Supplementary

A Additional Visualization of Rooms

Please see https://sites.google.com/view/nafs-neurips2022 for videos of loudness plots as we move an emitter.

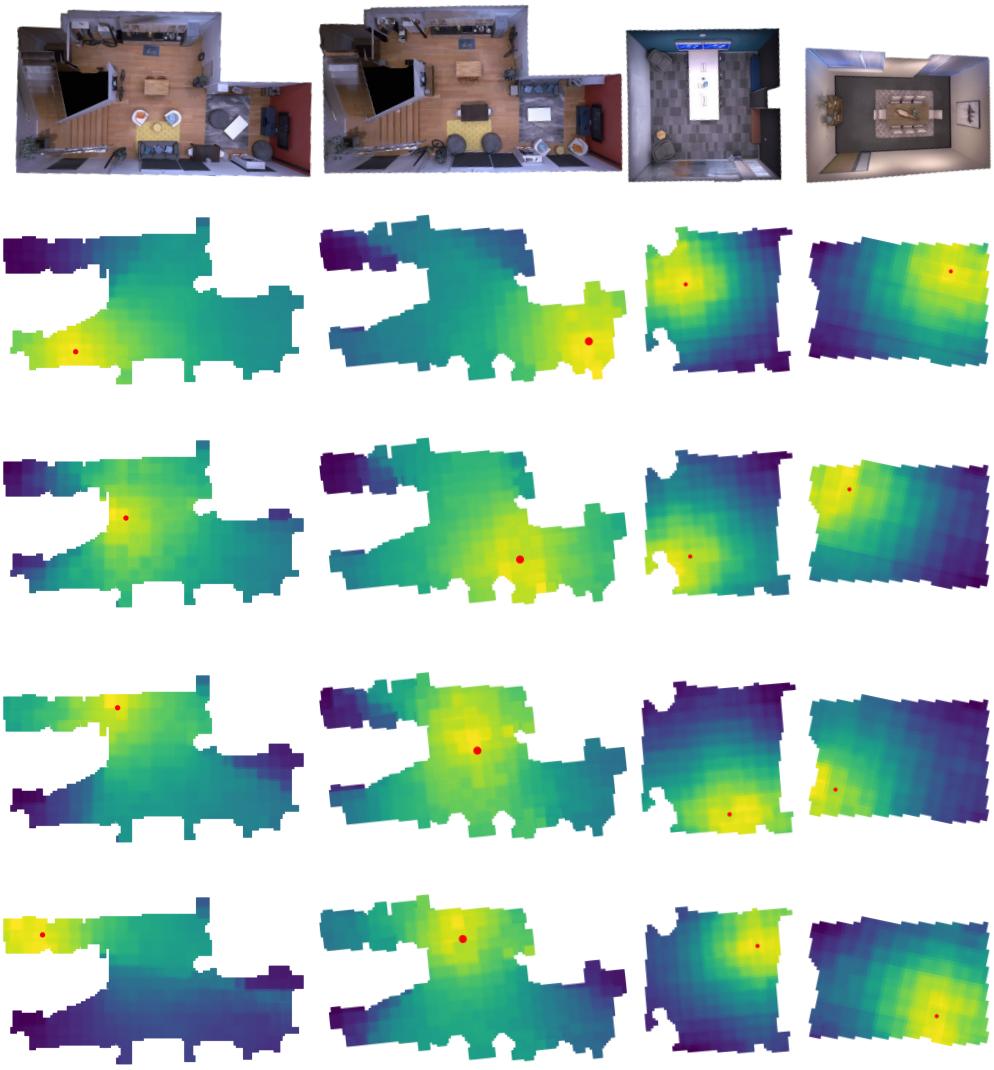

Figure A1: **Additional Qualitative Predictions of NAF.** Qualitative visualization of the loudness map as predicted by NAF across four different rooms.

We show additional NAF predictions of loudness as we move an emitter inside different rooms in Figure A1. For each room, note how the sound is affected by the geometry. In wide open spaces the sound is highly dispersed. While in thin structures the sound tends to concentrate locally. As we move farther from the source, the loudness of the sound decreases.

