INRAS: Implicit Neural Representation for Audio Scenes

Anonymous Author(s)
Affiliation
Address
email

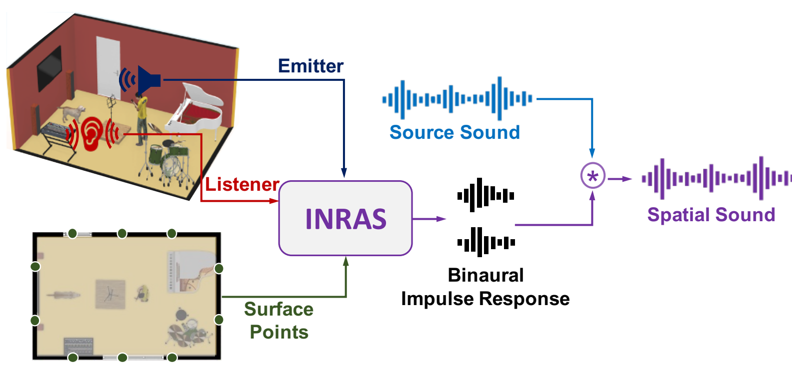

Figure 1: INRAS learns an implicit neural representation for audio scenes such that given the geometry of a scene, emitter and listener positions, INRAS renders the sound perceived by the listener. See supplementary video of demonstration examples of spatial sound rendering.

Abstract

The spatial acoustic information of a scene, i.e., how sounds emitted from a particular location in the scene are perceived in another location, is key for immersive scene modeling. Robust representation of scene's acoustics can be formulated through a continuous field formulation along with impulse responses varied by emitter-listener locations. The impulse responses are then used to render sounds perceived by the listener. While such representation is advantageous, parameterization of impulse responses for generic scenes presents itself as a challenge. Indeed, traditional acoustic field coding methods only implement parameterization at discrete probe points and rely on handcrafted features. In this work, we introduce a novel method for Implicit Neural Representation for Audio Scenes (INRAS) which renders high fidelity time-domain impulse responses at any arbitrary emitter-listener positions using neural network parameterization. Our experimental results show that INRAS outperforms existing approaches for representation and rendering of sounds for varying emitter-listener locations in all aspects, including the impulse response quality, inference speed, and storage requirements. INRAS achieves such enhancement in performance by introducing a novel audio scene feature decomposition, which leads to efficient reuse of scene-dependent features for any arbitrary emitter-listener positions. Furthermore, such a decomposition allows INRAS to generalize the representation from one scene to another with only a few additional parameters.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

1 Introduction

There are more than a billion buildings in the world, each of them with unique architecture, interior design and activities they are intended for. While vision is the primary sense for overall impression and navigation through the world's interior scenes, hearing plays a key role for a full immersion in a scene. Indeed, many of our daily activities in an interior scene, such as having a conversation with someone somewhere in the scene, listening to music or watching TV, calling our pets and locating them, are dependent on the hearing function. Hearing is the sense that allows us to experience the scene and interact with it, and the sound quality and its synchronization with the scene, completes our audio-visual perception. Indeed, the selection of scene acoustics plays a significant role in the activities that the scene would be used for. For example, a dedicated IMAX theater with the latest surround sound system will draw audience to watch the latest movies, while educational activities will be held in quiet classrooms, and coffee shops with their energetic, but not noisy, environment will draw visitors to work on their laptops. In these examples, spatial sound perception is affected by the collection of the reflected sounds bounced off the floor, walls, ceiling, and other reflective surfaces in the scene.

It is thus imperative to computationally model spatial audio aspects of interior scenes in order to adequately render a scene with spatial audio. However, computational modeling and representation of spatial audio in an arbitrary scene is a non-trivial task, and has been an ongoing research theme with long history in acoustics research [1]. Typically, the relationship between an arbitrary emitter sound and spatial sound can be represented by an impulse response, which is the function of time and the positions of the emitter and the listener [2]. For a real scene, an impulse response between the emitter and the listener can be usually measured by playing a sine sweep, using a loudspeaker at the emitter and recording the sound pressure with a microphone at the listener [3]. Alternatively, the impulse responses can be also simulated by computational geometry-based sound propagation techniques for a real or virtual scene [4, 5, 6, 7]. In both cases, it is time-consuming and computationally expensive to render impulse responses in a continuous space, and therefore prohibit more immersive, interactive spatial sound rendering in scenes. Classic encoding approaches parameterize the impulse responses using a few perceptual parameters that guide reproduction of reverberations [8, 9]. However, such features are typically custom and designed specifically for some scenes and therefore are difficult to reproduce impulse responses with high fidelity.

In this work, we propose an Implicit Neural Representation for Audio Scenes, INRAS, for efficient representation of spatial audio fields with high fidelity. In recent years, neural networks have been shown to parameterize implicit, continuous representations and achieved remarkable progress in computer graphics [10]. The infinite resolution property of such representation could be advantageous for representing the acoustic field as well. Since the acoustic wave equation governs the sound propagation from an emitter in a scene and its solution can be considered as a continuous field of impulse responses, the acoustic field can be encoded via a smooth, continuous representation which can alleviate the drawbacks of the approaches that encode the impulse response in discrete positions and perform interpolation during rendering [8, 9]. Furthermore, our approach is motivated by interactive sound propagation techniques using precomputed acoustic transfer operator for the scene, where the transfer operator is dependent on the scene geometry and decoupled from the emitter and the listener positions to render impulse response efficiently in interactive sound rendering applications [11, 12]. INRAS integrates the benefits of implicit neural representations and interactive acoustic transfer to render high fidelity impulse responses in an efficient way.

