Learning Acoustic Scattering Fields for Highly Dynamic Interactive Sound Propagation

HSIEN-YU MENG, University of Maryland, USA

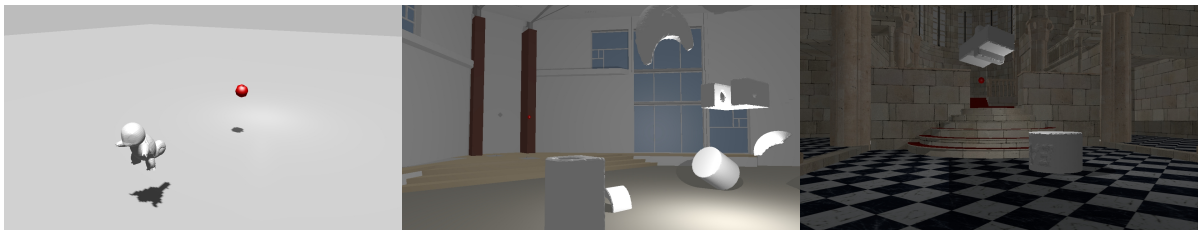

Fig. 1. We highlight the dynamic scenes with multiple moving objects that are used to evaluate our hybrid sound propagation algorithm. We compute the acoustic scattered fields of each object using a neural network and couple them with interactive ray tracing to generate diffraction and occlusion effects. Our approach can handle arbitrary dynamic scenes and takes few milliseconds (per frame) on a multi-core PC.

We present a novel hybrid sound propagation algorithm for interactive applications. Our approach is designed for arbitrary dynamic scenes and uses a neural network-based learned scattered field representation along with ray tracing to generate specular, diffuse, diffraction, and occlusion effects efficiently. To handle general objects, we exploit properties of the acoustic scattering field and use geometric deep learning on differential coordinates to approximate the field using spherical harmonics. We use a large dataset for training, and compare its accuracy with the ground truth generated using an accurate wave-based solver. The additional overhead of computing the learned scattered field at runtime is small and we highlight the interactive performance by generating plausible sound effects in dynamic scenes.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; **Virtual reality**; Point-based models.

ACM Reference Format:
Hsien-Yu Meng. 2025. Learning Acoustic Scattering Fields for Highly Dynamic Interactive Sound Propagation. *ACM Trans. Graph.* 1, 1 (May 2025), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Interactive sound propagation and rendering are increasingly used to generate plausible sounds that can improve a user's sense of presence and immersion in virtual environments. Recent advances in geometric and wave-based methods have lead to integration of these methods into current games and VR applications like Microsoft Project Acoustics [Mic 2019], Oculus Spatializer [Ocu 2019], and Steam Audio [Ste 2018]. The underlying propagation algorithms are based on using reverberation filters [Valimaki et al. 2012], ray tracing [Schissler and Manocha 2018; Schissler et al. 2014], or precomputed wave-based acoustics [Raghuvanshi and Snyder 2014].

A key challenge in interactive sound rendering is handling arbitrary dynamic scenes that are frequently used in games and VR applications. Not only can the objects undergo arbitrary motion or deformation, but their topologies may also change. In addition to specular and diffuse effects, it is also important to simulate complex diffracted scattering, occlusions, and inter-reflections that are perceptible [James et al. 2006; Pulkki and Svensson 2019; Raghuvanshi and Snyder 2014]. Prior geometric methods are accurate in terms of simulating high-frequency effects and can be augmented with approximate edge diffraction methods that may work well in certain cases [Schissler et al. 2014; Tsingos et al. 2001]. On the other hand, wave-based precomputation methods can accurately simulate these effects, but are limited to static scenes [Raghuvanshi and Snyder 2014, 2018]. Some hybrid methods are limited to dynamic scenes with well-separated rigid objects [Rungta et al. 2018]. Overall, no good interactive solutions are known for general dynamic scenes.

A recent trend is to use machine learning techniques for audio processing, including recovering acoustic parameters of real-world scenes from recordings [Eaton et al. 2016; Genovese et al. 2019; Tsokaktsidis et al. 2019]. Furthermore, learning methods have been used to approximate diffraction scattering and occlusion effects from rectangular plate objects [Pulkki and Svensson 2019] and frequency-dependent loudness fields for convex shapes [Fan et al. 2020]. These results are promising and have motivated us to develop good learning based methods for arbitrary dynamic scenes.

Main Results: We present a novel approach to approximate the acoustic scattering field of any geometric object using neural networks for interactive sound propagation of highly dynamic scenes. Our approach is general and makes no assumption about the scene or the motion or topology of the objects. We exploit properties of the acoustic scattering field of objects for lower frequencies and use neural networks to learn this field from geometric representations of the objects. In particular, we compute a point cloud representation of 3D objects and use that to approximate the angular part of acoustic wave propagation in the free field using spherical harmonics.

Author's address: Hsien-Yu Meng, University of Maryland, School of Engineering, College Park, MD, 20742, USA, mengxy19@cs.umd.edu.

ACM 0730-0301/2025/5-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Given any dynamic object, we use the neural network to estimate the scattered field at runtime, which is used to compute the propagation paths when sound waves interact with objects in the scene. The radial part of the acoustic scattering field is estimated using geometric ray tracing, along with specular and diffuse reflections. Some of the novel components of our work include:

- **Learning Acoustic Scattering Fields:** We present novel techniques based on geometric deep learning on differential coordinates to approximate the angular component of acoustic wave propagation in the wave-field. Each point in the point cloud representation is encoded in a high-dimensional latent space. Moreover, the local surface shapes in the latent space are encoded using implicit surfaces. This enables us to handle arbitrary topology. We use a four-layer neural network that takes the point cloud as an input and outputs the spherical harmonic coefficients that represent the acoustic scattering field. We perform an ablation study to highlight the benefits of our approach.
- **Interactive Wave-Geometric Sound Propagation:** We present a hybrid algorithm that uses a neural network-based scattering field representation along with ray tracing to efficiently generate specular, diffuse, diffraction, and occlusion effects at interactive rates.
- **Sound rendering of highly dynamic scenes:** We present the first interactive approach for plausible sound rendering in arbitrary dynamic scenes. As the objects deform or come in close proximity, we compute a new spherical harmonic representation using the neural network. Compared with prior interactive geometric or filter-based methods, we can handle unseen objects in highly dynamic scenes at real-time, without using any precomputed transfer functions.

We highlight the performance in dynamic scenes with multiple moving objects. The additional runtime overhead of estimating the scattering field from neural networks is less than 1ms per object on a NVIDIA GeForce RTX 2080 Ti GPU. The overall running time of sound propagation is governed by the underlying ray tracing system and takes few milliseconds per frame on multi-core desktop PC. We also compare the accuracy of acoustic scattering fields approximated using a neural network with an accurate boundary-element method (BEM) solver as shown in Figure 6 and Figure 7. We evaluate the accuracy of our learning algorithm on a dataset of thousands of objects that are not seen in the training dataset and have varying size, orientation, convexity, and genus properties. We highlight the exact and approximated acoustic scattering fields. In practice, our approach generates plausible sound corresponding to continuous and smooth sound fields, as the listener moves in and out of occluded regions with respect to the sources. We plan to release the source code and the dataset on github.

2 RELATED WORK

2.1 Sound Propagation

Wave-based techniques to model sound propagation solve the acoustic wave equation directly using numerical solvers such as the finite-element method [Thompson 2006], the boundary-element method [Wrobel and Kassab 2003], the finite-difference time domain [Botteldooren 1995], adaptive rectangular decomposition [Raghuvanshi et al. 2009], etc. Their complexity increases linearly with the size of the environment (surface area or volume) and as a third or fourth power of frequencies. As a result, they are limited to lower frequencies (e.g., below $1000Hz$) [Mehra et al. 2013; Raghuvanshi et al. 2010; Yeh et al. 2013].

Geometric techniques model the acoustic effects based on ray theory and typically work well for high-frequency sounds to model specular and diffuse reflections [Funkhouser et al. 1998; Krokstad et al. 1968; Lauterbach et al. 2007; Savioja and Svensson 2015]. These techniques can be enhanced to simulate low-frequency diffraction effects. This includes the accurate time-domain Biot-Tolstoy-Medwin (BTM) model, which can be expensive and is limited to offline computations [Svensson et al. 1999]. For interactive applications, commonly used techniques are based on the uniform theory of diffraction (UTD), which is a less accurate frequency-domain model that can generate plausible results in some cases [Schissler et al. 2014; Taylor et al. 2012; Tsingos et al. 2001]. Moreover, the complexity of edge-based diffraction algorithms can increase exponentially with the maximum diffraction order.

2.2 Interactive Sound Rendering in Dynamic Scenes

At a broad level, techniques for dynamic scenes can be classified into reverberation filters, geometric and wave-based methods, and hybrid combinations. The simplest and lowest-cost algorithms are based on artificial reverberators [Valimaki et al. 2012], which simulate the decay of sound in rooms. These filters are designed based on different parameters and are either specified by an artist or computed using scene characteristics [Tsingos 2009]. They can handle dynamic scenes but assume that the reverberant sound field is diffuse, making them unable to generate accurate directional reverberation or time-varying effects.

Many interactive techniques based on geometric acoustics and ray tracing have been proposed for dynamic scenes [Schissler and Manocha 2017; Taylor et al. 2012; Vorländer 1989]. They use spatial data structures along with multiple cores on commodity processors and caching techniques to achieve higher performance. Furthermore, hybrid combinations of ray tracing and reverberation filters [Schissler and Manocha 2018] have been proposed for low-power, mobile devices. In practice, these methods can handle scenes with a large number of moving objects, along with sources and the listener, but can't model diffraction or occlusion effects accurately.