B Additional Visualization on Real-World Data

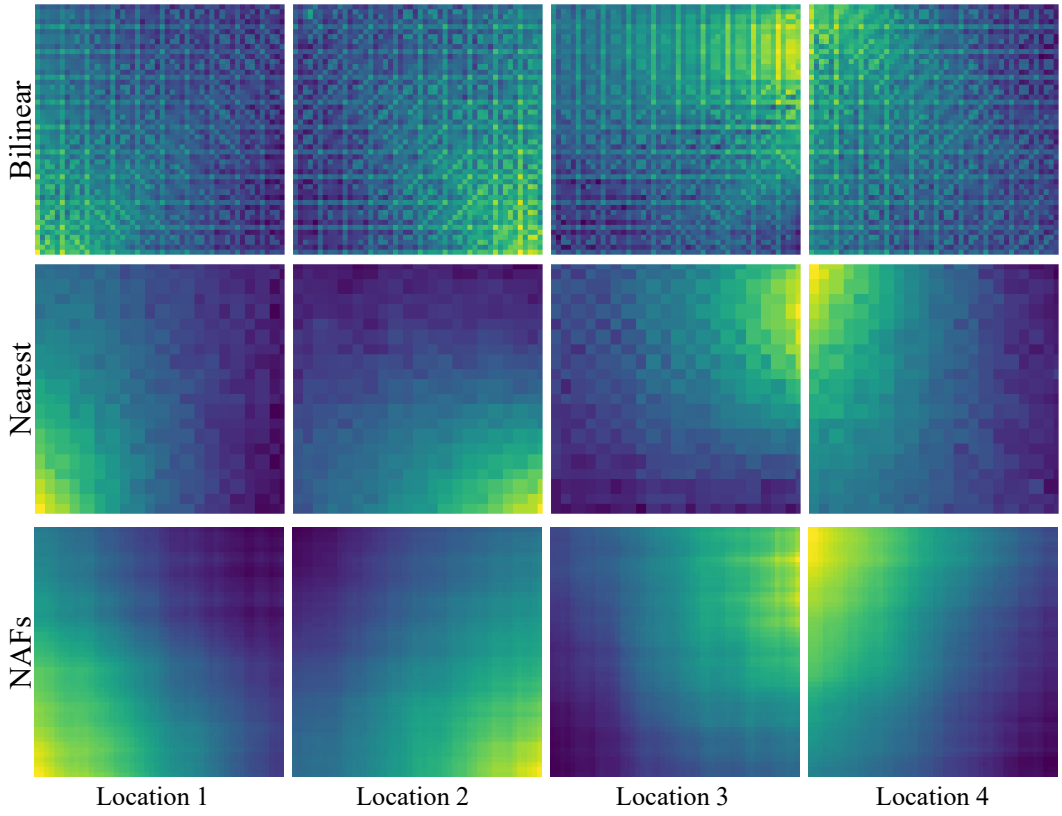

Figure A2: **Comparison on the MeshRIR real-world dataset.** We compare our method on the MeshRIR dataset across four emitter locations. **Top.** The loudness map using bilinear interpolation of the ground truth. **Middle.** The loudness map using nearest interpolation of the ground truth. **Bottom.** The loudness map predicted using NAFs. Our method can predict a smoothly varying loudness map without artifacts.

In Figure A2 we compare on the MeshRIR dataset which is collected from the real-world. Bilinear interpolation introduces characteristic artifacts at the sample boundaries, while nearest neighbor has discretization artifacts. In contrast, our NAFs are able to predict a smoothly varying acoustic field despite learning from discretely sampled training data.

C Additional quantitative results

	Spectral	T60	DRR
Ridge-Orig	2.539	8.192	2.497
Ridge-Unfiltered	1.370	6.294	3.702
NAF (Dual)	**0.403**	4.201	0.992
NAF (Shared)	**0.403**	**4.191**	**0.972**

Table A1: **Comparison against a kernel regression baseline** We compare against a kernel ridge regression baseline on the MeshRIR dataset. We find that our NAFs perform better on the metrics evaluated.

In Table A1, we compare our method against "Kernel Ridge Regression with Constraint of Helmholtz Equation for Sound Field Interpolation" on the MeshRIR dataset. "Ridge-Orig" denotes the authors proposed setup which applies a 500Hz low pass filter. While "Ridge-Unfiltered" is a modified setup where we do not perform a low pass. Note that their method requires an individual model for each unique emitter location, while our NAFs can be queried using any emitter/receiver position.

	DRR error ↓							
Method	Large 1	Large 2	Medium 1	Medium 2	Small 1	Small 2	MeshRIR	Mean
AAC-nearest	1.748	2.424	1.344	1.343	1.213	1.108	1.286	1.495
AAC-linear	1.797	2.147	1.457	1.458	1.117	1.226	1.222	1.490
Opus-nearest	2.931	3.275	2.756	2.769	3.548	3.255	2.698	3.033
Opus-linear	2.645	2.771	2.381	2.370	3.266	2.882	2.529	2.692
DSP	3.559	4.421	4.727	4.805	5.622	6.723	-	4.976
NAF (Dual)	1.645	1.830	1.113	**1.082**	**0.796**	**0.799**	0.992	1.179
NAF (Shared)	**1.468**	**1.793**	**1.083**	1.089	0.829	0.837	**0.972**	**1.153**

Table A2: **Mean absolute error of DRR.** We compute the direct-to-reverberant ratio (DRR). Here we show the mean absolute error of the DRR. Units are dB, left/right channel is processed independently.