Specifically, INRAS is a light-weighted and efficient neural network model that can produce high fidelity spatial impulse responses at arbitrary emitter-listener positions. INRAS includes two main stages. In the first stage, it decomposes the audio scene features into three parallel modules: i) the Scatter module, ii) the Bounce module, and iii) the Gather module. Motivated by the disentangled procedures in the interactive acoustic radiance transfer techniques [11, 12], we design these three modules to generate independent features for the emitter, scene geometry, and listener, respectively. Indeed, disentangling the scene geometry features allows our model to generalize to multiple scenes by adding only a few trainable parameters. In the second stage, the listener module fuses the three independent features and generates the directional and binaural impulse responses. We show an overview of INRAS in Figs 1 and 3. In summary, our main contributions in this work are: 1) We propose a novel approach, INRAS, to learn the implicit neural representation for audio scenes that produce high fidelity time-domain impulse responses at arbitrary emitter-listener positions in the scene. 2) INRAS outperforms existing approaches on all metrics of audio rendering, including the impulse response quality, inference speed, and storage requirements. 3) We show that INRAS is robust and capable of generalizing across multiple scenes with a few additional parameters.

2 Related Work

Scene Acoustics Modeling. Modeling scene acoustics can be divided into two categories, 1) wave-based and, 2) geometry-based approaches. The first type of wave-based algorithms aims to solve the acoustic wave equation using numerical techniques [13, 14, 15, 16]. Due to the computation complexity of the wave equation, these approaches are typically used for lower frequencies. While wave methods have become more utilized with advancement of CPU/GPU computing power [17, 18], this cost directs existing methods to prefer geometric approximations of scene acoustics [19]. This second type of geometry-based approaches assume that the sound travels along a straight line, and determine the path of sound propagation according to the energy attenuation. These methods are generally faster than wave-based methods and are suitable for high-frequency sound propagation. However, with such an approach, it is difficult to accurately simulate low-frequency acoustic phenomena such as edge diffractions and surface scattering of arbitrary order. The commonly used geometric approaches are image sources methods [4, 5], ray-tracing [6, 7, 20], radiosity [21], and acoustic radiance transfer [22].

Furthermore, a general model of geometric room acoustics can be formulated as an integral equation. One of the first equations is the Kuttruff's integral equation for diffuse reflections in a convex room [23]. Multiple extensions of this mathematical model have been proposed subsequently, such as the room acoustic rendering equation which provides a framework for most geometric acoustic methods for interiors [24]. These algorithms for sound propagation are limited to static sources and/or listeners. Interactive applications are usually achieved by precomputing sound propagation effects such as precomputing acoustic radiance transfer from static sources [11, 12, 25]. While our work aims to represent the scene acoustics instead of performing simulation from scratch, the proposed INRAS model is motivated by the interactive acoustic radiance transfer method [11, 12].

Sound Field Encoding. Classical sound field encoding approaches represent the field around a listener point by capturing the sound from spatially distributed sources. For example, Ambisonics [26] represents the sound field around a point using spherical harmonic coefficients and independently of the reproduction setup (speakers or headphones). Parametric surround approaches, such as MPEG-Surround [27], assume a known speaker configuration around the listener. MPEG-H [28] extends the idea to allow encoding that is agnostic to the reproduction setup and supports higher-order Ambisonics and binaural rendering. The Spatial Decomposition Method (SDM) [29] fits an image source model to responses measured with a microphone array, approximating it at a point with multiple delayed spherical wavefronts. In Directional Audio Coding (DiRaC) [30], the input is the directional sound signal at a listener, which is a superposition of all sound source signals in a scene convolved with the corresponding directional impulse responses. DiRaC computes direction and a diffuseness parameter for each of many time-frequency bins. These approaches are static and do not allow the listener to navigate the scene and experience the change in sound while doing so. Several works for interactive sound field encoding propose to extract important features from precomputed impulse responses and synthesize them back using digital signal processing techniques [8, 31, 9]. However, these encodings typically cannot reproduce impulse responses with high fidelity.

Deep Acoustics. In recent years, deep learning approaches have been applied and developed for various acoustics applications. These include neural sound spatialization from a mono audio [32], estimation of room geometry and reflection coefficients from impulse response [33], reverberation time and direct-to-reverberation ratio prediction [34, 35], and learning the head-related transfer functions (HRTFs) [36]. In relation to scene modeling, deep neural networks modeling room impulse responses (RIR) have been studied extensively. A convolutional neural network model has been proposed to estimate room impulse response from reverberant speech [37]. Deep generative models such as IR-GAN [38] and fast-RIR [39] have been proposed to generate new realistic impulse responses. Recently, the emergence of implicit neural representations has shown great success in representing 3D geometry [40] and the appearance [10] of a scene. Such representation approach could be generalized to represent images, videos, and sounds [41] by learning a continuous mapping capable of capturing data at an "infinite resolution".