Many precomputation-based wave based techniques tend to compute a global representation of the acoustic pressure field. They are limited to static scenes, but can handle real-time movement of both sources and the listener [Mehra et al. 2015; Raghuvanshi et al. 2010]. These representations are computed based on uniform or adaptive sampling techniques [Chaitanya et al. 2019]. Overall, the acoustic wave field is a complex high-dimensional function and many efficient techniques have been designed to encode this field [Raghuvanshi and Snyder 2014, 2018] within 100MB and with a small runtime overhead. A hybrid combination of BEM and ray tracing has been presented for dynamic scenes with well-separated rigid objects [Rungta et al. 2018].

2.3 Machine Learning and Acoustic Processing

Machine learning techniques are increasingly used for acoustic processing applications. These include isolating the source locations in multipath environments [Ferguson et al. 2018] and recovering the room acoustic parameters corresponding to reverberation time, direct-to-reverberant ratio, room volume, equalization, etc. from recorded signals [Eaton et al. 2016; Genovese et al. 2019; Tang et al. 2020; Tsokaktsidis et al. 2019]. These parameters are used for speech processing or audio rendering in real-world scenes. Neural networks have also been used to replace the expensive convolution operations for fast auralization [Tenenbaum et al. 2019], to render the acoustic effects of scattering from rectangular plate objects for VR applications [Pulkki and Svensson 2019], or to learn the mapping from convex shapes to the frequency dependent loudness field [Fan et al. 2020]. The last method formulates the scattering function computation as a high-dimension image-to-image regression and is mainly limited to convex objects that are isomorphic to spheres. In contrast, our learning method exploits deep learning on differential coordinates and can compute a good approximation of the acoustic scattering field of arbitrary objects (e.g. non-convex or non-manifold).

3 BACKGROUND AND OVERVIEW

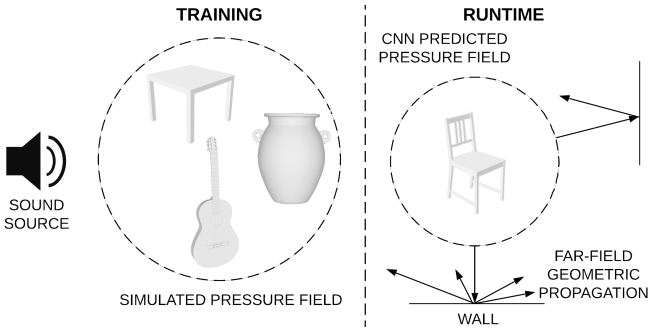

Fig. 2. **Overview:** Our algorithm consists of the training stage and the runtime stage. The training stage uses a large dataset of 3D objects and their associated acoustic pressure fields computed using a far-field source to train the network. The runtime stage uses the trained neural network to predict the sound pressure field from a point cloud approximation of different objects at interactive rates (about 1ms). The learned acoustic pressure field is used with geometric sound propagation techniques.

In this section, we provide some background on wave-based acoustics, pressure fields, point cloud representations and give an overview of our learning method. We highlight the notation and symbols in Table 1.

3.1 Wave Acoustics and the Helmholtz Equation

Our approach is designed for synthetic scenes and we assume a geometric representation (e.g., a triangle mesh) is given to us. A scalar acoustic pressure field, $P(\mathbf{x}, t)$, satisfies the homogeneous wave equation

$$\nabla^2 P - \frac{1}{c^2}\frac{\partial^2 P}{\partial t^2} = 0, \tag{1}$$

Table 1. Notation and symbols used throughout the paper.

$\mathbf{x}$	3D Cartesian coordinates.
(r, θ, ϕ)	Spherical coordinates.
c	Speed of sound, taken as constant $343m/s$.
ω	Frequency of sound.
k	Acoustic wavenumber.
$P(\mathbf{x}, t)$	Time domain acoustic pressure field.
$p(\mathbf{x}, \omega)$	Frequency domain acoustic pressure field.
m, l	Order and degree of spherical harmonics.
$Y_l^m(\theta, \phi)$	Spherical harmonics basis.
$c_l^m(\omega)$	Spherical harmonics coefficients.
$h_l^{(1)}(kr), h_l^{(2)}(kr)$	Hankel function of the first/second kind.
$Q(\mathbf{x}, \omega)$	Frequency domain sound sources.
$Pr(\theta, \phi)$	Probability of sampling a ray direction.

where c is the speed of sound. We can analyze the pressure field in the frequency domain using Fourier transform

$$p(\mathbf{x}, \omega) = \mathcal{F}_t\{P(\mathbf{x}, t)\} = \int_{-\infty}^{\infty} P(\mathbf{x}, t) e^{-j\omega t} dt. \tag{2}$$

At each frequency ω the pressure field satisfies the homogeneous Helmholtz wave equation

$$(\nabla^2 + k^2) p(\mathbf{x}, \omega) = 0, \tag{3}$$

where $k = \frac{\omega}{c}$ is the wavenumber. We can expand the Laplacian operator in terms of spherical coordinates (r, θ, ϕ) as

$$\left(\frac{\partial^2}{\partial r^2} + \frac{2}{r}\frac{\partial}{\partial r} + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta \frac{\partial}{\partial \theta}\right) + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2}{\partial \phi^2} + k^2\right) p = 0. \tag{4}$$

The general free-field solution of (4) can be formulated as

$$p(\mathbf{x}, \omega) = \sum_{l=0}^{\infty} \sum_{m=-l}^{+l} \left[A_{lm} h_l^{(1)}(kr) + B_{lm} h_l^{(2)}(kr)\right] Y_l^m(\theta, \phi), \tag{5}$$

where $h_l^{(1)}$ and $h_l^{(2)}$ are Hankel functions of the first and the second kind, respectively. A_{lm} and B_{lm} are arbitrary constants, $A_{lm} h_l^{(1)}(kr) + B_{lm} h_l^{(2)}(kr)$ together represents the radial part of the solution and the spherical harmonics term $Y_l^m(\theta, \phi)$ represents the angular part of the solution. In this work, we propose modeling the angular part using our learning based pressure field inference. The radial part is approximated using interactive geometric sound propagation combined with localized pressure fields.

3.2 Global and Localized Sound Fields

Sound fields typically refer to the sound energy/pressure distribution over a bounded space as generated by one or more sound sources. The global sound field in an acoustic environment depends on each sound source location, the propagating medium, and any reflections from boundary surfaces and objects. This requires solving the wave equation in the free-field condition and evaluating inter-boundary interactions of sound energy using a global numeric solver. In this case, the position of all scene objects/boundaries and sound sources needs to be specified beforehand, and any change in

these conditions changes the sound field. The exact computation of the global pressure field is very expensive and can takes tens of hours on a cluster [Mehra et al. 2013; Raghuvanshi and Snyder 2014; Raghuvanshi et al. 2010].

Our goal is to generate plausible sounds in indoor scenes with dynamic objects. Therefore, it is important to model the acoustic scattering field (ASF) of each object. The ASFs of different objects are used to represent the localized pressure field (e.g., the near-field), which is needed for diffraction and inter-reflection effects [James et al. 2006; Mehra et al. 2013]. At the same time, the sound field in the free space (e.g., the far-field) between two distant objects is approximated using ray tracing, and we do not compute that pressure field accurately using a wave-solver. In practice, computing the sound field in a localized space for each object in the scene is much simpler and easier to represent than using a global solver [Mehra et al. 2013; Rungta et al. 2018].

3.3 Overview

We present a learning method to approximate the ASFs of any static or dynamic object. It turns out that the acoustic pressure field due to scattering in a low-frequency sound source scattering exhibits fewer variations than in high-frequency sound scattering. As a result, it is more likely that a learning method will more accurately model the low-frequency sound effects. In terms of correlation between the object shape and its scattering field, the volume of the scatterer closely relates to its low-order shape characteristics that can be represented by coarse triangle faces, which dominate the low-frequency scattering behaviors; while at high frequencies, this relationship shifts to high-order shape characteristics (i.e., geometrical details). Given the powerfulness of deep learning inference, we hypothesize the scattering sound distribution can be directly learned from the scatterer geometry, without solving the complicated wave equations. The inference speed on a modern GPU far exceeds conventional wave solvers, making deep neural networks suitable for interactive sound rendering applications. Therefore, we propose using appropriate 3D representation of objects to feed a neural network that can learn its corresponding scattered acoustic pressure field. We build and evaluate our method mainly on low frequency sounds and leverage state-of-the-art geometric ray-tracing techniques to handle high frequency sounds. For each object, we consider a spherical grid of incoming directions and model the plane-waves from each direction of this grid. For each plane wave, our goal is to compute the scattered field for the object on an offset surface of the object. Our geometric deep learning method is used to compute the angular portion of the scattered field (Equation 5), which is expressed using a spherical harmonic basis. This scattered field approximation is computed at interactive rates at runtime using a pre-trained network. If two objects move and are in a touching configuration, our learning algorithm treats them as a one large object and estimates its scattered field. Similarly, we can recompute the scattered field for a deforming object.

An overview of our approach is illustrated in Figure 2. The precomputation phase consists of a large training module, and we compute a neural network-based representation to compute the acoustic scattering fields of objects. We use a large synthetic database and compute the ground truth scattering fields using an accurate BEM solver. At runtime, we use an interactive geometric propagation algorithm that uses ray and path tracing to generate specular and diffuse reflections and combine them with the scattered fields to simulate localized wave effects.