In Table A2 we evaluate the error in the direct-to-reverberant ratio for the impulse response in each method. The direct-to-reverberant measures the ratio of energy between the direct and reverberant component of an impulse response. We find that NAFs have lower DRR error than baseline methods.

	IACC error ↓						
Method	Large 1	Large 2	Medium 1	Medium 2	Small 1	Small 2	Mean
AAC-nearest	236.8	184.2	213.7	215.3	264.8	272.5	231.2
AAC-linear	212.3	156.7	185.9	187.8	245.2	265.2	208.8
Opus-nearest	73.75	45.97	71.97	74.70	103.8	**67.40**	72.93
Opus-linear	75.56	48.32	73.38	77.33	109.2	78.10	76.98
DSP	460.5	446.0	430.0	430.1	443.6	446.3	442.7
NAF (Dual)	74.01	45.94	71.89	74.70	103.8	**67.40**	72.96
NAF (Shared)	**73.68**	**45.90**	**71.52**	**73.58**	**103.6**	**67.40**	**72.62**

Table A3: **Mean absolute error of IACC.** We compute interaural cross correlation coefficient (IACC) using the impulse response from the left and right ears. Here we show the mean absolute error of the IACC for a given method and the ground truth. Units are seconds, for visualization values are multiplied by 1e6, lower is better.

In Table A3 we evaluate the error in the interaural cross correlation coefficient (IACC). The IACC is correlated with the ability for humans to localize a sound. We find that NAFs have low IACC error.

D Architecture and Training Details

We visualize all three models that we experiment with.

In Figure A3 is a network that uses different local feature grids for the emitter and receiver (dual grids). The network uses the emitter and listener positions to sample from the two different grids.

In Figure A4 we show a model where the local feature grids for the emitter and receiver are shared. This network uses the emitter and listener positions to sample from the same shared grid.

In Figure A5 we show a model that does not utilize any kind of local geometry conditioning.

The listener, emitter, phase, and time input are transformed using sinusoidal embedding, while the orientation and left/right are retrieved. All transformed inputs are directly fed to the network. We find that the sharing the feature grid performs better than using different local feature grids.

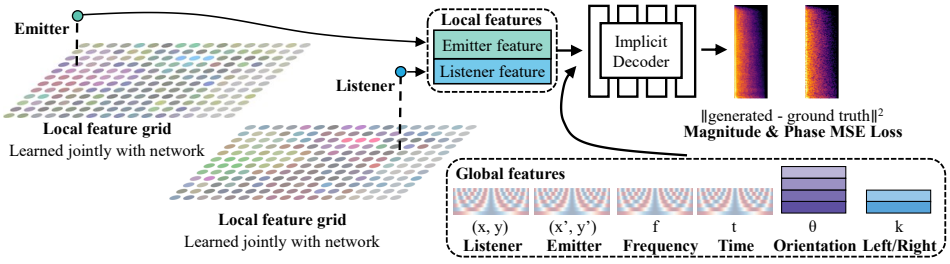

Figure A3: Architecture of the model that uses emitter and listener specific local geometry conditioning.

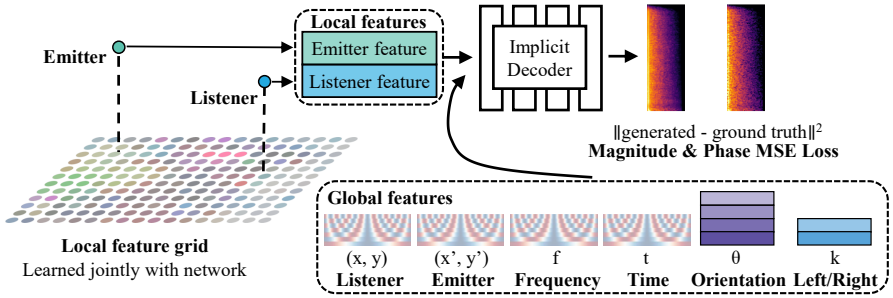

Figure A4: Architecture of the model that share emitter and listener local geometry conditioning.

