Room Acoustical Parameter Estimation from Room Impulse Responses Using Deep Neural Networks

Yu, Wangyang; Kleijn, W.Bastiaan

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Room Acoustical Parameter Estimation from Room Impulse Responses Using Deep Neural Networks

Wangyang Yu, *Student Member, IEEE,* W. Bastiaan Kleijn, *Fellow, IEEE*

Abstract—We describe a new method to estimate the geometry of a room and reflection coefficients given room impulse responses. The method utilizes convolutional neural networks to estimate the room geometry and multilayer perceptrons to estimate the reflection coefficients. The mean square error is used as the loss function. In contrast to existing methods, we do not require the knowledge of the relative positions of sources and receivers in the room. The method can be used with only a single RIR between one source and one receiver. For simulated environments, the proposed estimation method can achieve an average of 0.04 m accuracy for each dimension in room geometry estimation and 0.09 accuracy in reflection coefficients. For real-world environments, the room geometry estimation method achieves an accuracy of an average of 0.065 m for each dimension.

Index Terms—Room impulse response, room geometry, reflection coefficient, deep neural network.

I. Introduction

AUGMENTED reality (AR) is an immersive audio-visual environment where artificial objects are added to a real-world scenario, providing the user with an enhanced and interactive experience [1]. Augmented reality will play an increasingly important role in numerous contexts, such as education, manufacturing, and archaeology. An accurate description of acoustic environments is essential for generating perceptually acceptable sound in an AR system. Estimating room acoustical parameters forms an important aspect of modeling an acoustic environment accurately. In this paper, we consider the estimation of the room geometry and reflection coefficients from room impulse responses.

The room impulse response (RIR), the transfer function between the sound source and the listener, characterizes the acoustic environment of a room. It is composed of direct-direction sound, early reflections, and late reverberation. An RIR is affected by the position of the sound source and the receiver, the room geometry, and the reflection coefficients. In the context of this paper, we consider rectangular rooms and define room geometry to be a three-dimensional vector, which contains the length, width, and height of a room. The room geometry and the reflection coefficients can be used to model and analyze acoustic behavior inside a room via RIRs. We are interested in the estimation of the room acoustical parameters from RIRs.

In this paper, we use deep learning to solve this estimation problem. In recent years, deep learning has seen a rapid increase in usage as a result of the increased computational power and the availability of large databases. Relevant deep neural networks (DNNs) to our work are feedforward multi-layer perceptrons (MLPs) and convolutional neural networks (CNNs). MLPs [2] are composed of fully connected layers and can approximate most mapping functions. This property makes them applicable in various areas, such as ecology [3], chemistry [4], and climate change [5]. CNNs contain a set of generalized filters of different levels to extract features from the signals. CNNs have been used for various applications such as image classification [6]–[8], and speech recognition [9]–[11].

We use CNNs for room geometry estimation and MLPs for the estimation of reflection coefficients. CNNs can analyze data with salient spatial structures [12] and we hypothesize that the room geometry defines patterns in RIR signals. Reflection coefficients influence the strength of reflective pulses, which we hypothesize MLPs are able to learn from RIR signals. Due to the limited amount of real-world measured RIRs, we first train the neural network with artificial data. After that, we use transfer learning to make the model work with real-world measured RIRs.

The main contribution of this paper is the usage of deep neural networks to estimate room acoustical parameters. In contrast to state-of-the-art methods for estimating room acoustical parameters, our method only requires a random RIR between a single sound source and a single receiver in the room without any additional information. The new room geometry estimation model performs well with real-world measured RIRs.

This paper is organized as follows. We review the relevant background knowledge in section II. In section III, we formulate the estimation problem of the room acoustical parameters. We then describe the solutions of the room geometry estimation problem and the reflection coefficient estimation problem separately in section IV and section V. The experimental results are discussed and analyzed in detail in section VI. Finally, we conclude our paper in section VII.

II. Background

In this section, we discuss relevant background knowledge of our work. We first describe the image source method, which we use to generate the RIR database to train our base method of room geometry estimation as described in section IV and to estimate reflection coefficients as described in section V. We aim to use MLPs to estimate reflection coefficients and CNNs to estimate room geometry. Consequently, we discuss room acoustical parameters estimation, multilayer perceptrons, and convolutional neural networks in this section.

A. The image source method

The image source method [13]–[16], which was first proposed by Allen and Berkley [16] in 1979, is commonly used to model RIRs in empty and rectangular rooms. It assumes the sound only propagates along straight rays. The method is computationally efficient, which makes it suitable to generate a large scale database. In three-dimensional (3D) space, we denote the position of the receiver as (x_r, y_r, z_r) and the position of the source as (x_s, y_s, z_s). Implementing the image source method [17], the image source position can be represented as $(qx_s + 2m_x L_x, jy_s + 2m_y L_y, kz_s + 2m_z L_z)$, where (L_x, L_y, L_z) are the length, width and the height of the room, respectively, and where each element in (q, j, k) takes on values -1 or 1, indicating the direction of the considered image sources in each dimension, and each element in (m_x, m_y, m_z) takes on integers from $-N$ to $+N$, indicating reflection order with respect to each dimension with N the predefined maximum reflection order. The path length for each ray arriving at the receiver is the distance between the image source position and the receiver position. The gain along each path is calculated as the multiplication of its length and wall reflection coefficients for all reflections.

B. Room acoustical parameters estimation

In this subsection, we first discuss the existing work on estimating the room geometry vector. After that, we review a closely related topic, room volume estimation. Finally, we review the estimation of reflection coefficients and reverberation time.

Room geometry is an important room acoustic parameter. Existing algorithms to estimate room geometry from RIRs all require prior information about the locations of the sources and the microphones [18]–[22]. [21] uses single-channel RIRs which are simulated by the image source method in a rectangular room, and a set of time of arrival (TOA) measurements of reflections to estimate 2D room geometry. It assumes that the TOA measurements are labeled with image sources and that RIRs consist of direct sound and the first and second-order reflections. [22] uses TOAs for 3D room geometry estimation with the image-source method simulated RIRs and measured RIRs. In contrast to [21], [22] obtains sets of TOAs from RIRs by detecting and labeling peaks in RIRs. These TOAs are used to estimate the source position and image source positions with knowledge of the array geometry of receivers. Finally, the room geometry can be inferred with estimated positions.

[18] proposes a method to estimate the 3D room shape from real-measured RIRs by exploiting the properties of Euclidean distance matrices and the first-order reflections. Although it requires only a single source, it requires at least four receivers and their pairwise distances. In addition, it may misclassify higher-order reflections as first order reflections [19]. In [19], the room geometry is estimated from simulated RIRs between one sound source and five receivers by a two-step geometrical method. The method first identifies the first-order image source positions and estimates the room geometry based on the image source positions. It requires knowledge of the pairwise distances between receivers. This method can achieve 1 cm

estimation accuracy. [20] infers the room geometry efficiently from simulated RIRs obtained with the image-source method using a graph theoretical approach. The echo combinations are modeled as nodes and the task is to find the maximum independent set in the graph, which refers to a set of vertices without direct interconnection. The image source positions can be calculated when the echoes are correctly labeled. After that, the room geometry can be inferred efficiently. It can achieve an average of 2.4 cm accuracy with at least two sources and five receivers.

A relaxation of room geometry estimation is the room volume estimation problem. Room volume estimation was formulated as a classification problem in [23], where room volume is classified into six volume class values. Seven room acoustical parameters are first extracted from a given RIR and serve as the input of the model. With these parameters, a statistical pattern recognition approach is used for room volume classification. This method can achieve a 0.1% equal error rate (EER) with simulated RIRs and a 19.1% EER with real-measured RIRs and does not require source-to-receiver distance. However, room volume is continuously distributed. Recently, room volume estimation was formulated as a regression problem [24]. Room volume is estimated with CNNs from noisy reverberant signal-channel speech signals that are split into frames with a 25% overlap. After training, the estimated volume is within approximately a factor of two to the true volume value.