Indeed, very recently, it has been proposed to learn an implicit neural function to represent the room impulse responses [42, 43]. The Impulse Response Multi-layer perceptrons (IR-MLP) approach predicts impulse responses from spatio-temporal coordinates using an MLP but it does not support both moving sources and moving listeners scenarios [42]. Such a problem has been approached by Neural Acoustic Fields (NAF) [43], which proposes to learn a continuously map from all emitter and listener location pairs to a neural impulse response function using the magnitude component of the frequency-time spectrogram representation after applying Short-time-Fourier-Transform. While the smooth nature of the time-frequency spectrogram can be beneficial for training deep neural networks, the smoothness and entanglement of the time-frequency representation prediction also leads to imprecise modeling of high peaks that appear less frequently. For example, the sparse high peaks in the early reflection part of impulse response play a dominant role in our perceptual feelings for sound source directions and clarity. Moreover, modeling using the spectrogram magnitude ignores the phase information and adding random phase which may distort the audio signal significantly. In our approach, we learn the neural representation of the sound field for both *moving listeners* and *moving sources* scenarios. We aim to learn such implicit neural representation for rendering time-domain impulse responses instead of spectrograms. Our results show that INRAS can generate higher fidelity impulse responses with even fewer trainable parameters.

3 Methods

Problem Setup. INRAS implements several deep neural networks to model the continuous implicit function that maps scene's coordinates to the corresponding time-domain directional and binaural impulse responses of the sound field. More formally, for a given 3D scene D, we denote the sound emitter locations as $s \in \mathbb{R}^3$, the listener locations as $l \in \mathbb{R}^3$, and the listener head orientation $\theta \in \mathbb{R}^2$. Then $\forall (s, l, \theta) \in \mathbb{R}^8$ in the scene, there would be corresponding binaural impulse responses $h \in \mathbb{R}^{2 \times T}$ where T indicates the time length. We model the continuous function $f(s, l, \theta) \rightarrow h$ parameterized by a deep neural network that pairs s, l, θ with appropriate impulse response h. While the idea seems straightforward, training the network to learn the time domain impulse response from given coordinate inputs is challenging due to the typical long temporal length of impulse responses, and highly oscillating amplitude at different time samples, all which increase the training difficulty. One key insight is that while the scene geometry determines the impulse responses in the scene, it is always static no matter how emitter and listener positions vary and therefore the geometry based information could be shared with an arbitrary emitter and listener positions. Such an idea has been applied to interactive sound propagation based on acoustic radiance transfer [11, 12]. For training a neural network model, the approach would be to leverage the static scene geometry by learning reusable scene-dependent features, and associate with the emitter and the listener. This allows the model to realize that the differences between impulse responses at various emitter-listener locations are dependent on the scene geometry. Motivated by this approach, we propose two stage model. The first stage performs audio scenes feature decomposition to learn the independent scene geometric features and associate the emitter and listener to the scene. The second stage fuses these features to render the binaural impulse responses. In the following sections, we review the background of the interactive acoustic radiance transfer and then describe our model in detail.

Background on Interactive Acoustic Radiance Transfer. The acoustic radiance transfer is a classical approach to model sound propagation in complex room models and it can be derived from the acoustic rendering equation [24]

$$L(x, \Omega, t) = L_0(x, \Omega, t) + \int_S R(x, x', \Omega, t) L(x', \tfrac{x-x'}{|x-x'|}, t) dx', \tag{1}$$

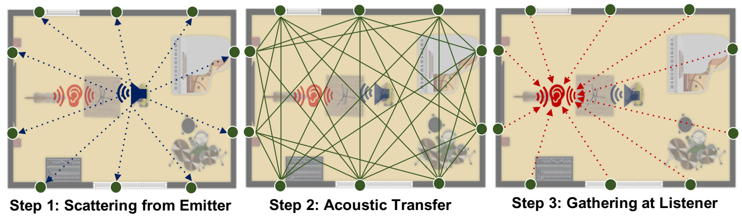

Figure 2: Acoustic radiance transfer steps overview.

where S is the set of all surface points in the scene, L is the total outgoing acoustic radiance, L_0 is the emitted acoustic radiance, Ω is the final radiance direction at x; the incident radiance direction at x is implicit in the specification of x', and R is the reflection kernel, which describes how radiance at point x' influences radiance at point x. The equation describes that the outgoing time-dependent radiance at any surface point is a combination of the reflected time-dependent radiance and the emitted time-dependent radiance.

The acoustic radiance transfer algorithm can be summarized in three steps (See Fig. 2). In the first step, the scene's boundary is divided into N bounce points, and energy is scattered from the emitter to all bounce points. In the second step, sound energy is emitted in all directions from a given bounce point. It propagates through the scene until the propagation is finally terminated upon an incidence at some other bounce point. The energy-time curve on each bounce point can be stored as an echogram. In the final step, the listener gathers energy responses from all bounce points. In interactive extensions [11, 12], a linear acoustic transfer operator is precomputed to model the propagation of acoustic radiance between bounce points distributed over the surface of the scene. In other words, the acoustic transfer operator can be seen as the scene-dependent features that are shared with all emitter-listener locations. Such disentanglement efficiently updates the impulse response at various emitter-listener positions by computing the propagation delay based on the relative distance to the bounce points. This motivates us to design a neural network model with similar decoupled modules to satisfy that the scene geometry information can be realized and reused by an arbitrary emitter and listener.

Implicit Neural Representation for Audio Scenes. INRAS includes two main components:(a) audio scenes feature decomposition, and (b) spatial binaural impulse response prediction. In (a), there are three parallel modules: 1) the Scatter module learns features to associate the emitter with bounce points; 2) the Bounce module learns the scene-dependent features shared by all emitter and listener positions; 3) the Gather module learns features to associate the listener with the bounce points. In (b), we fuse the output features of the three parallel modules and render the directional and binaural impulse responses. A system overview is shown in Fig. 3.

