Supplementary Material
INRAS: Implicit Neural Representation for Audio Scenes

1 Generated Samples and Overview Video

Please see the **attached video for a short overview and samples of video clips that were sounded with INRAS.** Please turn **Audio ON** and use **headphones** for best perception of the audio. Due to the 100MB size limit of supplementary material, we posted additional video samples on **figshare** (link).

2 Robustness: Training on Various Amounts of Data

We investigate whether INRAS is robust to be trained with limited percentage of the training data. We ran experiments on the room of "frl apartment 4" and used 5%, 10%, 20%, 40%, and 60% of training data. The quantitative results are shown in Table 1. Our results show that INRAS can achieve reasonable performance with only 10% of the training data.

Training Data \Metric	C50 err (dB) ↓	T60 err (%) ↓	EDT err (sec) ↓
5%	1.32	4.25	0.046
10%	0.72	2.64	0.025
20%	0.56	2.38	0.020
40%	0.53	2.24	0.018
60%	0.51	2.17	0.018
100%	0.44	2.07	0.017

Table 1: Quantitative Results using Different amount of Training data.

3 Additional Implementation Details

3.1 Architecture Details

The detailed configuration of INRAS is shown in Fig. 1.

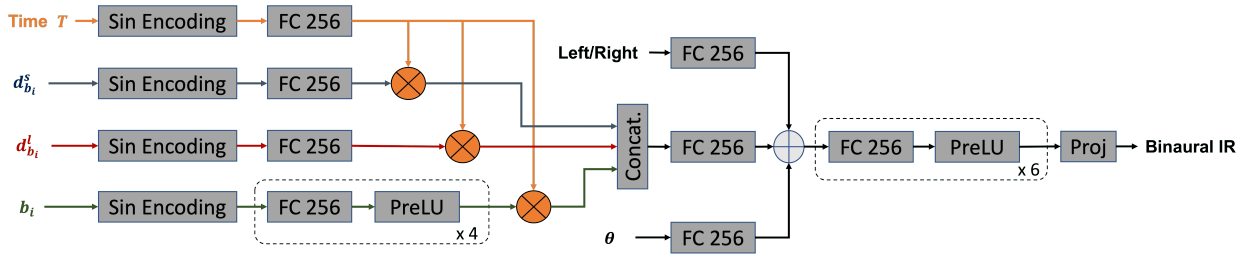

Figure 1: Detailed architecture of INRAS. 'FC' refers to a fully connected layer.

3.2 Emitter/Listener Locations and Bounce Points Visualization

We visualize available discrete locations for emitter/listener (blue) in the dataset and bounce points (red) used in our approach (see Fig. 2). To obtain bounce points, we extract the boundary from the mesh coordinates and sample points uniformly. We find that it is sufficient to cover the whole scene with 40 to 60 bounce points. We observe that in the datasets the emitter/listener locations are not dense. Therefore the ground truth nearest neighbor or interpolation approach could be problematic for continuous moving. In comparison, INRAS learns a continuous field as we showed in Fig. 4 & 5 in the main paper.

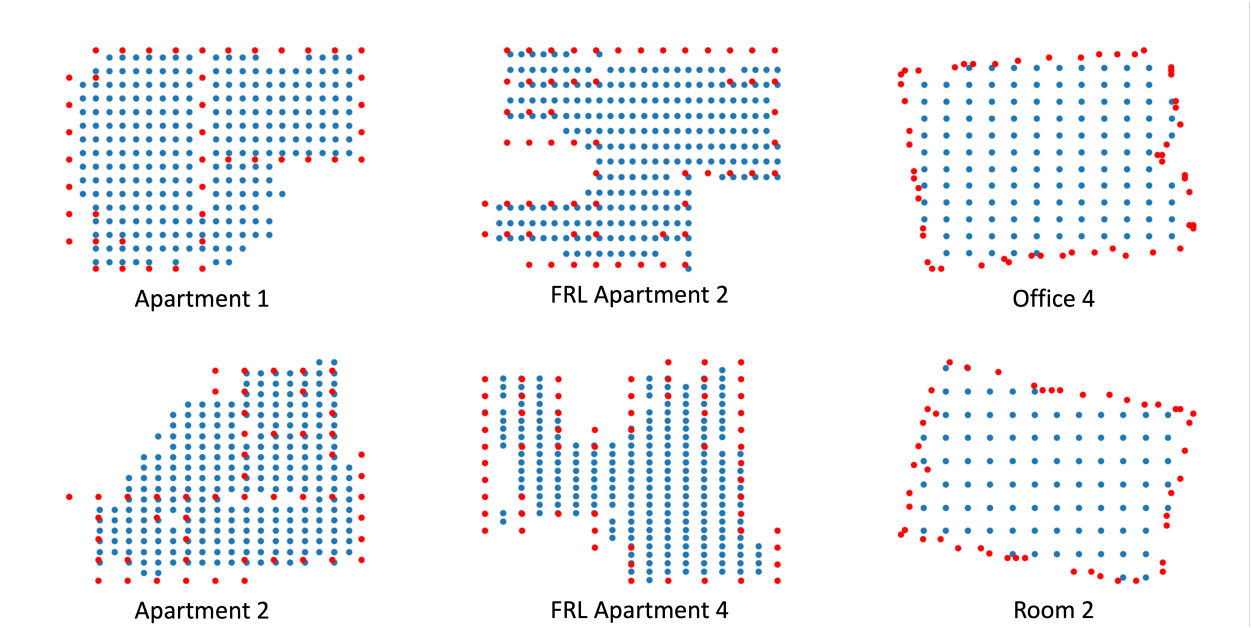

Figure 2: Blue dots indicate the available emitter/listener locations in the dataset. Red dots are the bounce points used for training.