3.4 Point Cloud Representations

Our goal is to use an appropriate geometric representation for the underlying objects in the scene so that we can apply geometric deep learning methods to compute the sound scattering field. It is important that our approach should be able handle highly dynamic scenes with arbitrary moving objects or changing topology. It can be difficult to handle such scenarios with mesh-based representations [Hanocka et al. 2019; Tan et al. 2018; Zheng et al. 2017]. For example, [Hanocka et al. 2019] calculates intrinsic geodesic distances for convolution operations, which cannot be applied when one big object breaks into two. Furthermore, we would like to represent the ASFs using spherical harmonics, so that they can be easily integrated with ray-tracing based sound rendering engines. Our approach uses a point cloud representation of the objects in the scene as an input. We represent each point and its local surface by a higher dimension implicit surfaces in the latent space formed by an implicit surface encoder as shown in the top of Figure 5 to estimate the spherical harmonics term c_l^m in (Equation 5). It turns out that we can easily handle dynamic or deforming objects with changing topologies with point cloud representations as shown in Figure 7.

3.5 Geometric Deep Learning and Shape Representations

There is considerable recent work on generating plausible shape representations for 3D data, including voxel-based [Meng et al. 2019; Sindagi et al. 2019; Wu et al. 2015; Zhou and Tuzel 2018], point-based [Charles et al. 2017; Li et al. 2018b,a; Monti et al. 2017; Qi et al. 2017; Wang et al. 2019; Yi et al. 2017] and mesh-based [Hanocka et al. 2019] geometric representations. This includes work on shape representation by learning implicit surfaces on point clouds [Smirnov et al. 2019], designing a mesh Laplacian for convolution [Tan et al. 2018], hierarchical graph convolution on meshes [Mo et al. 2019], encoding signed distance functions for surface reconstruction [Park et al. 2019], etc. However, previous methods on point cloud shape representations learn by designing loss functions to constrain surface smoothness on global Cartesian coordinates. Such functions only provide spatial information of each point and lack information about local shape of the surface compared to explicit discretization of the continuous Laplace-Beltrami operator and curvilinear integral [Do Carmo 2016]. Instead, we use point-cloud based learning algorithms, which do not require mesh Laplacians for graph neural networks. This makes our approach applicable to all kind of dynamic objects, including changing topologies. We extend these prior methods to compute a good representation of the ASFs based on point cloud representations, as described in Section 4.2.

4 LEARNING-BASED SOUND SCATTERING

In this section, we present our learning based sound scattering algorithm. Our goal is to design an efficient approach that can

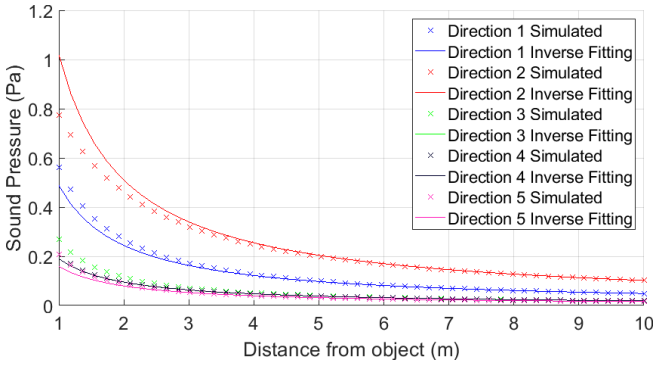

Fig. 3. **Simulated sound pressure fall-off and inverse-distance law fitted curves:** We calculate the sound pressure around a sound scatterer in our dataset using the BEM solver as reference. We examine the sound pressure from $1m$ to $10m$ scattered along 5 directions ($0°, 72°, 144°, 216°$, and $288°$). We regard the sound pressure value at $10m$ to correspond to far-field condition, and inversely fit the pressure values for distance within $10m$ according to Equation 8. We observe that starting from $5m$, all fitted curves closely match with the simulated values. Therefore $r_{ref} = 5m$ is used for generating ASFs for neural network training.

handle any object without assumptions regarding its shape (e.g., convexity) or topology.

4.1 Wave Propagation Modeling

4.1.1 Acoustic Wave Scattering. Equation (3) describes the behavior of acoustic waves in free-field conditions. When a propagating acoustic wave generated by a sound source interacts with an obstacle (the scatterer), a scattered field is generated outside the scatterer. The Helmholtz equation can be used to describe this scenario:

$$(\nabla^2 + k^2)p(\mathbf{x}, \omega) = -Q(\mathbf{x}, \omega), \quad \forall \mathbf{x} \in E, \tag{6}$$

where E is the space that is exterior to the scatterer and $Q(\mathbf{x}, \omega)$ represents the acoustic sources in the frequency domain. Common types of sound sources include monopole sources, dipole sources, and plane wave sources. To obtain an exact solution to (6), the boundary conditions on the scatterer surface S need to be specified. In this work, we assume all the scattering objects are sound-hard (i.e. all energy is scattered, not absorbed) and therefore use the zero Neumann boundary condition for all S:

$$\frac{\partial p}{\partial \mathbf{n}(\mathbf{x})} = 0, \quad \forall \mathbf{x} \in S, \tag{7}$$

where $\mathbf{n}(\mathbf{x})$ is the normal vector at $\mathbf{x}$. Alternatively, other conditions including the sound-soft Dirichlet boundary condition and the mixed Robin boundary condition [Pierce and Beyer 1990] can be used to model different acoustic scattering problems.

4.1.2 Data Generation and Augmentation. We aim to train neural networks that can learn the ASF from an object's shape represented as 3D point clouds. The main challenge in getting the training data is that we need to have a large number of commonly used 3D objects of moderate sizes for our networks to generalize well. To generate our learning examples, we choose to use the *ABC Dataset* [Koch et al. 2019]. This dataset is a collection of one million general Computer-Aided Design (CAD) models and is widely used for evaluation of

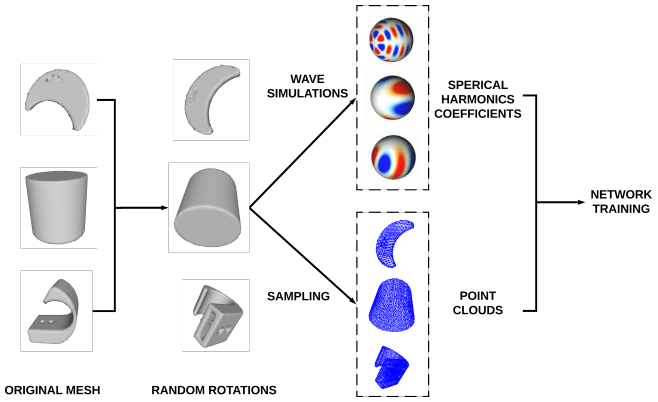

Fig. 4. **Our dataset generation pipeline for neural network training:** Given a set of CAD models, we apply random rotations with respect to their center of mass to generate a larger augmented dataset. We use a BEM solver to calculate the acoustic scattering field of each object assuming a plane wave incident from one direction. The computed fields are compactly represented as spherical harmonics coefficients, as the label for training.

geometric deep learning methods and applications. In particular, this dataset has been used to estimate of differential quantities (e.g., normals) and sharp features, which makes it attractive for learning ASFs as well.

We sample 100,000 models from the *ABC Dataset* and process them by scaling objects such that their longest dimension is in the range of $[1m, 2m]$. The choice of such an object size limit is arbitrary and could depend on the specific problem domain (e.g., size of objects used in applications like games or VR). Because the scattered pressure field is orientation-dependent, we augment our models by applying random 3D rotations to the original dataset to create an equal-sized rotation augmented dataset. To generate accurate labeled data, we use an accurate BEM wave solver, placing a plane wave source with unit strength propagating to the $-x$ direction. The solver outputs the ASF for each object, which becomes our learning target. The dataset pipeline is also illustrated in Figure 4.

4.1.3 Radial Decoupling. Our goal is to determine the scattering field over the exterior space E using a wave-solver. This field needs to be compactly encoded for efficient training. As shown in Equation (5), acoustic wave propagation in the free-field can be decomposed into radial and angular components. Furthermore, the radial sound pressure in the far-field follows the *inverse-distance law* [Beranek and Mellow 2012]: $p \sim 1/r$, as shown in Figure 3. We utilize this property to extrapolate the full ASF from one of its far-field "snapshots" at a fixed radius, so that the full ASF does not need to be stored. Following the inverse-distance law, the sound pressure at any far-field location (r, θ, ϕ) can be computed as

$$p(r, \theta, \phi, \omega) = \frac{r_{ref}}{r} p(r_{ref}, \theta, \phi, \omega), \tag{8}$$

where r_{ref} is the reference distance and only $p(r_{ref}, \cdot, \cdot, \cdot)$ needs to be computed and stored. For brevity, we will omit r in following sections.

4.1.4 Angular Pressure Field Encoding. A spherical field consisting of a fixed number of points (e.g., 642 points evenly distributed on

a sphere surface) is obtained by generating an icosphere with 4 subdivisions. Real valued scattered sound pressures are evaluated at these field points during wave-based simulation. Spherical harmonics (SH) can represent a spherical scalar field compactly using a set of SH coefficients; they have been widely used for 3D sound field recording and reproduction [Poletti 2005]. SH function up to order l_{max} has $M = (l_{max} + 1)^2$ coefficients. The angular pressure at the outgoing direction (θ, ϕ) can be evaluated as

$$p(\theta, \phi, \omega) = \sum_{l=0}^{l_{max}} \sum_{m=-l}^{+l} Y_l^m(\theta, \phi) c_l^m(\omega), \tag{9}$$

where $c_l^m(\omega)$ are the SH coefficients that encode our angular pressure fields. Increasing the number of coefficients can lead to a more challenging learning problem because the dimension of our learning target is raised.

4.2 Learning Spherical Pressure Field

In this section, we present our geometric learning algorithm to learn the angular pressure fields corresponding to ASFs. Our algorithm aims to estimate the SH coefficients for a given point cloud representation of an object.