Scatter Module. Similar to computing the initial radiance scattering from the emitter to all bounce points in acoustic radiance transfer, the Scatter module is dependent on the relative distance between the emitter position and every bounce point position. We divide the surface of the scene into N bounce points with 3D locations $\{b_i\}_{i=1}^{N} \in \mathbb{R}^3$. We compute the relative distance between the emitter position s to all bounce points $\{d_{b_i}^s\}_{i=1}^{N}$. Using relative distance as input instead of absolute position enables the emitter to be aware of the scene geometry and allows the model to learn smooth continuous features for various emitter positions. We use the sinusoidal encoding to map the input $\{d_{b_i}^s\}_{i=1}^{N}$ to a higher dimension, as also used in graphical implicit neural representation [10]. We learn a function F_Θ parameterized by a fully connected network. We denote the output feature as $I = F_\Theta(\{d_{b_i}^s\}_{i=1}^{N}) \in \mathbb{R}^{N \times D}$, where D indicates the feature dimension. In our experiments, we find that it is sufficient to use 40 to 60 bounce points to cover the scene structure. We perform more investigations of bounce points selection in ablation studies.

Bounce Module. We design the bounce module to generate features representing the geometry of static scenes shared with arbitrary emitter and listener locations. To model such scene dependent features, we learn a function U_Φ parameterized by a multi-layer perceptron (MLP) with residual

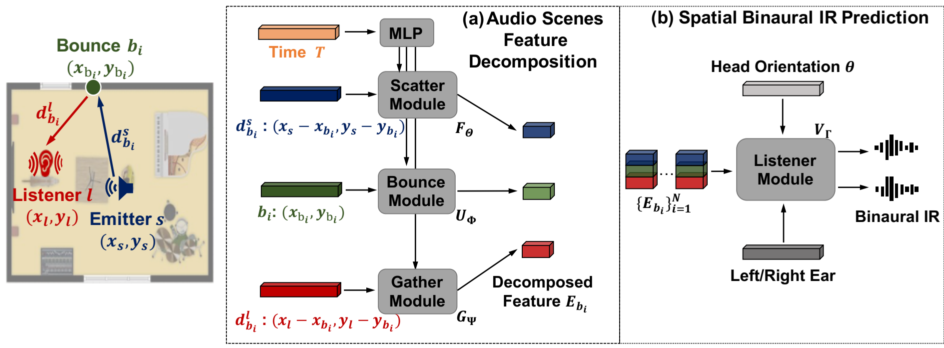

Figure 3: System Overview of INRAS. In audio scenes feature decomposition, inputs to scatter/gather module are the relative distances between the emitter/listener locations and bounce points. The bounce module takes all bounce points to generate scene-dependent features. In the second stage, the decomposed features are stacked and fed to the listener module which generates the spatial binaural impulse responses.

connections that takes all bounce points positions $\{b_i\}_{i=1}^N \in \mathbb{R}^3$ as input and outputs the features $Q = U_\Phi(\{b_i\}_{i=1}^N) \in \mathbb{R}^{N\times D}$.

Gather Module. This module is similar to the scatter module. We aim to associate the listener with the bounce points in the scene. We compute the relative distance between the listener position l to all bounce points: $\{d^l_{b_i}\}_{i=1}^N$. We also use sinusoidal encoding and learn a function G_Ψ parameterized by a fully connected network to generate the output feature $O = G_\Psi(\{d^l_{b_i}\}_{i=1}^N) \in \mathbb{R}^{N\times D}$.

Spatial-Time Feature Composition. The modules do not incorporate the time-dependencies. Adding the time dimension in every module could significantly slow down the training procedure. Motivated by the acoustic operator decomposition in the interactive sound propagation [12], the energy-time echogram for a specific bounce point $b_i(t)$ can be represented by a set of time domain basis functions $\{\tau^k(t)\}_{k=1}^K$ via a linear combination: $b_i(t) = \sum_{k=1}^K \alpha_k \tau^k(t)$, where α's are coefficients in the basis space. Similarly, we learn a function P_τ through a fully connected network to obtain a set of time-domain basis functions which can be reused by all spatial features. We encode the time samples $\{t_j\}_{j=1}^T$ using sinusoidal encoding. The output is denoted as $M = P_\tau(\{t_j\}_{j=1}^T) \in \mathbb{R}^{T\times D}$. We then perform fast matrix multiplication to obtain spatial-time features $\hat{I} = MI^\top$, $\hat{Q} = MQ^\top$ and $\hat{O} = MO^\top$.

Listener Module. In the stage of spatial binaural impulse response prediction, the listener module first performs feature fusions by concatenating the three features together $E = \{\hat{I}, \hat{Q}, \hat{O}\} \in \mathbb{R}^{T\times 3N}$, where $\{E_{b_i}\}_{i=1}^N \in \mathbb{R}^{T\times 3}$ represents fused spatial-time features for l and s associated with the bounce point b_i. We feed E as input to the listener module and further takes care of the head orientation conditions θ encoded by a learnable embedding matrix. We model the listener module via MLP and generate binaural impulse responses in time-domain $h = V_\Gamma(E, \theta)$.