3.3 Code

We provide our implementation code based on PyTorch in the **code** folder. We plan to release the full code: training code, data and checkpoints on GitHub.

4 Limitations

One limitation of INRAS is that the boundary of the scene should be given. While it is usually not a problem for scenes with 3D models, it could be an issue for scenes with unknown geometry and recorded impulse responses. Another limitation is that INRAS requires a sufficient amount of training data to learn a reasonable acoustic field for the scene. This would be not a problem for virtual scenes that usually simulate impulse responses as many as possible but it could be challenging to collect a large number of impulse responses data in a real scene that has not been scanned.

5 Additional Visualizations of Loudness Maps for Multiple Scenes

We show additional rendered loudness maps for different scenes and various emitter locations in Fig. 3.

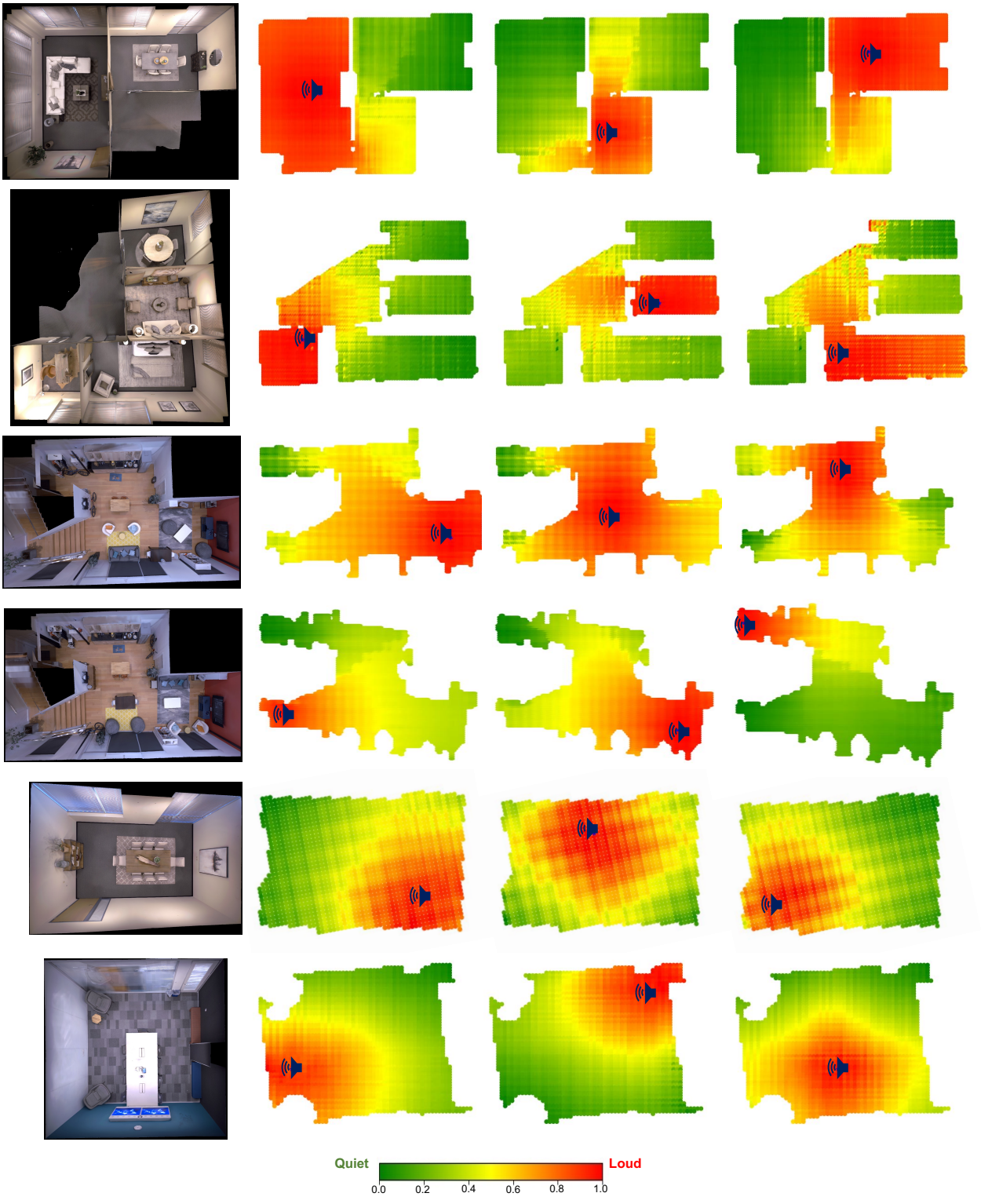

Figure 3: Loudness Maps for various scenes and emitter positions.

6 Potential Negative Societal Impact

Due to the fact that INRAS learns a continuous representation for audio scenes, one possible concern could be that the rendered spatial sounds could be used to manipulate an original non-spatial sound, and to create a non-authentic impression of the audio. Therefore, to mitigate these risks, the output of INRAS, i.e., generated impulse responses, would need to be authenticated before application to the non-spatial sound to prevent potential unethical or illegal use.

7 Dataset License