Reflection coefficients characterize room reverberation effects. However, they are difficult to estimate directly and we are not aware of existing work on reflection coefficients estimation. Since reverberation time also characterizes room reverberation effects and is closely related to reflection coefficients, we briefly discuss work on reverberation time estimation. The reverberation time, $RT60$, of a room is defined as the time it takes for sound to decay 60 dB. Sabine-Franklin's formula [25] is commonly used to estimate the reverberation time:

$$RT_{60} = \frac{24 \ln 10}{c_{20}} \frac{V}{Sa} \approx 0.1611 \text{sm}^{-1} \frac{V}{Sa}, \qquad (1)$$

where c_{20} is the speed of the sound in the room for 20 degrees Celsius, V is the room volume, S is the total surface area of the room and a is the average absorption coefficient of room surfaces. From (1), we can conclude that reverberation time is related to room geometry and reflection coefficients. Given RIRs, the reverberation time can be directly estimated from the calculated energy decay curve [26], [27].

C. Multilayer perceptron

MLPs refer to neural networks that are composed of multiple layers (perceptrons), where each unit in one layer is connected to all units in the previous layer. The perceptron concept was first proposed by Rosenblatt in 1958 [28]. With each layer, an intermediate result is computed as the dot product of the input and the weights and an added bias, which is forwarded to the non-linear activation function. Each perceptron can be written mathematically as

$$y = \varphi(w^T x + b), \qquad (2)$$

where φ denotes the non-linear activation function, w and b are the weights and bias, and x and y are the input and the output of the perceptron.

[29] demonstrates that an MLP with only one hidden layer and an arbitrary continuous sigmoidal nonlinearity can uniformly approximate any continuous function. Although an MLP with only one hidden layer can uniformly approximate any continuous function, the number of neurons has to be exponentially large. It has been proved that considering the expressiveness of an MLP with ReLU activation, depth is more important than width [30]. This motivates us to use MLPs with more hidden layers instead of a wide shallow network. MLPs are relatively straightforward to implement and widely used in a variety of classification and regression problems, e.g., [3]–[5], [31], [32].

D. Convolutional neural networks

CNNs show a good modeling ability in various applications. CNNs capture spatial relationships of the input by means of parameter sharing and sparse connection. CNNs were first proposed by [33] for visual pattern recognition.

The layers of CNNs each perform a set of filtering operations, each commonly referred to as a *channel*, with a non-linear function operating on the biased filter output. The resulting output is a set of feature maps, which generally is reduced in dimensionality using a pooling layer. With increasing depth the features extract signal patterns that are increasingly position independent, as each kernel does not change when it slides over the signal reduces resolution. The parameters of the kernels are learned through the training process.

Many variations of CNN architectures have been developed, such as LeNet, AlexNet and VGGNet. LeNet, a classical CNN, was first proposed in the 1990s for handwritten and machine printed character recognition [34]. In 2012, AlexNet was proposed for image classification problems and obtained a considerably lower error rate than the previous state-of-art [35]. This error rate was further reduced with VGGNet [36]. From these classical CNN architectures, we can learn how to build a convolutional neural network. A CNN commonly consists of several convolutional layers, each followed by a pooling layer for downsampling, a few dropout layers to prevent overfitting, and several fully connected layers at the end.

CNNs have been used for various applications. CNNs are primarily used in computer vision, for example, image classification [6]–[8]. In addition to image data, CNNs can also analyse videos [37]–[39]. Until recently, CNNs were not widely used in acoustic signal processing. Recent applications confirm that CNNs show a good modeling ability for acoustic problems and can outperform state-of-the-art algorithms in this context. Such applications include speech dereverberation [40]–[42], speech enhancement [43]–[45].

III. PROBLEM FORMULATION

In this section, we formulate our problem, i.e., room acoustical parameter estimation from RIRs, and discuss the motivation for using deep neural networks to solve it.

We aim to use deep neural networks to estimate room acoustic parameters separately and blindly from a *single* RIR. Since the room acoustical parameters are described by continuous variables, we formulate the room acoustical parameter estimation problem as a regression problem. We define the input and output pair of the neural network with a random variable pair (X, Y). Specifically, in our problem, X is an $\mathbb{R}^{d_X}$-valued random variable that represents RIRs where d_X denotes the length of each RIR signal vector, and Y is an $\mathbb{R}^{d_Y}$-valued random variable that represents the room acoustical parameters where d_Y denotes the length of each room acoustical parameter vector.

We aim to learn a continuous deterministic function h to predict y from x, where (x, y) is a realisation of the random variable pair (X, Y). Hence, we have $\hat{y} = h(x)$ where $\hat{\cdot}$ labels an estimate. To measure the generalisation ability of the learned function h, we use a loss function $l : \hat{y} \times y \to \mathbb{R}_+$. The risk R of the predictor can then be defined as:

$$R = \mathbb{E}[l(h(x), y)], \tag{3}$$

where the expectation $\mathbb{E}$ is calculated with respect to the distribution $f_X(x)$ (recall y is a deterministic function of x). As the neural network does not know the distribution $f_X(x)$ of the input data during learning, we approximate the risk R of the predictor with the empirical risk R_{emp} on the training set:

$$R_{\text{emp}} = \frac{1}{m} \sum_{i=1}^{m} l(h(x_i), y_i), \tag{4}$$

where m denotes the size of training dataset and each (x_i, y_i) pair is one copy of the realisation $(x, y) \in \mathbb{R}^{d_X} \times \mathbb{R}^{d_Y}$ in the training dataset.

As we have mentioned above, the RIR is affected by both room geometry and reflection coefficients. For a given room geometry, reflection coefficients, and source and microphone position, the corresponding RIR can be computed for an empty box-shaped room. However, given an RIR in the real world, we might be not able to determine a set of parameters due to the existence of obstacles, a non-regular room shape, changes in temperature, and measurement noise. As a result, we conclude the relationship between the RIR and the room acoustical parameter is probabilistic. It is difficult to use conventional signal processing techniques to estimate room geometry and the reflection coefficients since the RIR can not be formulated as an analytical function of the room acoustical parameters. This motivates us to use deep neural networks as a non-linear mapping function to estimate room geometry and reflection coefficients from RIRs.

When we consider the effect of room geometry on RIRs, each geometry corresponds to a characteristic set of arrival times for the pulses. We hypothesize that the kernels of CNNs can extract the arrival-time patterns, where the room geometry information lies. Hence we use CNNs to estimate the room geometry from RIRs.

The effect of the reflection coefficients on RIRs is encoded in the strength of each pulse in the RIRs. It is independent of the time of arrival (TOA) of each pulse. With a multilayer perceptron, these pulses can be treated as features. This motivates

us to use MLPs when we estimate reflection coefficients since we assume this information is mainly related to the feature values.

IV. ROOM GEOMETRY ESTIMATION

In this section, we describe room geometry estimation based on convolutional neural networks. We solve the problem first for simulated data and then use transfer learning to solve the problem for real-world data.

In convolutional neural networks (CNNs), the receptive field of each neuron is processed with a set of kernels that do not vary across the input data. For our geometry-estimation problem, this corresponds to assuming that the RIR contains similar structures with respect to room geometry across all delays. In this section, we describe how we use convolutional neural networks to estimate room acoustical parameters. We first describe our base method and how we evaluate the precision of our model. We then propose two methods to improve the accuracy of the base method. Finally, we generalize our method to real-world RIRs.

A. Baseline method

As our base method, we use CNNs to estimate the room geometry vector from RIRs blindly. We hypothesize room geometry vectors can be estimated from a single random RIR of a room without any additional information. To solve the problem, our neural network has three output nodes for the length, width, and height of a room. We use the time-domain RIR as the input of our regression model without any pre-processing. Since the ordering of the three lengths of the geometry is arbitrary, we re-order the geometry vector in ascending order as a pre-processing step.