Training and Rendering. All components and modules of INRAS are trained jointly. We use a combination of mean square error loss $L_{\text{mse}} = \|h - \hat{h}\|_2^2$ and multi-resolution STFT loss $L_{\text{mr\_stft}}$ which has been shown effective in modeling audio signals in the time domain [44]. The multi-resolution STFT loss first converts the impulse response into frequency-time domain $H = \text{STFT}(h)$ and computes the spectral convergence loss $L_{\text{sc}} = \frac{\||H| - |\hat{H}|\|_2}{\||H|\|_2}$, the magnitude loss $L_{\text{mag}} = \||H| - |\hat{H}|\|_1$ and the phase loss $L_{\text{phase}} = \|\phi(H) - \phi(\hat{H})\|$, our total loss can be summarized as follow:

$$L_{\text{mr\_stft}} = L_{\text{sc}} + L_{\text{mag}} + L_{\text{phase}}, L_{\text{total\_loss}} = L_{\text{mse}} + L_{\text{mr\_stft}} \tag{2}$$

Once we obtain the impulse response h, we can render sounds perceived at the listener location by convolving the impulse response with a sound source y. The final sound is denoted as $\hat{y} = h \circledast y$.

Generalization to Multiple Scenes. The design of INRAS enables the emitter and the listener to be aware of scene geometry by computing the relative distance to the bounce points in scatter and gather modules and the bounce module provides a static scene-dependent feature. Intuitively, we can include the collection of bounce points from multiple scenes and let the emitter and listener realize which scene they are in to achieve the generalization goal. Specifically, we normalize the coordinate space of multiple scenes and adapt the total number of bounce points $N_{\text{total}} = \sum_{i=1}^{K} N_i$ for K scenes. When computing the relative distance and bounce points features for the emitter/listener in a specific scene, we mask the other irrelevant bounce points. Since all other components and feature dimension are kept the same, such operation adds a handful of trainable parameters due to the increased bounce points number and in turn enables the generalization from scene to scene.

4 Experiments

Datasets. To evaluate our method, we use the *Soundspaces* dataset which consists of dense pairs of impulse responses generated by geometric sound propagation methods [45]. All scenes have the same height and provide the binaural impulse responses for four different head orientations $(0, 90, 180, 270$ degrees). For a fair comparison to the previous work [43], we re-sample all impulses responses to 22050 sampling rate and use the same 6 scenes including 2 multi-room layouts, 2 rooms with non-rectangular walls, and 2 single rooms with rectangular walls. For each scene, we use 90% data for training and hold 10% data for testing.

Implementation Details. We use Pytorch to implement all INRAS models. For all scenes, we extract the bounce points from the mesh boundary, $(40$ to 60, depending on the scene). We encode the relative distance from emitter/listener to bounce points using sinusoidal encoding with 10 frequencies of $\sin$ and $\cos$ functions. We use a fully connected layer in the scatter module and gather module. In the bounce module, we use a 4-layer residual MLP. In the listener module, we use a 6-layer residual MLP. In all MLPs, we use 256 neurons and set PreLU as the activation function. We use AdamW optimizer [46] to train all models on a Tesla T4 GPU for 100 epochs with a batch size of 64. The initial learning rate is set as 5e-4 and is gradually decreased by a factor of 0.95.

Baseline Methods. We compare our method to existing learning-based and classical approaches. For learning-based approaches, we compare INRAS with NAF [43]. We also compare two audio coding methods Advanced Audio Coding (AAC) and Xiph Opus by applying both linear and nearest neighbor interpolation to the coded acoustic fields.

Evaluation Metrics. We evaluate all methods on three aspects: the impulse response quality, the storage requirements and inference speed. We first compute acoustic parameters to evaluate the impulse response quality. We use acoustic parameter Clarity (C50) to quantify the part of early reflections of the impulse response which is associated with music loudness, speech intelligibility, and clarity. To study the effects of the late reverberation parts, we use reverberation (T60) and early decay time (EDT) to illustrate the statistical portion of the impulse response. The reverberation time (T60) measures how long it takes for the acoustic energy to decay by 60 dB. EDT is closely related to the listener's perception of reverberation but it is also affected by the early reflections of the impulse responses. We illustrate for the acoustic metrics can be found in Fig. 4. In addition, we also compute the storage requirements for saving audio scenes representations and the inference speed for rendering a binaural impulse response in the scene. For fair comparison, we test inference speed for all methods consistently on a Telsa T4 GPU.

Results. The quantitative evaluation results are shown in Table 1. INRAS outperforms both traditional audio coding and learning-based methods in all metrics. In particular, C50 and EDT errors outperform NAF by 43% and 39%, indicating that the early reflection part of our rendered impulse responses is much closer to the ground truth. Fig. 4 illustrates comparison of two examples of rendered impulse responses waveforms of AAC-linear, NAF and INRAS method. On the top left of the figure, we visualize the impulse responses loudness map of INRAS where colors indicate the loudness amplitude. In the two right columns, the comparison shows that the AAC-linear results have large gaps from the ground truth. While NAF is able to capture the exponentially decay pattern for reverberation, it cannot capture the the early reflection part of impulse responses which include the high peaks that are important for clarity. In comparison, INRAS can render both the early reflections and late

