverova, Kristen Grauman, and Andrea Vedaldi. Novel-view acoustic synthesis. In *CVPR*, pages 6409–6419, 2023. 6, 7

[8] Ziyang Chen, David F Fouhey, and Andrew Owens. Sound localization by self-supervised time delay estimation. *European Conference on Computer Vision (ECCV)*, 2022. 9

[9] Samuel Clarke, Negin Heravi, Mark Rau, Ruohan Gao, Jiajun Wu, Doug James, and Jeannette Bohg. Diffimpact: Differentiable rendering and identification of impact sounds. In *Conference on Robot Learning*, pages 662–673. PMLR, 2022. 9

[10] Orchisama Das, Paul Calamia, and Sebastia V. Amengual Gari. Room impulse response interpolation from a sparse set of measurements using a modal architecture. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 960–964, 2021. 2

[11] Gary Davis and Ralph Jones. *The Sound Reinforcement Handbook*. Hal Leonard, 1987. 9

[12] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016. 10

[13] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*, 2020. 9

[14] D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984. 10

[15] Sahar Hashemgeloogerdi and Mark Bocko. Invertibility of acoustic systems: An intuitive physics-based model of minimum phase behavior. page 055002, 2015. 9

[16] Jun-Hyeok Heo, Deok-Ki Kim, and Byoung-Duk Lim. Application of minimum phase condition to the acoustic reflection coefficient measurement. *Transactions of the Korean Society for Noise and Vibration Engineering*, 15, 2005. 9

[17] M. S. Howe. *Introduction*, page 1–24. Cambridge University Press, 2002. 13

[18] Ole Kirkeby and Philip A. Nelson. Reproduction of plane wave sound fields. *The Journal of the Acoustical Society of America*, 94(5):2992–3000, 1993. 2

[19] 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. 1, 10

[20] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, page 3. Atlanta, GA, 2013. 10

[21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 10, 12

[22] Henrik Møller and Christian Sejer Pedersen. Hearing at low and infrasonic frequencies. *Noise & health*, 6(23):37–57, 2004. 12

[23] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 10

[24] Alexander Richard, Peter Dodds, and Vamsi Krishna Ithapu. Deep impulse responses: Estimating and parameterizing filters with deep networks. In *ICASSP*, pages 3209–3213. IEEE, 2022. 11

[25] Robin Scheibler, Eric Bezzam, and Ivan Dokmanic. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018. 8

[26] Julius O. Smith. *Spectral Audio Signal Processing*. https://ccrma.stanford.edu/~jos/sasp/, accessed ¡date¿. online book, 2011 edition. 9

[27] Kun Su, Mingfei Chen, and Eli Shlizerman. INRAS: Implicit neural representation for audio scenes. In *Advances in Neural Information Processing Systems*, 2022. 1, 11