We adopt a commonly used CNN architecture as a basis. In this architecture, each convolutional layer is followed by a batch normalization layer [46] and an activation function. Since our input signal is a time-domain signal, we use one-dimensional convolutional layers and one-dimensional batch normalization layers. To keep a balance between the number of parameters and the modeling ability of neural networks, the neural network consists of eight one-dimensional convolutional layers and three fully connected layers. The number of channels (filters) in the convolutional layers increases with depth while the output dimensionality of the convolutional layers decreases.

In a regression problem, a quadratic loss is commonly used to track the training process and measure the generalization ability. Using this quadratic loss in (4), we define the mean square error (MSE) as the empirical risk, which is used as the objective function to train our CNN in order to minimize the squared distance between the estimated room geometry and the true room geometry. We chose the MSE loss since it is relatively sensitive to outliers. The loss function is then defined as

$$l(g, \hat{g}) = \frac{1}{m} \sum_{i=1}^{m} \parallel g_i - \hat{g}_i \parallel_2^2, \tag{5}$$

where $\parallel \cdot \parallel_2^2$ is the l^2-norm, m denotes the size of training dataset, $g \in \mathbb{R}^{m \times 3}$ denotes the true room geometry and $\hat{g} \in \mathbb{R}^{m \times 3}$ denotes the corresponding estimated room geometry.

To characterize the estimation performance of our method, we evaluate bias and variance on the test data. Bias measures the mean deviation of our estimates from the true value and variance measures how much our estimates vary from the mean estimated value. Minimizing the MSE results in a balance between bias and variance since the relationship between MSE, bias and variance can be described as

$$\text{MSE} = \text{Bias}^2 + \text{Variance.} \tag{6}$$

Since bias is also a parameter that a neural network tries to learn during the training process, our CNN model should in principle result in an unbiased estimator. For an unbiased estimator, we can increase the precision by averaging over the estimates.

B. Improved methods

Two methods can be used to improve the accuracy of our baseline method, i.e., the averaging method and the semi-blind estimation method. We describe both methods separately in this subsection.

Multiple RIRs can be used to increase estimation precision by averaging estimates. For each room, we select N random independent RIRs. The method is to average over the N estimates to calculate the final estimate for the room. The variance of the estimator will decrease by averaging over N independent estimates. Although the accuracy is limited by the bias, the estimation precision can be increased.

In addition to the above mentioned averaging method, we can also increase accuracy by adding restrictions when we generate RIRs. When we estimate room geometry from RIRs, the source/receiver position, and reflection coefficients can be considered as nuisance factors. We want to reduce the effect of nuisance factors in our problem to increase estimation accuracy. It requires more effort and more information to assume knowledge of reflection coefficients or exact source/receiver position. However, we can consider a setup where the relative position between the source and the receiver is fixed without the system knowing the distance or absolute position. We then remove one nuisance factor in RIR generation. By adding such a restriction, we hypothesize the estimation accuracy can be increased compared to blind room geometry estimation.

C. Generalization to real-world room impulse responses

Our goal is to generalize our method to real-world RIRs. On the one hand, since the amount of available real-world data is insufficient for training, we augment our data by processing our simulated RIRs to make our simulated RIRs close to real-world data. On the other hand, due to the imbalanced amount of simulated database and real database, transfer learning can be applied to improve generalization performance. In this subsection, we will first discuss how we use transfer learning. After that, the data augmentation technique will be covered. Finally, we describe how we apply our method to real-world RIRs.

Transfer learning [47] was proposed to improve the performance of a new task based on prior knowledge from a related trained task. Since we are able to generate a simulated RIR database of sufficient size to cover a wide range of room geometriesfor training, we can first train a neural network with an RIR database generated with the image source method. Then this trained neural network can be used as initialization when we train the neural network with a real RIR database of small size.

Instead of directly using transfer learning for real RIR database from the pre-trained model, which is trained on the ISM generated RIRs, we augment data as a transition stage. Compared to real-world measured RIRs, RIRs that are generated by the ISM lack some distortions, for example, additive environmental noises. Consequently, the neural network, which is trained by simulated RIRs, may adapt to certain features that are obscured to a real-world database and may fail to generalize well to a real RIR database. [48] proposed a simple and computationally cheap method to augment data for speech recognition, where they warp the features, mask blocks of frequency channels, and blocks of time steps. With this simple augmentation method, they could outperform prior work and achieve state-of-art performance. Inspired by this work, we can add some distortions to our simulated RIR as a data augmentation policy. In the following several paragraphs, we will introduce how we augment our data.

In the real world, it is almost impossible to obtain clean RIRs. In rooms and concert halls, a signal to noise ratio (SNR) of an RIR is commonly between 30 and 50 dB [27]. Hence, it is reasonable to include additive noise with an SNR between 30 and 50 dB in the RIR.

Obstacles are quite common in the real world, but we are not aware of an efficient method to simulate the effect of obstacles. Since we want to apply our model to real-world data, we have to mimic the effect of obstacles in our simulated RIR database. In the context of this paper, we discuss two artificial distortion types and one analytical method to simulate RIRs with obstacles in rectangular rooms. We will discuss these three methods separately.

The first type of artificial distortion to simulate the effect of obstacles is computationally inexpensive although rudimentary. The existence of obstacles will block some reflection paths and add some extra reflection paths. As a consequence, the first method is to randomly add and delete a random number of pulses in each RIR generated by the ISM.

As the second method, we add patterns to the blocked pulses due to the existence of obstacles. This method is also computationally feasible for simulations. Since each RIR can be viewed as a composition of a direct path between each image source and the receiver, the reflective pulse is blocked when the corresponding image source is blocked by the obstacle. This method is not physically correct since it only considers the blocked reflective pulses when their last reflection segment is blocked by the obstacle. Our derived pattern covers a subset of true blocked reflective patterns. To avoid the occlusion effect, we consider 2D non-reflective obstacles to simplify the problem. The blocked area, which is extended to infinity, can be then be defined with the receiver as

the vertex and the obstacle as the base. When the shape of the obstacle is a quadrilateral, the blocked area can be considered as a pyramid that extends to infinity. Our task is to determine whether the image source lies inside this extended pyramid. To determine the position of the image source, we calculate the dot product between the normal of each face and the vector between the receiver and the image source position. If the dot products are negative with respect to each face, then the image source is inside this pyramid. The method can be generalized to determine whether the reflective pulse is blocked when the obstacle is any polygon.

As the third method of modeling obstacles, we use a method based on adaptive rectangular decomposition (ARD) to simulate the sound propagation in 3D space with obstacles, which was proposed to model sound propagation in 3D complex environments [49]. This method utilizes the analytical solution of the wave equation in a rectangular domains and an efficient implementation of the discrete cosine transform (DCT) to facilitate computation on a desktop computer. However, it remains a challenge to generate an RIR database of sufficient size to train a neural network with this ARD-based method. As a result, this method is only used as a data augmentation method in the context of this paper. The procedure can be summarised as follows. We approximate each obstacle as a cuboid. Adaptive rectangular decomposition is then utilized to decompose the scene into rectangular partitions. After that, sound propagation can be simulated in each partition with the analytical solution to the wave equation on rectangular domains based on the DCT [13]. For the absorbing boundary, a perfectly matched layer absorber is employed [50]. A finite-difference approximation is used for sound propagation between two neighboring rectangular partitions. The RIRs that are generated with this method provide a useful transitional RIR between the RIRs generated with the image source method and real measured RIRs.

Our ultimate goal is to make the model work with a real-world RIR database. We first use transfer learning from the ISM generated RIRs to the transitional RIR database, which includes RIRs with noise, RIRs with obstacles generated with the three different methods. We then use transfer learning again from this transitional model with a real RIR database. To make efficient use of the small number of real world RIRs for our experiments, we use cross-validation [51] to train and test room geometry estimation. That is, we first divide the database into distinct parts. Each time, we select one subset as the test dataset and mix the remaining subsets as the train dataset. Finally, we average the test results over the folds of the cross-validation method.