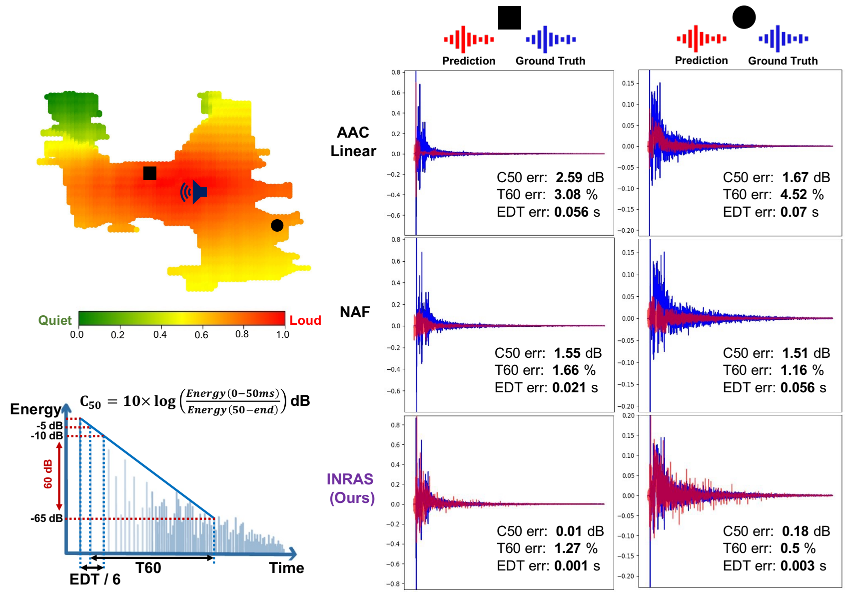

Figure 4: Rendered Impulse Responses Waveform Visualization. The speaker indicates the emitter location. We show examples of rendered waveforms at two listener locations (black square and circle) demonstrating metrics upon which performance of is evaluated AAC-Linear, NAF and INRAS rendering methods.

Model\Metric	C50 error (dB) ↓	T60 error (%) ↓	EDT error (sec) ↓	Parameters (Million) ↓	Storage (MB) ↓	Speed (ms) ↓
Opus-nearest	3.58	10.10	0.115	-	181.37	-
Opus-linear	3.13	8.64	0.097	-	181.37	-
AAC-nearest	1.67	9.35	0.059	-	346.74	-
AAC-linear	1.68	7.88	0.057	-	346.74	-
NAF	1.06	3.18	0.031	2.23	8.55	37.86
INRAS (Ours)	**0.6**	**3.14**	**0.019**	**0.67**	**2.56**	**9.47**

Table 1: Quantitative evaluation for impulse response quality, storage requirements and inference speed. Results are in the average of six single scene models.

reverberation much closer to the ground truth impulse responses. For more qualitative visualization on loudness maps and waveforms, please refer to Suppl. Materials. Moreover, our INRAS model only takes about 0.65 million trainable parameters which results in less than 3MB storage and 4ms inference speed, indicating the INRAS is significantly light-weighted and efficient.

Generalization to Multiple Scenes. As discussed in the method section, the effective audio scene feature decomposition allows us to train a single INRAS to generalize from scene to scene. We investigated this property by training a single INRAS model on three scenes with different types of layouts. We selected one multi-room layout, one room with non-rectangular walls, and one room with rectangular walls (See Fig. 5). As expected, INRAS can learn continuous implicit neural representations for all three scenes. We illustrate the loudness maps for all three scenes learned by one single model and in Table 2. We show quantitative results of the multi-scene model. For other methods, we compute the average values for the three scenes. In addition to the acoustic parameters that evaluate the impulse response quality, we further evaluate the quality of the final rendered audio signal after convolving the impulse response with a sound source. Specifically, we compute the Signal-to-Noise ratio (SNR) and audio Peak Signal to Noise Ratio (PSNR). The results in Table 2 clearly shows that our generalized model can achieve high-quality results and better overall accuracy than NAF. Notably, the number of trainable parameters in INRAS increases by 0.1M to extend the single-scene to multi-scenes thus keeping the storage requirement less than 3MB. In comparison, other approaches have increased the storage size linearly.

Ablation Studies. To show the effectiveness of INRAS v.s. similar variants, we use a representative scene to perform ablation studies. Table 3 shows comparison results of INRAS and its ablated variants.

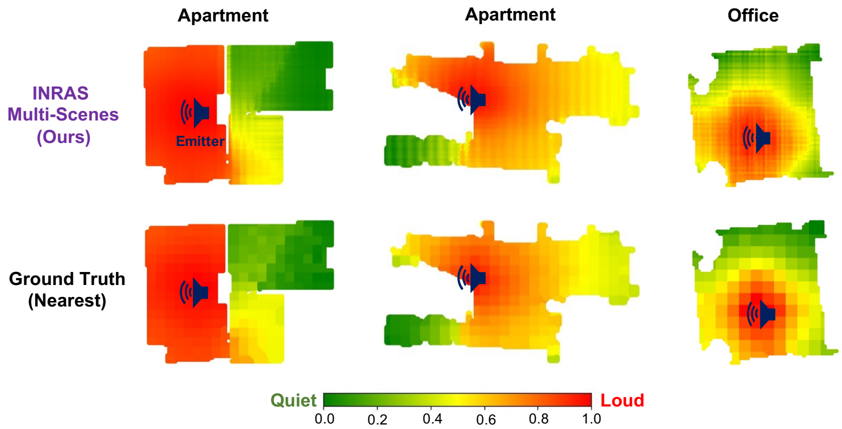

Figure 5: Loudness map visualization comparing INRAS multi-scenes rendering on three scenes (Top) with the ground truth using nearest neighbors (Bottom)

Model\Metric	Multi-scenes	SNR (dB) ↑	PSNR (dB) ↑	C50 error (dB) ↓	T60 error (%) ↓	Storage (MB) ↓
Opus-nearest	✗	3.18	13.35	3.6	10.1	544.11
Opus-linear	✗	3.57	13.45	3.23	8.7	544.11
AAC-nearest	✗	6.48	17.84	1.51	9.64	1040.31
AAC-linear	✗	7.52	18.7	1.57	8.05	1040.31
NAF	✗	-1.54	11.25	1.05	**3.01**	25.65
INRAS (Ours)	✓	**8.06**	**18.80**	**0.68**	4.09	**2.99**

Table 2: Quantitative evaluation of INRAS Multi-Scene generalization on three scene layouts. Results for other methods are computed as an average of three scenes.