4.2.1 Local Surface Shape and Implicit Surface Encoder. Previous works on point cloud learning algorithms mostly focus on designing per-point operations [Charles et al. 2017], encoding per-point features to estimate continuous shape functions [Park et al. 2019; Xu et al. 2019], or minimizing loss between a point normal vector and its connected vertices [Liu et al. 2019]. However, high frequency ASFs are affected by fine-grained geometric details. These point-based methods lack a good discretization of the curvilinear integral around a given point and thereby lack sufficient shape details to correctly approximate the ASFs and may not scale well as the simulation frequency increases.

For each point in the input cloud and its neighborhood in the Euclidean space, we assume that it can form a piecewise smooth surface around the point and each point is encoded by the shared multi-layer perceptron (MLP) [Rumelhart et al. 1985] and can be represented by a vector in the higher dimensional latent space (see Figure 5). Thus, a piecewise-linear approximation of the surface around a given point can be used to estimate the local surface shape, where differential coordinates [Do Carmo 2016; Sorkine 2006] (i.e. $\delta-$ coordinates) of each vertex v_i can be expressed as:

$$\overrightarrow{\delta_i} = \frac{1}{d_i} \Sigma_{j \in N(i)} (\overrightarrow{v_i} - \overrightarrow{v_j}). \tag{10}$$

Here δ_i encapsulates the local surface shape, $N(i)$ represents the k nearest neighbors of vertex v_i in the Euclidean space, and $d_i = |N(i)|$ is the number of immediate neighbors of v_i. To estimate the mean curvature of the local surfaces formed by each point and its spatial neighbors, we use the radial basis function $\varphi(\cdot) = \exp^{-||\cdot||^2}$ to weight each vector, rather than using the uniform weight shown in Equation 10. Since there are $N!$ permutations for a point cloud with N vertices, every operation on point clouds should be permutation invariant (i.e. input permutation of points should not change the output of our network). Our weight function is designed to be positive definite and symmetric for any choice of data coordinates.

4.2.2 Implicit Surfaces and Discrete Laplacian. To encapsulate the local surface shape, each point v_i is projected onto higher dimension space using MLPs, and the implicit surface is defined on the latent space z as shown on the top of Figure 5 . For one layer MLP, this is $z_i = relu(\overrightarrow{w} \cdot \overrightarrow{v_i} + b)$, where $\overrightarrow{w}$ and b are learnable parameters in our network. To calculate the $\delta-$ coordinates, the closed simple surface curve around v_i and its immediate neighbors in the Euclidean space (illustrated as green circles with blue outline in Figure 5) is used to evaluate the Equation 11. In the latent space, the local surface shape is encoded as an implicit surface. The direction of the differential coordinate vector, as defined in Equation (11), approximates the local normal direction. Following [Taubin 1995], the discrete Laplacian of implicit surface signal z given by the weighted average over the neighborhoods is represented as:

$$\overrightarrow{\delta_i^{implicit}} = \Sigma_{j \in N_{Euclidean}(i)} \frac{\exp(-||\overrightarrow{v_i} - \overrightarrow{v_j}||)(\overrightarrow{z_i} - \overrightarrow{z_j})}{\Sigma_{j \in N_{Euclidean}(i)} \exp(-||\overrightarrow{v_i} - \overrightarrow{v_j}||)} \tag{11}$$

To compare the two $\delta-$ coordinate representations, we highlight the dB error between the pressure fields reconstructed from groundtruth spherical harmonics term and the predicted ones using different neural networks in Table 3. We observe that $\delta^{implicit}-$ coordinates result in lower loss. This signals that our formulation provides a good approximation of ASFs.

4.2.3 Neural Network Design. Our neural network takes the point cloud as an $N \times 3$ input where N represent the number of points in the point cloud. The output is the Spherical Harmonic coefficients with length 16. For each point marked as red circle with blue outline in Figure 5, four layers of shared-MLP are applied to encode the implicit surface as demonstrated in the top left of Figure 5. Moreover, its k neighbors in the Euclidean space, marked as green circle with blue outline, together with the center point, marked as red circle, are fed into the implicit surface encoder and forming a higher dimensional representation of the center point in the latent space. Next, the discrete Laplacian defined in Equation (12) is evaluated to estimate the implicit surface in the closed simple surface curve around the given point in the latent space, marked yellow in Figure 5. The final representation of the center point, illustrated in the left bottom of Figure 5, for point z_i (marked as a red circle with yellow outline), is defined as:

$$\begin{aligned} feature(v_i) = \Bigg(& z_i^T, \frac{\varphi(\overrightarrow{v_i} - \overrightarrow{v_0})(\overrightarrow{z_i} - \overrightarrow{z_0})}{\Sigma_{j \in N_{Euclidean}(i)} \exp(-||\overrightarrow{v_i} - \overrightarrow{v_j}||)}^T, \\ & \frac{\varphi(\overrightarrow{v_i} - \overrightarrow{v_1})(\overrightarrow{z_i} - \overrightarrow{z_1})}{\Sigma_{j \in N_{Euclidean}(i)} \varphi(\overrightarrow{v_i} - \overrightarrow{v_j})}^T, \ldots, \frac{\varphi(\overrightarrow{v_i} - \overrightarrow{v_n})(\overrightarrow{z_i} - \overrightarrow{z_n})}{\Sigma_{j \in N_{Euclidean}(i)} \varphi(\overrightarrow{v_i} - \overrightarrow{v_j})}^T \\ & \frac{\varphi(\overrightarrow{z_i} - \overrightarrow{z_0})(\overrightarrow{z_i} - \overrightarrow{z_0})}{\Sigma_{j \in N_{latent}(i)} \varphi(\overrightarrow{z_i} - \overrightarrow{z_j})}^T, \ldots, \frac{\varphi(\overrightarrow{z_i} - \overrightarrow{z_n})(\overrightarrow{z_i} - \overrightarrow{z_n})}{\Sigma_{j \in N_{latent}(i)} \varphi(\overrightarrow{z_i} - \overrightarrow{z_j})}^T \Bigg) \end{aligned} \tag{12}$$

The shape of the final per-point feature representation for point v_i is $(1 + 2k, 128)$, where $k = 5$ is the number of the nearest neighbors and 128 is the dimension of the latent space. The implicit surface representation, as in Equation (12), is further fed into the MLP, forming the differential coordinates in Equation (11), and global pooling is applied to extract the global features. The global features regress the predicted spherical harmonic term c_l^m (Equation 5) using fully connected layers.

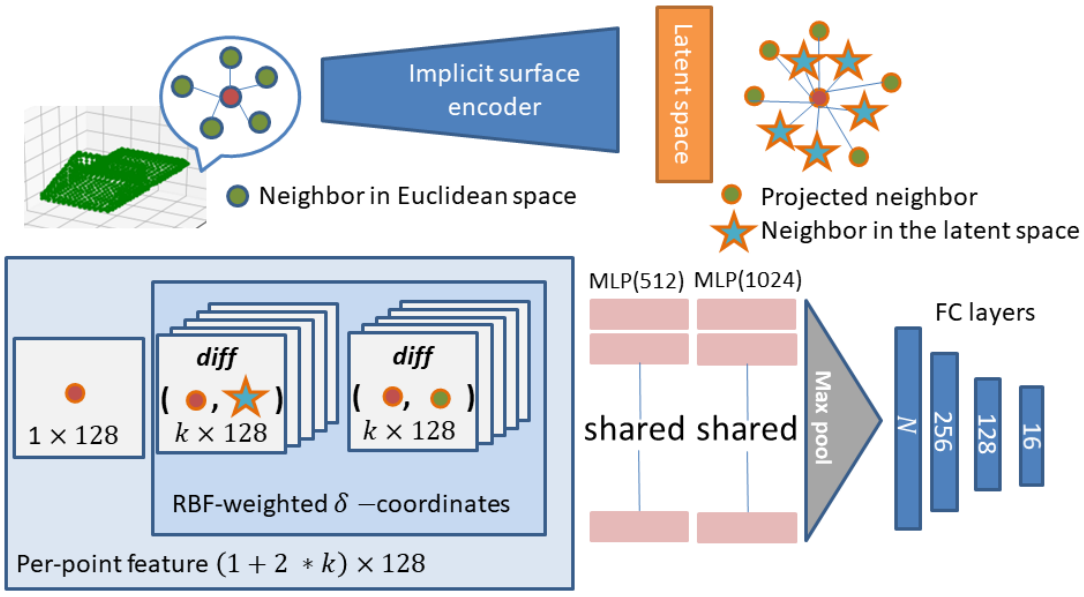

Fig. 5. **Architecture of our point cloud regression network :** The input of our neural network is a $N \times 3$ point cloud and the output of our network is the spherical harmonic coefficients as a vector of length 16. For each point, marked as red circle with blue outline, we incorporate $\delta-$ coordinates to learn the local shape around the point through a encoder. Next, we continue to find the local shape around the point, marked as red circle with yellow outline. Finally we incorporate the $\delta-$ coordinates both in the latent space and in the Euclidean space to represent the point feature of the given point. By applying 2 MLP layers, we leverage the geometric details and output the predicted SH coefficients.

5 INTERACTIVE SOUND PROPAGATION WITH WAVE-RAY COUPLING

In this section, we describe how our learning-based method can be combined with geometric sound propagation techniques to compute the impulse responses for given source and listener positions. Then, we render them.

Hybrid Sound Propagation. We use a hybrid sound propagation algorithm that combines wave-based and ray acoustics. Each of them handles different parts of wave acoustics phenomena, but they are coupled in terms of incoming and outgoing energies at multiple localized scattering fields. Specifically, our trained neural network estimates the scattering field and is used to compute propagation paths when sound interacts with obstacles in the scene. On the other hand, modeling sound propagation in the air along with specular and diffuse reflections at large boundary surfaces (e.g., walls, floors) is computed using ray and path tracing methods [Schissler et al. 2014].