V. ROOM REFLECTION COEFFICIENTS ESTIMATION

We now describe room reflection coefficients estimation. Since databases that contain both RIRs and reflection coefficients are not available, the method will be applied to simulated data only. RIRs are composed of reflective pulses. The strength of reflective pulses depends on reflection coefficients and propagation path length. We hypothesize MLPs are able to learn reflection coefficients from a RIR without any additional information.

We first describe the general estimation procedure and discuss the effect of re-ordered reflection coefficients on estimation accuracy. After that, we discuss the frequency dependency of the reflection coefficients. Finally, we describe how we link the reflection coefficients with the room geometry.

A. General reflection coefficients estimation

The reflection coefficient is a factor determining the RIR and this factor is encoded in the strength of reflective pulses in an RIR. We hypothesize there exists a continuous mapping function from the RIR signal to the reflection coefficient. Since MLPs can uniformly approximate any continuous function, we use MLPs to estimate reflection coefficients from a random RIR blindly. We use the time-domain RIR as the input of our regression model without any transformation. Similarly to our reflection coefficient estimation problem

In a real-world room, reflection coefficients are different on different walls and can even be different in different areas of a single wall. We will not cover different reflection coefficients on a single wall. Thus, In a rectangular room, we assume there are six reflection coefficients corresponding to the six walls. We re-order the six reflection coefficients in ascending order as a pre-processing step.

Similarly to the room geometry estimation problem, we use the MSE as our objective function to train the model, which is defined as

$$l(c, \hat{c}) = \frac{1}{m} \sum_{i=1}^{m} \| c_i - \hat{c}_i \|_2^2, \tag{7}$$

where $c \in \mathbb{R}^{m \times 6}$ is the true reflection coefficient matrix and the $\hat{c} \in \mathbb{R}^{m \times 6}$ is the estimated output.

We then discuss the effect of ordered reflection coefficients. We aim to verify that our neural network does learn the reflection coefficients from the RIRs and does not just correspond to an ordering of random outputs unrelated to the reflection coefficients. We use $X = [X_1, ..., X_6]$ to denote the six reflection coefficients and $Y = [Y_1, ..., Y_6]$ to denote the target of our neural network, i.e., the six ordered reflection coefficients. The real output of our neural network is denoted by $\hat{Y} = [\hat{Y}_1, ..., \hat{Y}_6]$. In the following we assume that the coefficients each have a uniform distribution, which we will impose in our simulation experiments.

We use $\tilde{Y} = [\tilde{Y}_1, ..., \tilde{Y}_6]$ to denote a set of ordered but unrelated random variables. Thus, distance measures between Y and $\tilde{Y}$ form an upper bound on the expected error of our neural network output: $E[|Y_i - \hat{Y}_i|^2] < E[|Y_i - \tilde{Y}_i|^2]$. $E[|Y_i - \hat{Y}_i|^2]$ will be computed experimentally for each i, which corresponds to the MSE. Our objective here is to compute $E[|Y_i - \tilde{Y}_i|^2]$ theoretically for each i.

We first need to compute the probability density function of Y_i and $\tilde{Y}_i$. Since Y_i and $\tilde{Y}_i$ are the i-th order statistic of $X_1 \cdots, X_6$ respectively, they are identically independent distributed for each i. We assume $X_1, \cdots, X_6$ are iid random variables that follow a standard uniform distribution. We can then compute the probability density function of Y_i and $\tilde{Y}_i$ respectively according to the order statistic [52]. That is, $Y_i \sim \text{Beta}(i, 7 - i)$ and $\tilde{Y}_i \sim \text{Beta}(i, 7 - i)$, where

Beta$(.,.)$ denotes the beta distribution. The Beta distribution is a continuous distribution defined on the range $(0, 1)$ with density

$$f_Y(y) = \frac{1}{\text{B}(i, 7 - i)} y^{r-1} (1 - y)^{s-1}, \tag{8}$$

where $\text{B}(.,.)$ is the Beta function. The pdf of $\tilde{Y}_i$, $f_{\tilde{Y}}(y)$, is identical to that of $f_Y(y)$.

With the probability density function of Y_i and $\tilde{Y}_i$, our next step is to compute the probability density function of $Y_i - \tilde{Y}_i$, which is denoted as D_i. Following Theorem 2.1 in [53], if Y_i and $\tilde{Y}_i$ are two independent random variables having support in $(0, 1)$, the pdf of $D_i = Y_i - \tilde{Y}_i$ is defined as

$$f_{D_i}(d) = \begin{cases} \int_0^{1+d} f_Y(t) f_{\tilde{Y}}(t - d) \mathrm{d}t & -1 < d < 0 \\ \int_0^{1-d} f_Y(d + t) f_{\tilde{Y}}(t) \mathrm{d}t & 0 < d < 1 \end{cases}. \tag{9}$$

With this pdf, we can compute the second moment of D_i, which corresponds to the expected value of $|Y_i - \tilde{Y}_i|^2$, as

$$E[D_i^2] = \int_{-1}^{1} d^2 f_{D_i}(d) \mathrm{d}d. \tag{10}$$

With the above derivation, we are able to calculate the expected value of $|Y_i - \tilde{Y}_i|^2$ for each i. Taking the square root of the expected values, we can compute the expected upper bound of the root mean square error (RMSE), $\sqrt{E[|Y_i - \tilde{Y}_i|^2]}$, which for the six dimensions is $[0.1750, 0.2259, 0.2474, 0.2474, 0.2259, 0.1750]$.

B. Frequency dependent reflection coefficients estimation

In this subsection, we discuss the frequency dependency of the reflection coefficients. To define an appropriate model for estimating frequency-dependent reflection coefficients, we must know how reflection coefficients vary with frequency. [54] lists several absorption coefficients in different frequencies. For example, the absorption coefficients of a painted concrete block change from 250 Hz (0.05) to 4000 Hz (0.08), the absorption coefficients of a lightweight drapery change from 125 Hz (0.03) to 250 Hz (0.04), and the absorption coefficients of plaster on lath change 500 Hz (0.06) to 4000 Hz (0.03). As all these examples change only moderately over frequency, we assume a simple model with piecewise constant reflection coefficients.

With the piecewise constant reflection coefficient assumption, we add a preprocessing step to divide the full-band RIR into several frequency bands with bandpass filters so that we can estimate reflection coefficients in different frequency bands. Among different kinds of bandpass filters, Chebyshev filters show a good computational speed although they are not perfect on stop-band attenuation [55]. Consequently, we choose Chebyshev type I filters [56] as our lowpass filter, which can be transformed into a bandpass filter or highpass filter as needed. With this pre-processing process, we will get access to RIRs in different frequency bands. We can then apply the previously discussed estimation methods for each frequency band separately.

C. Linking reflection coefficients with room geometry

Knowledge of six reflection coefficients only is generally insufficient. In this subsection, we focus on how to link the reflection coefficients with the room geometry. We assume that we already know the room geometry that can be estimated as described in Section IV. This linking problem can be solved by two methods, a machine learning based method and a conventional signal processing method.

With the machine learning based method, we build a CNN that takes an RIR signal conditioned on the room geometry as the input. The choice of CNN architecture is based on the logic in Section IV, where the conditioning is the only difference. The conditioning is fed into the network twice, at the input layer and at a middle layer. The output is a combination of the room geometry and the corresponding pairs of reflection coefficients. Within each pair, since there does not exist an order between two reflection coefficients, we re-order the two reflection coefficients in ascending order.