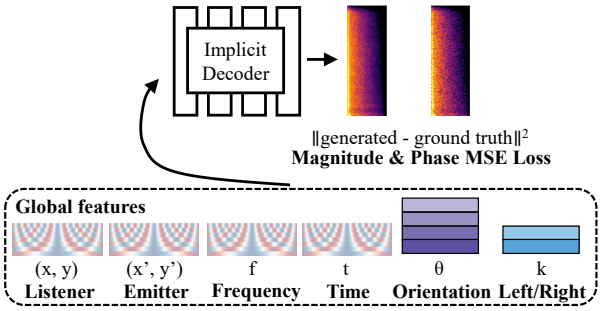

Figure A5: Architecture of the model that uses no local geometry conditioning.

Each network consists of 8 fully connected layers in a feedforward fashion, as well as a skip connection consisting of two fully connected layers. The skip connection takes the input and adds its output to that for the fourth intermediate layer. We utilize an intermediate feature size of 512, and Leaky ReLU with a slope of 0.1 as the activation function. The grid is initialized to stretch the bounding box of a scene. Each point is located at a distance of 0.25m from the nearest neighbor. 64 features are used for each point. Each element of the grid is initialized i.i.d. from $\mathcal{N}(0, \frac{1}{\sqrt{64}})$. We initialize the bandwidth for each point at $\sigma = 0.25$, and jointly train the bandwidth as part of the network. For the network and the grid, we utilize an initial learning rate of $5e-4$. The

Adam optimizer is used when training our network. We utilize a orientation embedding of shape $\mathcal{R}^{7\times4\times512}$ where 7 is the number of intermediate outputs, 4 is the number of orientations, and 512 is the feature dimension. For the left-right embedding, we use a shape of $\mathcal{R}^{7\times2\times512}$. We perform additive conditioning by adding a $\mathcal{R}^{512}$ vector to each intermediate output for both the orientation and the left/right.

For each scene, to generate a log-spectrogram for each impulse response, we compute the mean and standard deviation $\mu_{(t,f)}, \sigma_{(t,f)}$ for each time/frequency index in the log-spectrogram, and normalize the data prior to training:

$$\boldsymbol{v}_{\text{STFT\_mag}}(t,f) = \frac{\boldsymbol{v}_{\text{STFT\_mag}}(t,f) - \mu_{(t,f)}}{3.0 \times \sigma_{(t,f)}}$$

To generate the instantaneous frequency (phase) representation for each impulse response, we normalize the data prior to training:

$$\boldsymbol{v}_{\text{STFT\_IF}} = \frac{\boldsymbol{v}_{\text{STFT\_IF}}}{3.0 \times \sigma_{\text{IF}}}$$

For the sinusoidal embedding, we utilize both $\cos$ and $\sin$ with 10 frequencies each for encoding position, phase, and time. For encoding position we utilize a max frequency of 2^7Hz, while for encoding time and frequency we utilize a max frequency of 2^{10}Hz.

Since we do not know beforehand the time duration of an impulse response at an unseen location, we compute the maximum impulse length for each scene and use this length to zero pad the training impulse responses. Because the padded regions do not contain useful information, we want the network to focus modeling efforts on the early regions of the impulse response. We achieve this by stochastically padding the impulse response to maximum impulse length with 0.1 probability. Because the implicit function is trained on individual (t, f) coordinates within a given v_{STFT}, training samples do not need to be of the same length. During test time, we perform inference up to the maximum duration of scene impulse response.

E Dataset Visualization

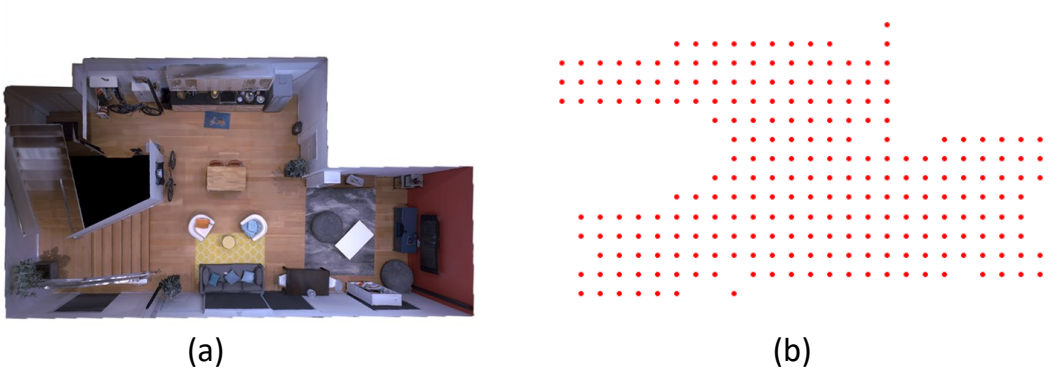

Figure A6: A room the emitter-listener probes. **(a)** The 3D structure of a room. **b** The probes marking the location of emitters/listeners.