Ray Tracing with Localized Fields. Our localized ASFs are represented using SH coefficients. Given the most general ray tracing formulation at a scattering surface, the sound intensity I_{out} of an outgoing direction (θ_o, ϕ_o) from a scattering surface is given by the integral of the incoming intensity from all directions:

$$I_{out}(\theta_o, \phi_o, \omega) = \int_S I_{in}(\theta_i, \phi_i, \omega) f(\theta_i, \phi_i, \theta_o, \phi_o, \omega) dS, \qquad (13)$$

where S represents the directions on a spherical surface around the ray hit point, $I_{in}(\theta_i, \phi_i, \omega)$ is the incoming sound intensity from direction (θ_i, ϕ_i), and $f(\theta_i, \phi_i, \theta_o, \phi_o, \omega)$ is the bi-directional scattering distribution function (BSDF) that is commonly used in visual rendering [Pharr et al. 2016]. Our problem of acoustic wave scattering is different from visual rendering in two aspects: (1) sound wave scatters around objects, whereas light mostly transmits to visible directions or propagates through transparent materials; (2) BSDFs are point-based functions that depend on both incoming and outgoing directions, whereas our localized scattered fields are region-based functions, as shown in Figure 8. Therefore, we replace BSDFs in Equation (13) with our localized scattered field $p(\theta, \phi, \omega)$ representation from Equation (9). Our choice of a spherical offset surface to model the scattered field also enables us to perform integration over the whole spherical surface in a straightforward manner, since evaluating spherical coordinates is efficient with SH functions. Although $p(\theta, \phi, \omega)$ encodes only the outgoing directions and assumes incoming plane waves to $-x$ direction, one can easily rotate the point cloud to align any other incoming direction to the $-x$ direction and then use our network to infer $p(\theta, \phi, \omega)$ for that direction. We update Equation (13) to

$$I_{out}(\theta_o, \phi_o, \omega) = \int_S I_{in}(\theta_i, \phi_i, \omega) p^2(\theta_i, \phi_i, \omega) dS. \qquad (14)$$

We use the Monte Carlo integration to numerically evaluate the outgoing scattered intensity:

$$I_{out}(\theta_o, \phi_o, \omega) \approx \frac{1}{N} \sum_{j=1}^{N} \frac{I_{in}(\theta_j, \phi_j, \omega) p^2(\theta_j, \phi_j, \omega)}{Pr(\theta_j, \phi_j)}, \qquad (15)$$

where N is the number of samples and $Pr(\theta_j, \phi_j)$ is the probability of generating a sample for direction (θ_j, ϕ_j). A uniform sampling over

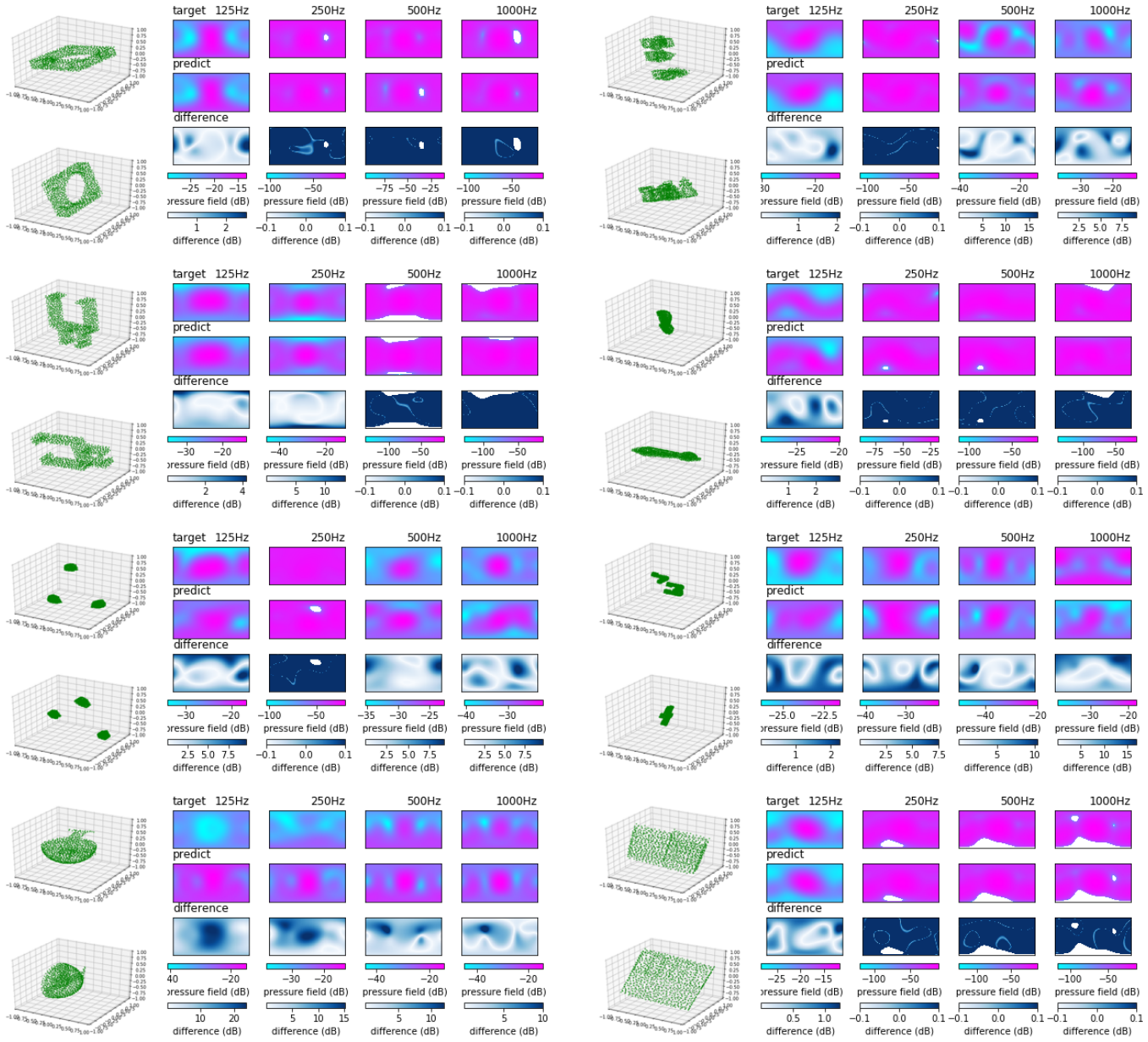

Fig. 6. **Comparing Acoustic Scattering Fields:** We illustrate the acoustic scattering fields for different objects (shown in separate rows) and four different simulation frequencies (shown in separate columns). These objects are not seen during training and their shape varies based on convexity and topology (e.g., genus, connected components) characteristics, as shown on the left. For each image block, the left column shows two different views of an object; the top row (*target*) is the groundtruth ASF computed using a BEM solver on the original mesh (takes about a few minutes); the middle row (*predict*) represents the ASF computed using our neural network based on point-cloud representation (takes $\sim 1ms$ on a GPU). The bottom row (*difference*) highlights the difference between the groundtruth and our prediction, using a separate colorbar. We see a close match for most cases and these results demonstrate that our learned scattering fields are a good approximation of those computed using an accurate wave-solver. More visualization results are shown in the supplementary file.

the sphere surface gives $Pr(\theta_j, \phi_j) = \frac{1}{4\pi}$. In theory any probability distribution can be used. As N increases, the approximation becomes more accurate.

6 IMPLEMENTATION AND RESULTS

In this section, we describe our implementation and highlight the performance on many dynamic benchmarks.