With the conventional signal processing method, we use $RT60$ as a bridge. On the one hand, ISO 3382 [57] shows how to measure $RT60$ from the reverberation time $T20$ or $T30$. We first need to calculate the energy decay curve from the RIR signal. The energy decay curve EDC at time t is defined as [26]

$$EDC(t) = \int_t^\infty h^2(\tau)\, \mathrm{d}\tau, \tag{11}$$

where $h(\tau)$ is the room impulse response. The reverberation time $T20$ ($T30$) is defined as the time that the energy decays from -5 dB to -25 (-35) dB, which can be calculated from the energy decay curve. With this, $RT60$ is three times $T20$ or twice $T30$. On the other hand, we can compute $RT60$ with Sabine-Franklin's formula as in (1). As what we have mentioned, we can estimate room geometry as in Section IV and estimate reflection coefficients as in Section V-A. Different combinations of room geometry and reflection coefficients result in a different $RT60$. By performning an exhaustive search, we are able to find a combination of room geometry and reflection coefficients that is closest to the correct $RT60$.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present our experiments. In the first subsection, we describe the setup of our experiments. We describe experiments on room geometry estimation in the second subsection. Finally, we present our experiments on the estimaton of the reflection coefficients.

A. Experimental setup

In the following, we first discuss the database we used to train and test our model. After that, we describe the configuration of our neural networks and how we train and test them. Finally, we introduce how we use bandpass filters for subband RIRs in the frequency-dependent reflection coefficient estimation problem.

TABLE I
DATABASE DESCRIPTION

Dataset	# rooms	# sources	# receivers
Real-world RIRs	9	5	31
Clean RIRs of empty room	400000	1	1
RIRs with noises	200000	1	1
RIRs with the 1st artificial distortion type	200000	1	1
RIRs with the 2nd artificial distortion type	50000	1	1
RIRs generated with the ARD-based analytical method	144	1	1000

1) Database: As is discussed in Section IV-C, a large-scale dataset of good quality is needed to train neural networks. An overview of the database we use is shown in Table I.

We used [58] as our real-world RIR database because it contains a relatively large number of real RIRs, several room types are covered, and the room geometry was measured in each room. This database contains nine distinct rectangular rooms that are not empty. Since we aimed our work at moderate or small rooms, we did not include three large rooms of the database, i.e., one conference room (with geometry $28 \times 11 \times 3$ m) and two lecture rooms (with geometry $20 \times 12 \times 5$ m and $23 \times 17 \times 7$ m). The selected six rooms include one hotel room, one meeting room, three office rooms, and one enclosed staircase. The geometry of these selected rooms varies between $4.4 \times 2.8 \times 2.2$ m and $14.2 \times 6.9 \times 3.6$ m. The corresponding RT30, the time that it takes to decay 30 dB, varies between 0.59 s and 1.85 s. Within each room, an average of 155 RIRs is given between five sources and 31 receivers.

To build an RIR dataset, we used the ISM to simulate RIRs [17]. We refer to this dataset as a clean RIR dataset of empty rooms. The shape of the rooms is rectangular and the rooms are empty. The speed of sound was set to $c = 340$ m/s. The sampling frequency was set to 8000 Hz. The length of each RIR was 4096 because an approximate 0.5 s RIR contains at least the direct path signal and early reflections in an indoor environment. Each dimension of the room geometry, i.e., length $\times$ width $\times$ height, was assumed to be iid between $6 \times 5 \times 4$ m and $10 \times 8 \times 6$ m. The room geometry range covers moderate and small rooms and is close to the real-world RIR database described above. The reflection coefficients of the walls were simulated as iid between 0 and 1. We randomly placed one source and one receiver in each room and generated the corresponding RIR. We labeled each RIR with room geometry and reflection coefficients. In our experiments, the number of the image-source method simulated RIRs was 400000, which was divided into a training dataset, a validation dataset, and a test dataset with the ratio $7 : 2 : 1$ for the baseline method.

The clean RIR training dataset of empty rooms was randomly divided into two equal parts for RIRs with noise and the first artificial distortion type. With one part, an additive Gaussian white noise was added to each RIR with an SNR uniformly distributed between 30 dB and 50 dB.

With the first artificial distortion of the RIR as defined in Section IV-C, a random number (this number was set to be uniformly distributed between 10 and 100) of pulses was added or deleted from the first 0.1 s of the clean RIRs. This choice was motivated by the hypothesis that the early reflection part of RIR provides more information for room geometry estimation than late reverberation.

With the second artificial obstacle pattern as defined in Section IV-C, we generated an RIR database of 50000 rooms. For each room, we randomly placed one rectangular obstacle of an arbitrary size inside the room and generated the corresponding RIR. This process was repeated nine times, i.e., there were nine distinct distorted RIRs for each room in this database.

For the RIRs generated with the analytical method based on ARD, due to the restriction of computational cost, we simulated a scenario with one source and 1000 receivers in each of 144 rooms. We randomly placed one to three obstacles of a random size in each room. We changed the reflection coefficients and geometry of the room. Each combination was denoted as one configuration.

2) Neural network description: In this subsection, we describe how we train and test our neural networks. In addition, we describe the configuration of our neural networks for different objective functions. We did an ablation study on network architecture and hyperparameter tuning with a grid search as a preliminary experiment for each neural network. The network architecture and hyperparameters below were chosen based on this preliminary experiment with our target database. If some properties of the target database change, we always performed an ablation study on network architecture and hyperparameter tuning with grid search.

We used a GPU node to train our neural network. The output node is the room acoustical parameter of the given room. The network was trained with the Adam optimizer [59], to minimize the training loss. The learning rate of the Adam optimizer was 0.001 and the coefficients used for computing running averages of the gradient and its square were set to be $(0.9, 0.999)$. We iterated for 2000 epochs and recorded the MSE loss for each epoch. To prevent overfitting, early stopping is used as regularisation in our model [60]. Early stopping is performed when the validation performance degrades in 100 successive epochs to guarantee the training performance without overfitting and keep a balance on the computational effort. In each epoch, we set the model on evaluation mode and computed the validation error for early stopping. In addition, mini-batch based training is used to increase computational efficiency [61]. The batch size was set to be 50. After training, we set the model to evaluation mode and computed the RMSE per dimension in the test set.

For geometry estimation, our network architecture and the corresponding parameters are shown in Table II, where b denotes the batch size. First the layer size decreases as the number of channels (feature maps) increases. The features are finally mapped to the geometry with fully connected layers. We use a leaky rectified linear unit (Leaky ReLU) [62] as the activation function. After each convolutional layer, there are always a batch normalization layer and a Leaky ReLU layer [62], which we do not list in the Table II since the output size does not change. The network contains 4577763 trainable parameters in total.

To estimate six frequency-dependent reflection coefficients, we use a multilayer perceptron regressor with nine hidden layers. The size of each layer was halved with each layer, from 2048 to 8. A rectified linear unit (ReLU) [63] was used as an activation function after each hidden layer.

TABLE II
NETWORK ARCHITECTURE OF ROOM GEOMETRY ESTIMATION

Operation	Kernel Size	Stride	# Channels	Output Size
Input				$(b, 4096)$
Reshape				$(b, 1, 4096)$
Conv1D	4	4	32	$(b, 32, 1024)$
Conv1D	2	2	32	$(b, 32, 512)$
Conv1D	8	8	128	$(b, 128, 64)$
Conv1D	2	2	128	$(b, 128, 32)$
Conv1D	2	2	512	$(b, 512, 16)$
Conv1D	4	4	512	$(b, 512, 4)$
Conv1D	4	4	1024	$(b, 1024, 1)$
Conv1D	1	1	1024	$(b, 1024, 1)$
Reshape				$(b, 1024)$
Fully connected				$(b, 160)$
Fully connected				$(b, 64)$
Fully connected				$(b, 3)$

To link the reflection coefficients to the room geometry, the network is described in Table III, where b denotes the batch size and we omit the batch normalization layer and the Leaky ReLU layer in the table. The conditioning, i.e., the room geometry vector, is concatenated to the RIR at the input layer and to the reshaped output vector before the fully connect layers. Each output vector is reshaped to a 3×3 matrix, where the first column is the room geometry vector, each row of the second and the third columns is a pair of reflection coefficients corresponding to that edge.