We first implement a brute-force model (Simple INRAS) using a residual MLP like NAF architecture and provide the normalized emitter and listener positions as input to predict the time domain impulse response using MSE loss only. The result turns out to be unsuccessfully in all metrics. We further show that adding the multi-resolution STFT loss can improve the T60 error but still fails to capture the early reflection part. Next, we show that without using the relative distance impairs the results since the emitter and the listener could not realize the scene geometry. Besides, removing the bounce module eliminates the static scene feature and therefore impairs the performance. We also investigate to the importance of bounce point selection. We sample two types of bounce points that both have the same total number as the original setting but they do not cover the whole scene, i.e., missing some boundaries. The results show that only using bounce points covered the full scene geometry can achieve the best performance in all results.

Model\Metric	C50 err (dB) ↓	T60 err (%) ↓	EDT err (sec) ↓
Simple INRAS w. L_{mse}	1.47	49.6	0.048
Simple INRAS w. $L_{\text{mse}} + L_{\text{mr\_stft}}$	2.20	6.40	0.074
INRAS w.o. rel. dist.	1.12	3.52	0.038
INRAS w.o. bounce module	0.63	2.30	0.019
INRAS w. more incomplete bounce points	0.50	2.31	0.019
INRAS w. less incomplete bounce points	0.49	2.17	0.018
INRAS (Ours)	**0.44**	**2.07**	**0.017**

Table 3: Ablation Studies of INRAS variants.

5 Conclusion

In conclusion, here we present INRAS, a novel implicit neural representation for audio scenes. INRAS is a light-weight, fast model that effectively renders high fidelity impulse responses for multiple audio scenes. We achieve such function by leveraging a novel reusable representation of scene-dependent features and associate them with emitter and listener. Experimental results demonstrate that INRAS outperforms other methods in all metrics and we further show that INRAS generalizes across scenes.

References

[1] Shiguang Liu and Dinesh Manocha. Sound synthesis, propagation, and rendering: a survey. *arXiv preprint arXiv:2011.05538*, 2020.

[2] Tor Erik Vigran. *Building acoustics*. CRC Press, 2014.

[3] Guy-Bart Stan, Jean-Jacques Embrechts, and Dominique Archambeau. Comparison of different impulse response measurement techniques. *Journal of the Audio engineering society*, 50(4):249–262, 2002.

[4] Jont B Allen and David A Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

[5] Jeffrey Borish. Extension of the image model to arbitrary polyhedra. *The Journal of the Acoustical Society of America*, 75(6):1827–1836, 1984.

[6] Asbjørn Krokstad, Staffan Strom, and Svein Sørsdal. Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound and Vibration*, 8(1):118–125, 1968.

[7] Michael Vorländer. Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm. *The Journal of the Acoustical Society of America*, 86(1):172–178, 1989.

[8] Nikunj Raghuvanshi and John Snyder. Parametric wave field coding for precomputed sound propagation. *ACM Transactions on Graphics (TOG)*, 33(4):1–11, 2014.

[9] Nikunj Raghuvanshi and John Snyder. Parametric directional coding for precomputed sound propagation. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[10] 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 *European conference on computer vision*, pages 405–421. Springer, 2020.

[11] Lakulish Antani, Anish Chandak, Micah Taylor, and Dinesh Manocha. Direct-to-indirect acoustic radiance transfer. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):261–269, 2011.

[12] Lakulish Antani, Anish Chandak, Lauri Savioja, and Dinesh Manocha. Interactive sound propagation using compact acoustic transfer operators. *ACM Transactions on Graphics (TOG)*, 31(1):1–12, 2012.

[13] Brian Hamilton and Stefan Bilbao. Fdtd methods for 3-d room acoustics simulation with high-order accuracy in space and time. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(11):2112–2124, 2017.

[14] Lonny L Thompson. A review of finite-element methods for time-harmonic acoustics. *The Journal of the Acoustical Society of America*, 119(3):1315–1330, 2006.

[15] Nail A Gumerov and Ramani Duraiswami. A broadband fast multipole accelerated boundary element method for the three dimensional helmholtz equation. *The Journal of the Acoustical Society of America*, 125(1):191–205, 2009.

[16] Nikunj Raghuvanshi, Rahul Narain, and Ming C Lin. Efficient and accurate sound propagation using adaptive rectangular decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):789–801, 2009.

[17] Ravish Mehra, Nikunj Raghuvanshi, Lauri Savioja, Ming C Lin, and Dinesh Manocha. An efficient gpu-based time domain solver for the acoustic wave equation. *Applied Acoustics*, 73(2):83–94, 2012.

[18] Hengchin Yeh, Ravish Mehra, Zhimin Ren, Lakulish Antani, Dinesh Manocha, and Ming Lin. Wave-ray coupling for interactive sound propagation in large complex scenes. *ACM Transactions on Graphics (TOG)*, 32(6):1–11, 2013.

[19] Samuel Siltanen, Tapio Lokki, and Lauri Savioja. Rays or waves? understanding the strengths and weaknesses of computational room acoustics modeling techniques. In *Proc. Int. Symposium on Room Acoustics*, 2010.