In Figure A6, we visualize both the room and underlying set of probe positions in the training data. Due to occlusion and the geometry, even slightly moving the emitter or listener position can result in different results. As we demonstrated, both nearest neighbor and linear interpolation perform poorly compared to our learned solution. In contrast, recovered acoustic fields from NAF trained on these probe positions is substantially denser (Figure A1).

F Storage Comparison

We compare the averaged on disk storage cost of the different methods for inferring the spatial audio using a precomputed training set in Table A4. Both linear and nearest interpolation methods require

	Storage (MiB)						
Method	Large 1	Large 2	Medium 1	Medium 2	Small 1	Small 2	Mean
AAC	495.97	478.55	483.42	451.14	116.75	54.64	346.74
Opus	258.51	257.08	245.65	231.06	66.15	29.75	181.37
NAF (Dual)	8.78	8.87	8.87	8.92	8.45	8.37	8.71
NAF (Shared)	8.44	8.49	8.49	8.51	8.28	8.23	8.41

Table A4: **Storage cost of different methods.** We average the amount of data required for different methods of inference for the six scenes. Our NAFs are able to compactly represent the scene while maintaining higher quality.

access to the entire training set, while our NAF based approaches compactly encode the acoustic scene.

G Details of the compression baselines

If uncompressed, the precomputed spatial acoustic field can reach gigabyte or terabyte sizes depending on probe density, scene size, and bandwidth of the impulse. When applied to gaming and virtual reality applications, minimizing the space taken up by these acoustic representations is critical and have been widely studied.

We utilize two state-of-the-art lossy coding methods applied to the audio. They are respectively Advanced Audio Coding (AAC-LC) and Xiph Opus. These two methods were chosen because they are in widespread usage for media encoding, are among the best coding methods for a given bitrate, and have high quality open-source implementations available. The bitrates were selected on the basis of attempting to match the size of the NAFs representations, while being allowed by the respective encoders.

We describe the parameters and additional details for these two coding methods.

G.1 AAC baseline

We utilize `ffmpeg 5.0`, and select the open source "aac" implementation. We set the combined stereo bitrate to 24 kBit/s (12kBit/s per channel) in constant bit rate mode, as we found that there are occasional encode/decode failures below this bitrate.

G.2 Opus baseline

We utilize `opustools 0.2` backed by `libopus 1.3.1`. The encoder is set to 12kBit/s for stereo (6kBit/s per channel) in constrained variable bitrate mode. Complexity it set to the maximum of 10, and music mode is set (as opposed to speech tuning mode).

H Alternative Neural Representations

Representation	Spectral loss ↓	T60↓
Time domain	2.046	49.72
NAFs	0.396	4.166

Table A5: **Learning different representations** We compare NAFs in the STFT domain against directly learning in the time domain.

Our current method follows prior work in learning the log-magnitude STFT and instantaneous frequency phase. In this section, we investigate a possible alternative of directly learning in the time domain. The MSE and T60 error percentage is presented in Table A5. We observe that modeling in the time domain performs poorly.

	Large 1		Large 2	
	PSNR ↑	MSE ↓	PSNR ↑	MSE ↓
NeRF + grid + L_2	22.69	6.956	24.86	7.128
NeRF + grid	**25.41**	**6.618**	**25.70**	**6.921**

Table A6: **Regularizing the grid.** In this experiment, we compare learning NeRF with a grid without regularization, and with L_2 regularization.

I L_2 regularized grid in NeRF

In Table A6 we compare NeRF that utilizes a grid and trained using image reconstruction loss, against a variant where a L_2 penalty with weight $1e-5$ to ensure a smooth latent space is added to the image reconstruction loss. There are 75 images used in the training set. We observe degraded performance when we apply this penalty. This indicates that our NAFs are providing more information than simple regularization to ensure a smooth latent grid.

J Societal Impact

Our work focuses on learning a high quality representation of acoustic fields. The primary use case for our work lies in virtual reality and gaming. As our work can lead to more believable and higher quality representations of spatial audio than alternative methods, it is possible that our work could increase the dependency and time spent on gaming. The more compact nature of our acoustic representations may allow for spatial audio to be deployed to more systems, and enable more equitable access.