TABLE III
NETWORK ARCHITECTURE OF LINKING REFLECTION COEFFICIENTS TO ROOM GEOMETRY

Operation	Kernel Size	Stride	# filters	Output Size
Input				$(b, 4099)$
Reshape				$(b, 1, 4099)$
Conv1D	3	3	32	$(b, 32, 1366)$
Conv1D	5	5	32	$(b, 32, 273)$
Conv1D	3	3	128	$(b, 128, 91)$
Conv1D	5	5	128	$(b, 128, 18)$
Conv1D	4	4	512	$(b, 512, 4)$
Conv1D	4	4	512	$(b, 512, 1)$
Conv1D	1	1	1024	$(b, 1024, 1)$
Conv1D	1	1	1024	$(b, 1024, 1)$
Reshape				$(b, 1024)$
Fully connected				$(b, 160)$
Fully connected				$(b, 64)$
Fully connected				$(b, 9)$

3) Sub-band RIRs: When we take frequency dependency into account, we assumed the reflection coefficients are piecewise constant. The order of the Chebyshev type I filter was set to be 10 for a relatively short transition band. The maximum ripple factor was set to be 1 dB. Each full-band RIR was transformed into three signals, a lowpass RIR ($0 - 1000$ Hz), a bandpass RIR ($1000 - 2000$ Hz), and a highpass RIR ($2000 - 4000$ Hz). With this transformation, we were available to four sets of sub-band RIR data. The training and test process, and the network configuration are the same as for the full band RIRs.

B. Experiments on room geometry estimation

In this subsection, we present experiments on room geometry estimation. We first compare the baseline method and the proposed semi-blind estimation method for simulated data. After that, we discuss experiments for the proposed

TABLE IV
COMPARISON OF BASE ROOM GEOMETRY ESTIMATION METHOD AND
SEMI-BLIND ROOM GEOMETRY ESTIMATION.

Method	Baseline method	Semi-blind method
RMSE (m)	$[0.0497, 0.0398, 0.0249]$	$[0.0180, 0.0181, 0.0167]$
Bias (m)	$[0.0048, -0.0032, -0.0013]$	$[0.0012, -0.0003, -0.0014]$
Variance (m^2)	$[0.0024, 0.0016, 0.0006]$	$[0.0003, 0.0003, 0.0003]$

TABLE V
ROOT MEAN SQUARED ERROR AND VARIANCE OF AVERAGING METHOD.

# RIRs	RMSE (m)	Variance (m^2)
1	$[0.049, 0.039, 0.045]$	$[0.0024, 0.0015, 0.0020]$
4	$[0.027, 0.033, 0.042]$	$[0.0007, 0.0011, 0.0018]$
8	$[0.022, 0.032, 0.040]$	$[0.0005, 0.0010, 0.0016]$
16	$[0.018, 0.031, 0.025]$	$[0.0003, 0.0009, 0.0006]$

averaging method. We then compare our proposed method with a reference signal processing based method. Finally, we describe how we generalize our method to real-world RIRs.

As the first experiment of room geometry estimation, we set up the experiments of our baseline method and the proposed semi-blind estimation method for simulated data. For the semi-blind room geometry estimation, we pre-set a random source-receiver relative position relationship and generated the corresponding RIR dataset, whose only difference with respect to our original RIR dataset was the receiver-source relative position. We compared the performance of these two cases in terms of RMSE, bias, median, and variance per dimension in the test set. We used the mean estimation error to approximate bias. In addition, we plot the error distribution of both methods in Figure 1, where the error here refers to the MSE of each room geometry estimation.
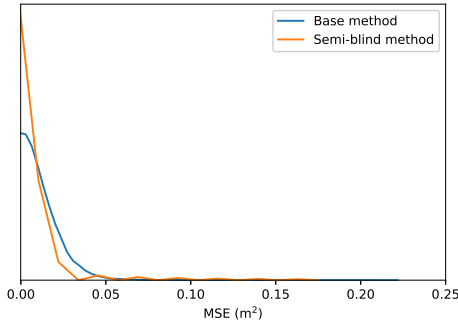

Fig. 1. MSE distribution of room geometry estimation.

We list the RMSE, bias, variance, and median of the base method and the semi-blind method in Table IV. A positive sign indicates our prediction is larger than the true geometry value. The RMSE, bias, and variance show different values with respect to length, width, and height because the range on these three elements is different and they are independent of each other. We also performed an experiment with our baseline method to compare the estimation accuracy between rectangular rooms and cube rooms. The RMSE of cube rooms is $[0.0534, 0.0374, 0.0243]$ m, which does not show a difference from rectangular rooms. This confirms that the estimation of length, width, and height are independent of each other. As shown in Table IV, the small bias vector confirms that our CNN model is not significantly biased after training and the small variance confirms that most estimation errors are relatively small and they do not vary much. The error distribution in the test set of both methods is shown in Figure 1. Observing the error distribution in Figure 1, the error follows a long-tailed distribution, which confirms that most

estimation errors are relatively small, which is consistent with the small variance in the test set. Comparing the experimental results of the baseline method and the semi-blind method, the semi-blind method outperforms the baseline method in terms of accuracy. To conclude, by the addition of a restriction on the relative source-receiver position relationship, the estimation accuracy of room geometry estimation is increased.

The second experiment of room geometry estimation was related to the proposed averaging method to increase the estimation accuracy. We aim to investigate the effect of the number of available RIRs in each room. For this experiment only, we generated a dataset with 16 RIRs per room to do the experiments and the RIRs in this dataset were distinct from those in the training dataset. In each room, 16 RIRs were generated independently, i.e., they correspond to 16 different randomly placed sources and 16 different randomly placed receivers. These RIRs were then use for inference with averaging. We ordered the estimates by the true room geometry and grouped the estimates to one, four, eight, and 16 estimates per room to perform the averaging method. Finally, we computed the RMSE, bias, and variance of the average method.

Next we describe the experimental result for the averaging method. The bias of the estimate is $[0.0045, -0.0027, -0.0015]$ m, which does not change by averaging over N estimates. The RMSE, median, and variance under different numbers of RIRs are listed in Table V. The method with one RIR corresponds to our baseline method. The RMSE, bias, and variance are slightly different from the results in Table IV because the test database is not the same. From Table V, we can conclude that, as expected, averaging leads to improved performance. The variance decreases with averaging but does not decrease by a factor of N since there exist nuisance factors, reflection coefficients, and source/receiver positions, which imply that the RIRs in each room are not independently conditioned on room geometry. To conclude, the performance is better when more RIRs are used for averaging although our estimation is still biased.

As the third experiment, we compared our proposed method with the signal processing method proposed in [20] in terms of system requirements, estimation error, and average run time. The experiments are both based on the RIRs generated by the ISM method. For calculating the run time, the experiments were averaged over 600 experiments. The result is shown in Table VI. The method in [20] uses five sources and five receivers and a 96000 Hz sampling frequency while the proposed method only requires sixteen random RIRs and an 8000 Hz sampling frequency. From the experimental results, our proposed method achieves approximately the same ac-

TABLE VI
COMPARISON OF PROPOSED METHOD AND STATE-OF-ART METHOD.

	Proposed method	Method in [20]
Average error (m)	0.0247	0.0235
Average run time (s)	3.22×10^{-4}	2.43

TABLE VII
ROOM GEOMETRY ESTIMATION WITH REAL-WORLD MEASURED RIRS.

Room	RMSE (m)	RMSE after averaging (m)
Hotel room	[0.1516, 0.1276, 0.2615]	[0.1046, 0.0505, 0.1169]
Meeting room	[0.1083, 0.0639, 0.1508]	[0.0916, 0.0220, 0.0440]
Office 1	[0.0508, 0.0532, 0.1023]	[0.0056, 0.0249, 0.0384]
Office 2	[0.0803, 0.0757, 0.2240]	[0.0390, 0.0207, 0.0938]
Enclosed staircase	[0.1790, 0.0998, 0.0970]	[0.1696, 0.0923, 0.0825]
Office 3	[0.1516, 0.0365, 0.1305]	[0.1432, 0.0081, 0.0112]

TABLE VIII
EVALUATION OF THE IMPORTANCE OF FOUR DATA AUGMENTATION
METHODS.