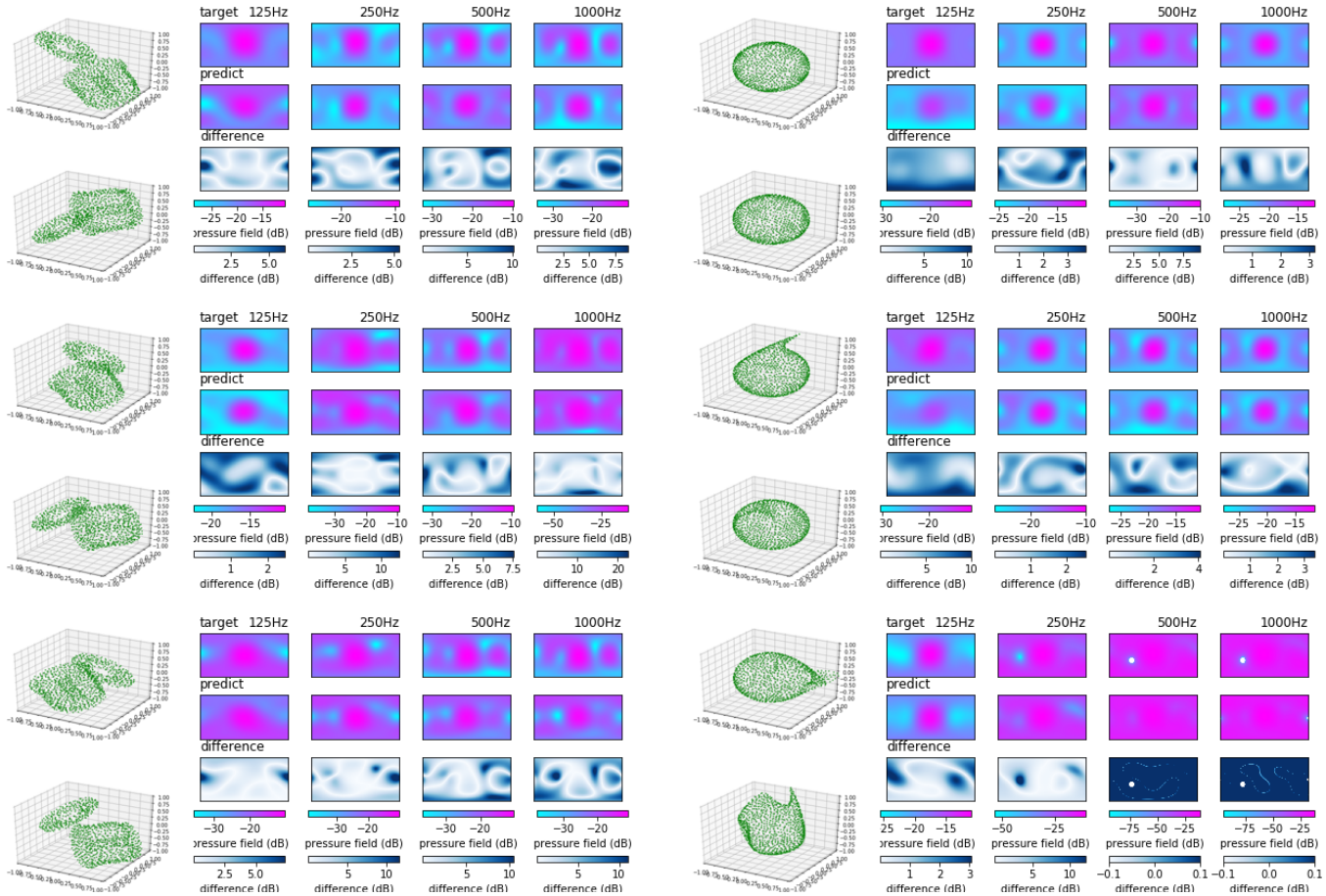

Fig. 7. **Moving and Deforming Objects:** The left column represents two objects moving close to each other. When the objects are very close, our approach treats them as one unified object and computes its point cloud representation. The right column shows different frames of a deforming sphere. The training dataset (from *ABC Dataset*) does not contain deforming objects nor moving objects. Our algorithm generates good approximations to ASFs for such dynamic objects, as we compare with the exact BEM solver.

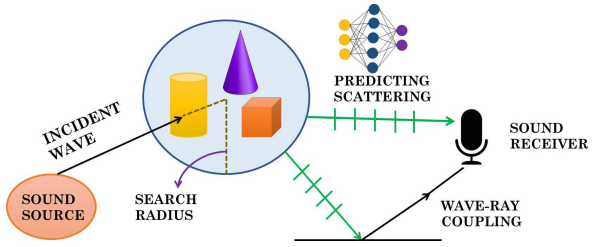

Fig. 8. **Hybrid Sound Propagation:** When a far-field wave intersects a scattering object, we construct a point cloud containing surrounding objects within a pre-defined search radius (shown as a circle). The outgoing sound direction is sampled over the whole spherical surface and the scattered energy in each direction is computed by the acoustic scattered field generated by our trained neural network. This can handle diffraction and occlusion effects, while the ray tracer also computes specular and diffuse reflections.

6.1 Parameters

Our algorithm involves various parameters. In this section, we explain the choice of those parameters and their impacts on our implementation.

Mesh Pre-processing. The original meshes from the *ABC Dataset* have high levels of details with fine edges of length shorter than $1cm$. Dense point cloud inputs could also be modeled or collected from the real-world scenes with granularity similar to this dataset. However, a high number of triangle elements in a mesh will significantly increase the simulation time of BEM solvers. For wave-based solver, our highest simulation frequency is $1000Hz$, which converts to a wavelength of $34cm$. Therefore, we use a combination of mesh simplification and mesh clustering algorithm to ensure that our meshes have a minimum edge length of $1.7cm$, which is $1/20$ of our shortest target wavelength. This is sufficient according to the standard techniques used in BEM simulators [Marburg 2002]. Most meshes after pre-processing have fewer than 20% number of elements than the original and the BEM simulation for dataset generation gains over $10\times$ speedup.

Reference Field Distance. Since the inverse-distance law does not hold in the near-field of objects, we need to find a suitable distance for computing our reference field. We experimentally simulate the sound pressure fall-off with respect to distance and observe that

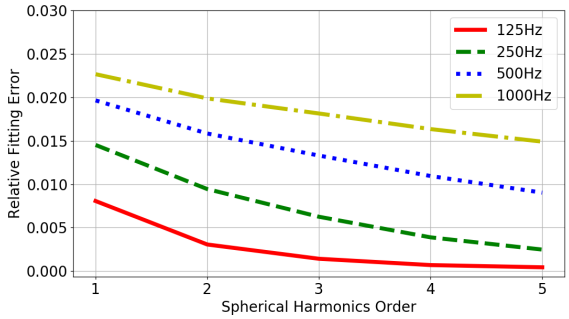

Fig. 9. **Spherical harmonics approximation of sound pressure fields:** We evaluate different orders of SH functions to fit our pressure fields at 4 frequencies and calculate the relative average fitting errors. We observe that high-frequency pressure fields result in larger fitting errors, as compared low-frequency pressure fields. Therefore, we use learned scattered fields for lower frequencies (i.e., $\leq 1000Hz$) during our training phase. We can handle higher frequencies by increasing the SH order, though that would increase the training time.

sound pressure that is $5m$ or further away from the scatterer closely agrees with this far-field approximation (see Figure 3). Therefore, we choose to calculate the pressure field on an offset surface $5m$ away from the scatterer's center using a BEM solver (i.e., setting $r_{ref} = 5m$ in Equation 8). Note that this choice of $5m$ is not strict or fixed. If higher accuracy along the radial line is desired, multiple locations (especially in the near field) can be sampled during the simulation to interpolate the curve at a higher accuracy. The precomputation time and memory overhead will increase linearly w.r.t the number of sampled distance fields.

Max Spherical Harmonics Order. We experiment with the number of SH coefficients by projecting our scattered sound pressure fields to SH functions with different orders, as shown in Figure 9. Based on this analysis, we choose to use up to a 3rd order SH projection, which yields sufficiently small fitting errors (relative error smaller than 2%) with 16 SH coefficients. This also sets the output of our neural network (Section 4.2.3) to be a vector of length 16.

6.2 Wave-Solver and Training

We use the *FastBEM Acoustics* software [1] as our wave-based solver. Simulations are run on a Windows 10 workstation that has 32 Intel(R) Xeon(R) Gold 5218 CPUs with multi-threading. In order to accelerate the overall computation, we use two different versions of BEM solvers. First we use the adaptive cross approximation (ACA) BEM [Kurz et al. 2002] to compute the ASF since it can achieve near $O(N)$ computational performance for small to medium sized models (e.g., element count $N \leq 100,000$). If this solver fails to converge within some fixed number of iterations, we use the conventional and accurate BEM solver. Overall, it takes about 12 days to compute the ASF up to $1000Hz$ frequency of about 100,000 objects from the *ABC Dataset*. The sound pressure field is evaluated at 642 field points that are evenly distributed on the spherical field surface. Next, we use the *pyshtools* [2] software [Wieczorek and Meschede 2018] to compute the spherical harmonics coefficients from the pressure field using least squares inversion.

Training Settings. Each network model is trained on a GeForce RTX 2080 Ti GPU using the *Tensorflow* framework [Abadi et al. 2016]. The dataset is split into training set and test set using the ratio 9 : 1. In the training stage, we use Adam optimizer to minimize L_2 norm loss between predicted spherical harmonic coefficients and the groundtruth. In practice, the learning rate is set to 1×10^{-3} and decays exponentially. The batch size is set to 128 and typically our network converges after 100 epochs. The number of our trainable parameters is 267k, and the number of parameters in PointNet is about 0.8M.

6.3 Runtime System and Benchmarks

We use the geometric sound propagation and rendering algorithm described in [Schissler et al. 2014]. Our sound rendering system traces sound rays at octave frequency bands at $125Hz$, $250Hz$, $500Hz$, $1000Hz$, $2000Hz$, $4000Hz$, and $8000Hz$. The direct output from ray tracing for each frequency band is the energy histogram with respect to propagation delays. We take square root of these responses to compute the frequency dependent pressure response envelopes. Broadband frequency responses are interpolated from our traced frequency bands, and the inverse Fourier transform is used to re-construct the broadband impulse response. Our method does not preserve phase information, so a random phase spectrum is used during the inverse Fourier transform. In practice, this random phase spectrum does not introduce noticeable sound difference [Kuttruff 1993].

We require that the wall boundaries are explicitly marked in our scenes. As a result, when a ray hits the wall, only conventional sound reflections occur for all frequencies. During audio-visual rendering, when a ray hits a scattering object, we first extend the hit point along its ray direction by $0.5m$ and use it as the scattering region center. We include all the points within a search radius of $1m$ from the region center to generate a point cloud approximation of the scatterer. This point cloud is resampled using furthest point sampling and fed into our neural networks. Our network predicts the ASFs for sound frequencies corresponding to $125Hz$, $250Hz$, $500Hz$ and $1000Hz$. The higher frequencies (i.e., $2000Hz$, $4000Hz$, and $8000Hz$) are handled by conventional geometric ray-tracing with specular and diffuse reflections and it does not use ASFs. Our neural network implementation is light-weight with only four layers and has small prediction overhead of less than $1ms$ per view on an NVIDIA GeForce RTX 2080 Ti GPU. The interactive runtime propagation system is illustrated in Figure 8. Our ray-tracer performs 200 orders of reflections to generate late reverberation effects.

We evaluate the performance of our hybrid sound propagation and rendering algorithms on the following benchmark scenes. They have with varying levels of dynamism in terms of moving objects and are highlighted in the supplemental video. The runtime performance of audio rendering is highlighted in Table 2.