[20] Chunxiao Cao, Zhong Ren, Carl Schissler, Dinesh Manocha, and Kun Zhou. Interactive sound propagation with bidirectional path tracing. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016.

[21] Eva-Marie Nosal, Murray Hodgson, and Ian Ashdown. Improved algorithms and methods for room sound-field prediction by acoustical radiosity in arbitrary polyhedral rooms. *The Journal of the Acoustical Society of America*, 116(2):970–980, 2004.

[22] Samuel Siltanen, Tapio Lokki, and Lauri Savioja. Room acoustics modeling with acoustic radiance transfer. *Proc. ISRA Melbourne*, 2010.

[23] Heinrich Kuttruff. *Room acoustics*. Crc Press, 2016.

[24] Samuel Siltanen, Tapio Lokki, Sami Kiminki, and Lauri Savioja. The room acoustic rendering equation. *The Journal of the Acoustical Society of America*, 122(3):1624–1635, 2007.

[25] Carl Schissler, Ravish Mehra, and Dinesh Manocha. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.

[26] Michael A Gerzon. Periphony: With-height sound reproduction. *Journal of the audio engineering society*, 21(1):2–10, 1973.

[27] Jeroen Breebaart, Sascha Disch, Christof Faller, Jürgen Herre, Gerard Hotho, Kristofer Kjörling, Francois Myburg, Matthias Neusinger, Werner Oomen, Heiko Purnhagen, et al. Mpeg spatial audio coding/mpeg surround: Overview and current status. In *Audio Engineering Society Convention 119*. Audio Engineering Society, 2005.

[28] Jürgen Herre, Johannes Hilpert, Achim Kuntz, and Jan Plogsties. Mpeg-h audio—the new standard for universal spatial/3d audio coding. *Journal of the Audio Engineering Society*, 62(12):821–830, 2015.

[29] Sakari Tervo, Jukka Pätynen, Antti Kuusinen, and Tapio Lokki. Spatial decomposition method for room impulse responses. *Journal of the Audio Engineering Society*, 61(1/2):17–28, 2013.

[30] Mikko-Ville Laitinen, Tapani Pihlajamäki, Cumhur Erkut, and Ville Pulkki. Parametric time-frequency representation of spatial sound in virtual worlds. *ACM Transactions on Applied Perception (TAP)*, 9(2):1–20, 2012.

[31] Ravish Mehra, Lakulish Antani, Sujeong Kim, and Dinesh Manocha. Source and listener directivity for interactive wave-based sound propagation. *IEEE transactions on visualization and computer graphics*, 20(4):495–503, 2014.

[32] Alexander Richard, Dejan Markovic, Israel D Gebru, Steven Krenn, Gladstone Alexander Butler, Fernando Torre, and Yaser Sheikh. Neural synthesis of binaural speech from mono audio. In *International Conference on Learning Representations*, 2020.

[33] Wangyang Yu and W Bastiaan Kleijn. Room acoustical parameter estimation from room impulse responses using deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:436–447, 2020.

[34] Hannes Gamper and Ivan J Tashev. Blind reverberation time estimation using a convolutional neural network. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 136–140. IEEE, 2018.

[35] Nicholas J Bryan. Impulse response data augmentation and deep neural networks for blind room acoustic parameter estimation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2020.

[36] Israel D Gebru, Dejan Marković, Alexander Richard, Steven Krenn, Gladstone A Butler, Fernando De la Torre, and Yaser Sheikh. Implicit hrtf modeling using temporal convolutional networks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3385–3389. IEEE, 2021.

[37] Christian J Steinmetz, Vamsi Krishna Ithapu, and Paul Calamia. Filtered noise shaping for time domain room impulse response estimation from reverberant speech. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 221–225. IEEE, 2021.

[38] Anton Ratnarajah, Zhenyu Tang, and Dinesh Manocha. Ir-gan: Room impulse response generator for far-field speech recognition. *arXiv preprint arXiv:2010.13219*, 2020.

[39] Anton Ratnarajah, Shi-Xiong Zhang, Meng Yu, Zhenyu Tang, Dinesh Manocha, and Dong Yu. Fast-rir: Fast neural diffuse room impulse response generator. *arXiv preprint arXiv:2110.04057*, 2021.

[40] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[41] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[42] Alexander Richard, Peter Dodds, and Vamsi Krishna Ithapu. Deep impulse responses: Estimating and parameterizing filters with deep networks. *arXiv preprint arXiv:2202.03416*, 2022.

[43] Andrew Luo, Yilun Du, Michael J Tarr, Joshua B Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *arXiv preprint arXiv:*, 2021.

[44] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE, 2020.

[45] 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 *European Conference on Computer Vision*, pages 17–36. Springer, 2020.

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The sections of Methods and Experiments clearly describe the claims we made.
   (b) Did you describe the limitations of your work? [Yes] We describe the limitations of our work in the supplementary material.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We describe such impacts in the supplementary material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The inference code is available in the supplementary material. The full code will be available in the Github after the review process.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We describe the training details in the implementation details section and more details can be found in the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] No. We fix the random seed for reproduction purpose. The errors bars are not reported because it would be too computationally expensive.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We describe resources in the implementation details section.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite all the existing assets used in our work.
   (b) Did you mention the license of the assets? [Yes] we mention the license of the assets in the supplementary material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We discuss it in the supplementary material.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] The human evaluation is fully described in the supplementary material.

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] there is no potential risk in our human evaluation.

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] we include these material in the supplementary material.