The left out data augmentation method	Average RMSE difference (m)
RIRs with noises	0.0310
RIRs with the 1st artificial distortion type	0.0570
RIRs with the 2nd artificial distortion type	0.0648
RIRs generated with the ARD-based analytical method	0.1210

curacy while requiring approximate 10^4 less computational effort after training. To conclude, our CNN based room geometry estimation method is computationally efficient with approximately the same estimation error and, in contrast to the conventional signal processing based method, does not require prior knowledge or knowledge of the measurement configuration. Moreover, if lower accuracy is required, our method allows the usage of fewer measurements.

Our last experiment on room geometry estimation was the generalization to real-world RIRs with transfer learning. Before feeding the real-world RIRs into the neural network, we first resampled the real-world RIRs to 8000 Hz and then used the first 4096 samples of the resampled RIR as the input. With transfer learning, the base method model was adopted as initialization and the learning rate of the optimizer was set to be one-tenth of the original learning rate. This generalization was split into two steps. We trained 500 epochs for each step to prevent overfitting. We describe the two steps in detail in the next two paragraphs.

The first step was the transfer learning from the base model with additive noise, randomly deleted and added pulses, derived approximate distorted RIRs due to obstacles, and the RIR genereated with the RD-based analytical method for obstacles. These distorted RIRs were mixed as the training dataset for transfer learning in the first step. The model after the first step was saved as an initialization for the second step.

In the second step, we used transfer learning with real-world RIRs [58]. Cross-validation was used for the six selected rooms in the database. In each test set, we computed the RMSE per dimension to evaluate the generalization performance. Since there were multiple RIRs per room, the proposed averaging method was performed in each test set to increase accuracy.

The experimental results for room geometry estimation with real-world measured RIRs are shown in Table VII. Before averaging over multiple estimates from multiple RIRs, the minimal RMSE on a single dimension is 0.05 m and the maximum error is 0.26 m. The 0.26 m RMSE appears in the hotel room with two beds and other furniture inside, which is a room with relative many obstacles, but this error reduces to 0.12 m after averaging. After averaging, the minimal RMSE is 0.01 m and the maximal is 0.17 m. The 0.17 m RMSE after averaging method appears in the enclosed staircase, which

is relatively difficult to handle because of the stairs. The difference between RMSE with and without averaging method does not consistently follow the results shown in Table V. This is because the real measured 151 RIRs in each room are from five sources and 31 receivers, which indicates the measurements are not independent from each other.

We did an additional experiment to evaluate the importance of these four augmentation methods, where we left one data augmentation method out each time and repeated the two steps in the previous experiment. We computed the RMSE after averaging and compared it with Table VII. We computed the average RMSE difference, where the positive sign indicates an increase in the RMSE when one data augmentation method is left out.

The average RMSE in Table VII after averaging is 0.0644m. The leave-one-out experimental result is shown in Table VIII. Observing the result, when one data augmentation method is left out, the corresponding RMSE increases. This shows all four data augmentation methods are all necessary and make a contribution to the estimation accuracy. In addition, comparing the increased RMSE (m), we can conclude that RIRs generated with the ARD-based analytical method is the most important among these four methods. This is likely because this method simulates the effect of obstacles on real-world RIRs most accurately.

C. Experiments on the estimation of reflection coefficients

In this subsection, we describe our experiments that relate to the reflection coefficients. We first describe the experiments on estimating only reflection coefficients from RIRs, where we cover the frequency-independent case and the frequency-dependent case. Next, we describe the experiment on linking the reflection coefficients to room geometry.

We performed the reflection coefficient estimation experiments under the assumption of six distinct reflection coefficients, one for each wall. We divide this into two cases according to their frequency dependency. For the frequency-independent reflection coefficients, we estimate the reflection coefficients from the corresponding full-band RIR. With respect to the frequency-dependent reflection coefficients, we estimate the reflection coefficients from the sub-band RIRs independently. We compared the estimation error of the sub-band RIRs and the full-band RIRs to explore the effect of frequency bands on reflection coefficient estimation accuracy.

The experimental results of estimating six distinct reflection coefficients in a rectangular room are shown in Table IX. With the full band RIRs, the average RMSE per dimension is 0.09. With the sub-band RIRs, part of the information of the RIRs is lost. Consequently, the RMSE of the sub-band RIRs is larger. In addition, the RMSE of the low pass RIR is smaller than

TABLE IX
RMSE OF MULTIPLE REFLECTION COEFFICIENTS ESTIMATION.

Signals	RMSE
Full band RIRs	$[0.0872, 0.0954, 0.0984, 0.0929, 0.0826, 0.0837]$
Low pass RIRs	$[0.0904, 0.0979, 0.1001, 0.0971, 0.0903, 0.0873]$
Band pass RIRs	$[0.1098, 0.1213, 0.1124, 0.0978, 0.0906, 0.0884]$
High pass RIRs	$[0.1108, 0.1241, 0.1146, 0.0981, 0.0927, 0.0923]$

that of the bandpass RIR and the high pass RIR. This is likely because the relation between the RIR and the coefficients is smoother for low pass signals and it is easier to learn a smoother function by a neural network. In addition, when observing the RMSE for each reflection coefficient, the RMSE in the middle position is relatively large. This is consistent with the upper bound in Section V-A and results from having ordered reflection coefficients in the interval $[0, 1]$.

Comparing the experimental results in Table IX and the upper bound derived in Section V-A, each RMSE in Table IX are substantially smaller than the upper bound derived in Section V-A. This indicates our neural network does learn reflection coefficients from RIRs instead of simply generating a set of ordered random numbers.

In the remainder of this subsection, we describe the experiments on linking the reflection coefficients to the room geometry as outlined in Section V-C. We start with the machine learning based method. With the machine learning based method, we computed the RMSE for the reflection coefficients to evaluate the estimation accuracy. Since the room geometry serves as conditioning, the RMSE for the room geometry is negligible and not recorded here. Based the estimated reflection coefficients, which are linked to the room geometry, we computed the $RT60$ with the Sabine-Franklin formula, which is compared with the $RT60$ calculated from the energy decay curve to compute the RMSE. After that, we took the six reflection coefficients from each output, re-ordered them, and computed the RMSE for each reflection coefficient again to compare the accuracy with the previous reflection coefficients only estimation experiment.

The experimental result of linking reflection coefficients to room geometry using machine learning based method is shown in Table X. Each row of the second and the third columns is the RMSE for the pair of reflection coefficients corresponding to that edge. The RMSE for the paired reflection coefficients is slightly worse than for the previous experiment but the model can still link a pair of reflection coefficients to the room geometry. The corresponding RMSE for the $RT60$ based on these estimates is 0.0220 s. When we reordered the six estimated reflection coefficients, the RMSE is $[0.0795, 0.0742, 0.0809, 0.0854, 0.0854, 0.0915]$, which is approximately the same as the result in Table IX. This result proves that the estimation accuracy of the reflection coefficients does not decrease but the linking operation decreases the accuracy a little.

In addition to the machine learning based method, we can also link the reflection coefficients to the room geometry using the conventional signal processing method. Since we use estimated room geometry and reflection coefficients, we only recorded the RMSE for $RT60$. We computed $RT60$ with the

TABLE X
RMSE OF LINKING REFLECTION COEFFICIENTS TO ROOM GEOMETRY.

Room geometry	Reflection coefficients	
Edge 1	0.1017	0.1391
Edge 2	0.1058	0.1435
Edge 3	0.1117	0.1427

estimated room acoustical parameters using Sabine-Franklin's formula. We then compared it with the $RT60$ calculated from the energy decay curve, and recorded the RMSE.

Computing the $RT60$ using the conventional signal processing method, the corresponding RMSE is 0.0083 s, which is smaller compared to the machine learning based method. Since the difference in the RMSEs for estimates of the room geometry is negligible, the difference in the RMSEs for the $RT60$ is due to the linking process of the reflection coefficients.