Floor: The floor scene demonstrates the validity of our sound rendering in a simplest scenario containing only one static sound scatterer and a static sound source above an infinitely large floor. The listener moves horizontally so that the sound source

[1] https://www.fastbem.com/
[2] https://shtools.oca.eu/shtools/public/index.html

visibility changes periodically. We use this benchmark to evaluate the correctness of our approach as the listener enters or leaves the occluded region, and compare our results with groundtruth BEM simulation.

Sibenik: This scene consists of two moving disjoint objects, considered as scatterers. The two scatterers revolve around each other in close proximity such that there are complicated near-field interactions of sound waves. Prior techniques for dynamic scenes [Rungta et al. 2018] cannot handle such scenarios as the objects are not well-separated and their near-fields overlaps. In our approach, when two objects are in close proximity or touching, we treat them as one large object or a composite scatterer and compute its point cloud approximation. We use our neural network to compute the ASF for this composite scatterer.

Trinity: This benchmark showcases the ability of our method to handle scenarios with a large number of moving objects. In this scenario, many objects fly across the room and dynamically generate new composite scatterers or decompose them into separate scatterers. As a result, the total number of disjoint objects in the scene change. Moreover, the occluded regions in the scene also change dynamically and create challenging scenarios in terms of sound propagation. Our approach can still generate smooth audios despite highly dynamic nature of the scene.

Havana: This benchmark includes two moving walls that are generally larger than scatterers in previous benchmarks. We use this benchmark to show that our approach can also handle large static objects, in addition to a large number of dynamic objects.

Scene	#Vert	#Scatterers	Frame time
Floor	2037	1	10.65ms
Sibenik	46880	2	6.87ms
Trinity	138916	6	12.95ms
Havana	28542	2	6.78ms

Table 2. Runtime performance on our benchmarks. The computation of ASFs takes $\leq 1ms$ per view and most of the time is spent in ray tracing.

6.4 Analysis

Ablation Study and Comparisons. We perform ablation studies with our network design and summarize the results in Table 3. We justify the design of our network with this ablation study, including the use of $\delta-$ coordinates, RBF-weighted function as well as the implicit surface encoder, as highlighted in Figure 5. We use PointNet [Charles et al. 2017] as the baseline, where only per-point MLP layers are applied on each point in the point cloud. RBF-weighted $\delta-$ coordinates (as described in Equation 11), uniform-weighted $\delta-$ coordinates (described in Equation 10), implicit surface encoder shown in the top of Figure 5 are considered as subjects of ablation studies. We observe that our fine-grained geometric feature representation in Equation 12 results in larger reduction in dB error as compared to PointNet [Charles et al. 2017]. In general, the first four experiments on the $125Hz$ test dataset do not show significant differences. However, for $\{500Hz, 1000Hz\}$ test cases, our implicit surface encoder improves the performance by approx. 0.4 dB.

{RBF-weighted δ-coord | implicit surface}	dB Error { 125Hz , 250Hz , 500Hz , 1000Hz }			
{✗| ✗}	3.49	3.56	3.71	4.23
{✓| ✗}	3.38	3.41	3.57	4.47
{✗| ✓}	3.28	**3.38**	3.52	3.85
{✓| ✓} (ours)	**3.23**	3.44	**3.47**	**3.80**
PointNet [Charles et al. 2017]	3.96	4.42	3.89	4.43

Table 3. Ablation study: In this evaluation, we compare the performance on uniform $\delta-$ coordinate in Eq. (10), weighted $\delta-$ coordinates in Eq. (11) and implicit surface estimation in Eq. (12) on our test dataset(including $10k$ objects) at frequency bands {125Hz, 250Hz, 500Hz, 1000Hz}. The best result for each frequency is highlighted in **bold** (lower error is better). We alter between choosing Eq. (10) and Eq. (11) that results in four different combinations: Row 1 and 2, 3 an 4. Next, we experiment the use of implicit surface encoder (Row 3 and 4). Our proposed network design (Row 4) highlights superior performance in terms of ASF approximation for most frequencies. A lower value indicates a better result.

Evaluation. Our goal is to approximate the acoustic scattering fields of general 3D objects. While there is a preliminary 2D scattering dataset [Fan et al. 2020], there are no general or well-known datasets or benchmarks for evaluating such ASFs or related computations. Therefore, we use $10k$ objects from our test dataset to evaluate the performance of our trained network in terms of accuracy. Compared with the original *ABC Dataset*, our test dataset has been augmented in terms of scale and using different orientations to evaluate the performance of our learning method. We analyze the numerical accuracy of our method by comparing our neural network predictions with groundtruth ASFs generated by the BEM solver. All objects used to evaluate the accuracy have not been seen in the training set. We highlight the acoustic pressure fields computed using the accurate BEM solver (i.e., the groundtruth) along with the ones estimated using our network for different objects in Figure 6. We also highlight many configurations of objects in close proximity, such that their near fields overlap. Our approach treats such almost touching objects as one composite object, computes its point cloud approximation, and estimates the ASF using the neural network. Even for such challenging scenarios, our learned scattering fields closely matches the ones computed using the BEM solver. These example highlights that our learning approach can generalize to unseen objects and approximate the ASFs with good accuracy.

Frequency Growth. In theory, our learning-based framework and runtime system can also incorporate wave frequencies beyond $1000Hz$. However, two important factors need to be considered when extending our setup: 1) the wave simulation time increases with the simulation frequency (e.g., between a square and cubic function for an accurate BEM solver); and 2) the ASF becomes more complicated at higher frequencies, which makes it more difficult to be learned or approximated using the same neural network. We highlight these observations quantitatively in Figure 10. Note that the simulation time is governed much by the choice of the wave solver, as well as the relevant parameters/strategies used. We preprocessed our meshes according to the highest simulation frequency (i.e., the one with the shortest wavelength) and used that mesh representation for all frequencies. When a higher frequency needs to be added, the meshes need to have finer details, meaning more boundary elements will be involved (e.g., at least four times more elements when the simulation frequency doubles). A frequency-adaptive mesh simplification strategy [Li et al. 2015] can be used to

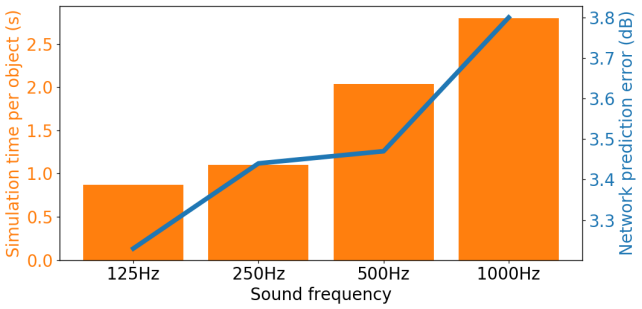

Fig. 10. **Simulation time and network prediction errors w.r.t frequency growth:** We highlight the cost of adding higher frequencies to our pipeline. *Left y-axis and bar plot*: the average simulation time for each object when parallelized with 8 threads; *Right y-axis and line plot*: our average network prediction error of ASFs.

reduce the simulation time at low frequencies. Our network prediction error also grows with the target frequency, but not at a prohibitive rate. We can reduce this error by using more training examples and more sophisticated neural network designs.

7 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

We present a new learning-based approach to approximate the acoustic scattering fields of objects for interactive sound propagation. We exploit properties of the acoustic scattering field and present a novel geometric learning algorithm that uses a point-based approximation and the local shapes are encoded using implicit surfaces. We use a four-layer neural network that computes a field representation using 3rd order spherical harmonics. We use a large training database of 100,000 objects, along with random 3D orientations and scaling of each object, and generate the accurate labeled data with a BEM solver. We evaluated the accuracy of our method on a large number of objects not present in the training dataset and the initial results are promising. Furthermore, we combine with a ray-tracing based sound propagation algorithm for sound rendering in highly dynamic scenes. Our approach is general, has low additional runtime overhead on top of ray tracing, and can handle diffraction effects and occluded regions for interactive applications.

Our approach has several limitations. These include all the challenges of geometric deep learning in terms of choosing an appropriate training dataset and large training time. Furthermore, we assume that objects in the scene are sound-hard and do not take into account various material properties. Our four-layer network has been tested for frequencies up to $1000Hz$, and we may need to design better learning methods for higher frequencies. The overall accuracy of our hybrid propagation algorithm lies between a pure geometric (ray-tracing) method and a global numeric solver. There is a linear scaling of training time with the number of frequencies and the number of scattering objects, while the simulation time could scale as a cubic function of the frequency. As a result, the precomputation overhead can be high. One way to overcome is to limit the training to the kind or class of objects that are frequently used in an interactive application (e.g., a game or VR scenario). This is equivalent to performing customized training for a specific scenario. There are many avenues for future work. In addition to overcoming these limitations, we need to evaluate its performance in other scenarios and integrate with different applications. It would be useful to take into account the material properties by considering them as an additional object characteristic in our training database. We would also like to use other techniques from geometric processing and geometric deep learning to improve the performance of our approach, e.g., using mean curvature vectors over linear approximations.

REFERENCES

2018. Steam Audio. https://valvesoftware.github.io/steam-audio.

2019. Microsoft Project Acoustics. https://aka.ms/acoustics.

2019. Oculus Spatializer. https://developer.oculus.com/downloads/package/oculus-spatializer-unity.

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

Leo Leroy Beranek and Tim Mellow. 2012. *Acoustics: sound fields and transducers*. Academic Press.

Dick Botteldooren. 1995. Finite-difference time-domain simulation of low-frequency room acoustic problems. *The Journal of the Acoustical Society of America* 98, 6 (1995), 3302–3308.