VII. CONCLUSION

We showed that it is possible to estimate the geometry of a shoebox-shaped room and also the reflection coefficients of its walls from RIRs using deep neural networks. We formulated the problem as a regression problem with the MSE as a loss function. In contrast to conventional methods, the proposed methods only requires a single RIR between a source and a receiver and do not require knowledge of their positions or relative distance. For the room geometry estimation task, we used convolutional neural networks. We first trained the neural network with artificial data. Then transfer learning was used to make the method work for real-world RIRs. We achieved an average of 0.065 m testing accuracy for real-world data. We used multilayer perceptrons to estimate the wall reflection coefficients from simulated RIRs. We obtained an RMSE of approximately 0.09 for each reflection coefficient when the reflection coefficients are different for the six walls. This value increased slightly if we require pairs of reflection coefficients to be associated with an estimated room geometry. In addition, we were able to estimate frequency-dependent reflection coefficients and achieved similar accuracy.

REFERENCES

[1] Wikipedia, "Augmented reality," 2019.
[2] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, cited By 8482.
[3] Y.-S. Park and S. Lek, "Chapter 7 - artificial neural networks: Multilayer perceptron for ecological modeling," in *Ecological Model Types*, ser. Developments in Environmental Modelling, S. E. Jrgensen, Ed. Elsevier, 2016, vol. 28, pp. 123 – 140.
[4] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43 – 62, 1997.
[5] R. Pearson, T. Dawson, P. Berry, and P. Harrison, "Species: A spatial evaluation of climate impact on the envelope of species," *Ecological Modelling*, vol. 154, no. 3, pp. 289 – 300, 2002.
[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[7] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Hcp: A flexible cnn framework for multi-label image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1901–1907, Sep. 2016.

[8] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyper-spectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, Oct 2017.

[9] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[10] S. Park, Y. Jeong, and H. S. Kim, "Multiresolution cnn for reverberant speech recognition," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, Nov 2017, pp. 1–4.

[11] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2263–2276, Dec 2016.

[12] J. Fan, C. Ma, and Y. Zhong, "A selective overview of deep learning," *arXiv preprint arXiv:1904.05526*, 2019.

[13] H. Kuttruff, *Room acoustics*. New York: CRC Press, 2014.

[14] D. R. Begault and L. J. Trejo, "3-d sound for virtual reality and multimedia," 2000.

[15] S. G. McGovern, "Fast image method for impulse response calculations of box-shaped rooms," *Applied Acoustics*, vol. 70, no. 1, pp. 182 – 189, 2009.

[16] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, 1979.

[17] I. A. L. Erlangen, "RIR generator," 2014.

[18] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proceedings of the National Academy of Sciences*, vol. 110, no. 30, pp. 12 186–12 191, 2013.

[19] T. Rajapaksha, X. Qiu, E. Cheng, and I. Burnett, "Geometrical room geometry estimation from room impulse responses," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 331–335.

[20] I. Jager, R. Heusdens, and N. D. Gaubitch, "Room geometry estimation from acoustic echoes using graph-based echo labeling," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 1–5.

[21] A. H. Moore, M. Brookes, and P. A. Naylor, "Room geometry estimation from a single channel acoustic impulse response," in *21st European Signal Processing Conference (EUSIPCO 2013)*, Sep. 2013, pp. 1–5.

[22] Y. E. Baba, A. Walther, and E. A. P. Habets, "3D room geometry inference based on room impulse response stacks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 857–872, May 2018.

[23] N. R. Shabtai, Y. Zigel, and B. Rafaely, "Room volume classification from room impulse response using statistical pattern recognition and feature selection," *The Journal of the Acoustical Society of America*, vol. 128, no. 3, pp. 1155–1162, 2010.

[24] A. F. Genovese, H. Gamper, V. Pulkki, N. Raghuvanshi, and I. J. Tashev, "Blind room volume estimation from single-channel noisy speech," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 231–235.

[25] A. Pierce, *Acoustics: An Introduction to Its Physical Principles and Applications*, 06 1989, vol. 34.

[26] M. R. Schroeder, "New method of measuring reverberation time," *The Journal of the Acoustical Society of America*, vol. 37, no. 3, pp. 409–412, 1965.

[27] M. Karjalainen, P. Antsalo, and T. Peltonen, "Estimation of modal decay parameters from noisy response measurements," *Journal of the Audio Engineering Society*, 2002.

[28] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, pp. 65–386, 1958.

[29] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, Dec 1989.

[30] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6231–6239.

[31] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5, pp. 183 – 197, 1991.

[32] M. H. Esfe, M. Afrand, S. Wongwises, A. Naderi, A. Asadi, S. Rostami, and M. Akbari, "Applications of feedforward multilayer perceptron artificial neural networks and empirical correlation for prediction of thermal conductivity of mg(oh)2eg using experimental data," *International Communications in Heat and Mass Transfer*, vol. 67, pp. 46 – 50, 2015.

[33] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, no. 2, pp. 119 – 130, 1988.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[37] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1725–1732.

[38] I. Aljarrah and D. Mohammad, "Video content analysis using convolutional neural networks," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, April 2018, pp. 122–126.

[39] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," *CoRR*, vol. abs/1503.08909, 2015.

[40] K. Han, Y. Wang, and D. Wang, "Learning spectral mapping for speech dereverberation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4628–4632.

[41] B. Wu, K. Li, F. Ge, Z. Huang, M. Yang, S. M. Siniscalchi, and C.-H. Lee, "An end-to-end deep learning approach to simultaneous speech dereverberation and acoustic modeling for robust speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, pp. 1289–1300, 2017.

[42] O. Ernst, S. E. Chazan, S. Gannot, and J. Goldberger, "Speech Dereverberation Using Fully Convolutional Networks," *arXiv e-prints*, p. arXiv:1803.08243, Mar 2018.

[43] T. Kounovsky and J. Malek, "Single channel speech enhancement using convolutional neural network," in *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, May 2017, pp. 1–5.

[44] N. Mamun, S. Khorram, and J. H. L. Hansen, "Convolutional neural network-based speech enhancement for cochlear implant recipients," *CoRR*, vol. abs/1907.02526, 2019.

[45] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," *CoRR*, vol. abs/1609.07132, 2016.

[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[47] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI Global, 2010, pp. 242–264.

[48] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," 2019.

[49] N. Raghuvanshi, R. Narain, and M. C. Lin, "Efficient and accurate sound propagation using adaptive rectangular decomposition," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 789–801, Sep. 2009.

[50] Y. S. Rickard, N. K. Georgieva, and Wei-Ping Huang, "Application and optimization of pml abc for the 3-d wave equation in the time domain," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 2, pp. 286–295, Feb 2003.

[51] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143.

[52] Wikipedia, "Order statistic," 2020.

[53] D. K. Nagar and Y. A. Ramirez-Vanegas, "Distributions of sum, difference, product and quotient of independent non-central Beta type 3 variables," 2013.

[54] T. Rossing, *The Science of Sound*. Addison-Wesley Publishing Company, 1990.

[55] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. USA: California Technical Publishing, 1997.

[56] L. Paarmann, *Design and Analysis of Analog Filters: A Signal Processing Perspective*, ser. The Springer International Series in Engineering and Computer Science.   Springer US, 2006.

[57] P. P. K. Normalizacyjny, *Acoustics - Measurement of room acoustic parameters - Part 2: Reverberation time in ordinary rooms (ISO 3382-2: 2008)*, 2008.

[58] I. Szoke, M. Skacel, L. Mosner, J. Paliesek, and J. Cernocky, "Building and evaluation of a real room impulse response dataset," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 863–876, Aug 2019.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[60] L. Prechelt, "Early stopping - but when?" March 2000.

[61] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: primal estimated sub-gradient solver for svm," *Mathematical Programming*, vol. 127, no. 1, pp. 3–30, 2011.

[62] A. L. Maas, "Rectifier nonlinearities improve neural network acoustic models," 2013.

[63] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10.   USA: Omnipress, 2010, pp. 807–814.