Chakravarty R Alla Chaitanya, John M Snyder, Keith Godin, Derek Nowrouzezahrai, and Nikunj Raghuvanshi. 2019. Adaptive Sampling for Sound Propagation. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 1846–1854.

R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Jul 2017). https://doi.org/10.1109/cvpr.2017.16

Manfredo P Do Carmo. 2016. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications.

James Eaton, Nikolay D Gaubitch, Alastair H Moore, Patrick A Naylor, James Eaton, Nikolay D Gaubitch, Alastair H Moore, Patrick A Naylor, Nikolay D Gaubitch, James Eaton, et al. 2016. Estimation of room acoustic parameters: The ACE challenge. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 24, 10 (2016), 1681–1693.

Ziqi Fan, Vibhav Vineet, Hannes Gamper, and Nikunj Raghuvanshi. 2020. Fast acoustic scattering using convolutional neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 171–175.

Eric L Ferguson, Stefan B Williams, and Craig T Jin. 2018. Sound source localization in a multipath environment using convolutional neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2386–2390.

Thomas Funkhouser, Ingrid Carlbom, Gary Elko, Gopal Pingali, Mohan Sondhi, and Jim West. 1998. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 21–32.

Andrea F Genovese, Hannes Gamper, Ville Pulkki, Nikunj Raghuvanshi, and Ivan J Tashev. 2019. Blind Room Volume Estimation from Single-channel Noisy Speech. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 231–235.

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. MeshCNN: a network with an edge. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

Doug L James, Jernej Barbič, and Dinesh K Pai. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM Transactions on Graphics (TOG)*, Vol. 25. ACM, 987–995.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. ABC: A Big CAD Model Dataset For Geometric Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Stefan Kurz, Oliver Rain, and Sergej Rjasanow. 2002. The adaptive cross-approximation technique for the 3D boundary-element method. *IEEE transactions on Magnetics* 38, 2 (2002), 421–424.

K Heinrich Kuttruff. 1993. Auralization of impulse responses modeled on the basis of ray-tracing results. *Journal of the Audio Engineering Society* 41, 11 (1993), 876–880.

Christian Lauterbach, Anish Chandak, and Dinesh Manocha. 2007. Interactive sound rendering in complex and dynamic scenes using frustum tracing. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1672–1679.

Dingzeyu Li, Yun Fei, and Changxi Zheng. 2015. Interactive acoustic transfer approximation for modal sound. *ACM Transactions on Graphics (TOG)* 35, 1 (2015), 1–16.

Jiaxin Li, Ben M. Chen, and Gim Hee Lee. 2018b. SO-Net: Self-Organizing Network for Point Cloud Analysis. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Jun 2018). https://doi.org/10.1109/cvpr.2018.00979

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2018a. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*. 820–830.

Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. 2019. Learning to infer implicit surfaces without 3d supervision. In *Advances in Neural Information Processing Systems*. 8293–8304.

Steffen Marburg. 2002. Six boundary elements per wavelength: Is that enough? *Journal of computational acoustics* 10, 01 (2002), 25–51.

Ravish Mehra, Nikunj Raghuvanshi, Lakulish Antani, Anish Chandak, Sean Curtis, and Dinesh Manocha. 2013. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Transactions on Graphics (TOG)* 32, 2 (2013), 19.

Ravish Mehra, Atul Rungta, Abhinav Golas, Ming Lin, and Dinesh Manocha. 2015. WAVE: Interactive wave-based sound propagation for virtual environments. *IEEE transactions on visualization and computer graphics* 21, 4 (2015), 434–442.

Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. 2019. VV-Net: Voxel vae net with group convolutions for point cloud segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 8500–8508.

Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J. Mitra, and Leonidas J. Guibas. 2019. StructureNet. *ACM Transactions on Graphics (TOG)* 38, 6 (Nov 2019), 1–19. https://doi.org/10.1145/3355089.3356527

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5115–5124.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 165–174.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically based rendering: From theory to implementation*. Morgan Kaufmann.

Allan D Pierce and Robert T Beyer. 1990. Acoustics: An introduction to its physical principles and applications. 1989 Edition.

Mark A Poletti. 2005. Three-dimensional surround sound systems based on spherical harmonics. *Journal of the Audio Engineering Society* 53, 11 (2005), 1004–1025.

Ville Pulkki and U Peter Svensson. 2019. Machine-learning-based estimation and rendering of scattering in virtual reality. *The Journal of the Acoustical Society of America* 145, 4 (2019), 2664–2676.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*. 5099–5108.

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

Nikunj Raghuvanshi and John Snyder. 2014. Parametric wave field coding for precomputed sound propagation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 38.

Nikunj Raghuvanshi and John Snyder. 2018. Parametric directional coding for precomputed sound propagation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 108.

Nikunj Raghuvanshi, John Snyder, Ravish Mehra, Ming Lin, and Naga Govindaraju. 2010. Precomputed Wave Simulation for Real-time Sound Propagation of Dynamic Sources in Complex Scenes. *ACM Trans. Graph.* 29, 4, Article 68 (July 2010), 11 pages. http://doi.acm.org/10.1145/1778765.1778805

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation*. Technical Report. California Univ San Diego La Jolla Inst for Cognitive Science.

Atul Rungta, Carl Schissler, Nicholas Rewkowski, Ravish Mehra, and Dinesh Manocha. 2018. Diffraction Kernels for Interactive Sound Propagation in Dynamic Environments. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1613–1622.

Lauri Savioja and U Peter Svensson. 2015. Overview of geometrical room acoustic modeling techniques. *The Journal of the Acoustical Society of America* 138, 2 (2015), 708–730.

Carl Schissler and Dinesh Manocha. 2017. Interactive sound propagation and rendering for large multi-source scenes. *ACM Transactions on Graphics (TOG)* 36, 1 (2017), 2.

Carl Schissler and Dinesh Manocha. 2018. Interactive Sound Rendering on Mobile Devices using Ray-Parameterized Reverberation Filters. *arXiv preprint arXiv:1803.00430* (2018).

Carl Schissler, Ravish Mehra, and Dinesh Manocha. 2014. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 39.

Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. 2019. MVX-Net: Multimodal VoxelNet for 3D Object Detection. *2019 International Conference on Robotics and Automation (ICRA)* (May 2019). https://doi.org/10.1109/icra.2019.8794195

Dmitriy Smirnov, Matthew Fisher, Vladimir G. Kim, Richard Zhang, and Justin Solomon. 2019. Deep Parametric Shape Predictions using Distance Fields. arXiv:1904.08921 [cs.GR]

Olga Sorkine. 2006. Differential representations for mesh processing. In *Computer Graphics Forum*, Vol. 25. Wiley Online Library, 789–807.

U Peter Svensson, Roger I Fred, and John Vanderkooy. 1999. An analytic secondary source model of edge diffraction impulse responses. *The Journal of the Acoustical Society of America* 106, 5 (1999), 2331–2344.

Qingyang Tan, Lin Gao, Yu-Kun Lai, Jie Yang, and Shihong Xia. 2018. Mesh-based autoencoders for localized deformation component analysis. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhenyu Tang, Nicholas J Bryan, Dingzeyu Li, Timothy R Langlois, and Dinesh Manocha. 2020. Scene-Aware Audio Rendering via Deep Acoustic Analysis. *IEEE Transactions on Visualization and Computer Graphics* (2020).

Gabriel Taubin. 1995. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 351–358.

Micah Taylor, Anish Chandak, Qi Mo, Christian Lauterbach, Carl Schissler, and Dinesh Manocha. 2012. Guided Multiview Ray Tracing for Fast Auralization. *IEEE Transactions on Visualization and Computer Graphics* 18 (2012), 1797–1810.

Roberto A Tenenbaum, Filipe O Taminaro, and VS Melo. 2019. Room acoustics modeling using a hybrid method with fast auralization with artificial neural network techniques. In *Proc. International Congress on Acoustics (ICA)*. 6420–6427.

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

Nicolas Tsingos. 2009. Precomputing geometry-based reverberation effects for games. In *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society.

Nicolas Tsingos, Thomas Funkhouser, Addy Ngan, and Ingrid Carlbom. 2001. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 545–552.

Dimitrios Tsokaktsidis, Timo Von Wysocki, Frank Gauterin, and Steffen Marburg. 2019. Artificial Neural Network predicts noise transfer as a function of excitation and geometry. In *Proc. International Congress on Acoustics (ICA)*. 4392–4396.

Vesa Valimaki, Julian D Parker, Lauri Savioja, Julius O Smith, and Jonathan S Abel. 2012. Fifty years of artificial reverberation. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 5 (2012), 1421–1448.

Michael Vorländer. 1989. 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 (1989), 172–178.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics* 38, 5 (Oct 2019), 1–12. https://doi.org/10.1145/3326362

Mark A Wieczorek and Matthias Meschede. 2018. Shtools: Tools for working with spherical harmonics. *Geochemistry, Geophysics, Geosystems* 19, 8 (2018), 2574–2592.

Luiz C Wrobel and AJ Kassab. 2003. Boundary element method, volume 1: Applications in thermo-fluids and acoustics. *Appl. Mech. Rev.* 56, 2 (2003), B17–B17.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1912–1920.

Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 2019. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*. 490–500.

Hengchin Yeh, Ravish Mehra, Zhimin Ren, Lakulish Antani, Dinesh Manocha, and Ming Lin. 2013. Wave-ray coupling for interactive sound propagation in large complex scenes. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 165.

Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. 2017. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2282–2290.

Xiaopeng Zheng, Chengfeng Wen, Na Lei, Ming Ma, and Xianfeng Gu. 2017. Surface Registration via Foliation. In *The IEEE International Conference on Computer Vision (ICCV)*.

Yin Zhou and Oncel Tuzel. 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Jun 2018). https://doi.org/10.1109/cvpr.2018.00472