Götz, Georg; Falcon Perez, Ricardo; Schlecht, Sebastian; Pulkki, Ville

Neural network for multi-exponential sound energy decay analysis

Published in:
Journal of the Acoustical Society of America

DOI:
10.1121/10.0013416

Published: 12/08/2022

Document Version
Publisher's PDF, also known as Version of record

JASA ARTICLE

Neural network for multi-exponential sound energy decay analysis

Georg Götz,[a] Ricardo Falcón Pérez, Sebastian J. Schlecht,[b] and Ville Pulkki
Aalto Acoustics Lab, Department of Signal Processing and Acoustics, Aalto University, P.O. Box 13100, 00076 Aalto, Finland

ABSTRACT:
An established model for sound energy decay functions (EDFs) is the superposition of multiple exponentials and a noise term. This work proposes a neural-network-based approach for estimating the model parameters from EDFs. The network is trained on synthetic EDFs and evaluated on two large datasets of over 20 000 EDF measurements conducted in various acoustic environments. The evaluation shows that the proposed neural network architecture robustly estimates the model parameters from large datasets of measured EDFs while being lightweight and computationally efficient. An implementation of the proposed neural network is publicly available.
 https://doi.org/10.1121/10.0013416

(Received 14 December 2021; revised 8 July 2022; accepted 18 July 2022; published online 12 August 2022)

[Editor: Shiu Keung Tang] Pages: 942–953

I. INTRODUCTION

Room impulse responses (RIRs) are signals that describe the sound recorded by a receiver in a room in response to an impulsive excitation. They characterize how room, source, and receiver affect sound on the investigated transmission path, implicitly assuming a linear time-invariant system. The RIR decays gradually, thus, causing the well-known acoustic phenomenon called reverberation.[1]

Due to pronounced fluctuations in the RIR, the energy-time curve may be hard to interpret and difficult to use in further analyses. Schroeder proposed a backward integration procedure for obtaining smooth steady-state energy decay functions (EDFs),[2] which are also called Schroeder decay functions. Schroeder decay functions are frequently used in architectural acoustics to calculate the reverberation time, which is the time until the sound energy in an enclosure has decreased by 60 dB. The reverberation time is commonly determined for multiple frequency bands by fitting a straight line to the band-limited logarithmic EDF.[3,4]

Measured RIRs are usually contaminated with noise, for instance, resulting from the measurement equipment, ambient sound, or quantization. The noise inevitably affects the EDF[5] and causes errors when estimating the reverberation time based on a straight-line fit.[6] Several approaches were proposed to counter the effect of noise on the reverberation time estimation.[3,5,7,8] Alternatively, Xiang[9] and Karjalainen *et al.*[10] proposed to include an additional noise term in the model and perform a nonlinear regression.

In coupled rooms or rooms with a considerably nonuniform absorption material distribution, sound energy can decay with multiple decay rates.[1,11–13] For this reason, Xiang and Goggans model EDFs with multiple exponential decays and a noise term.[14]

This paper presents a neural-network-based approach for fitting multiple exponential decays and a noise term to an EDF. Despite being trained on a fully synthetic dataset, we show that such a neural network structure can robustly analyze real-world measurements. Fully synthetic training datasets can be easily generated and subsequently extended to different scenarios. Previous methods for multi-exponential sound energy decay analysis rely on iterative algorithms. Our approach has the advantage of being fully deterministic at inference time and requiring no user tuning while being robust and computationally efficient. Therefore, the neural network structure is especially appealing for room acoustic analysis and modeling with machine-learning-based approaches, for which it is essential to achieve robust performance on large datasets without manual intervention. The proposed network is lightweight, allowing it to be implemented on mobile devices. Furthermore, the neural-network-based structure allows efficient up-scaling and parallelization on modern hardware with dedicated graphics processing units (GPUs).

Our contribution is threefold. First, we present a lightweight and computationally efficient neural-network-based structure for sound energy decay analysis that achieves a comparable fitting performance as those of state-of-the-art methods. Second, we evaluate the proposed network and previous decay analysis approaches on two large datasets of more than 20 000 EDFs. Third, we provide an open-source decay analysis toolbox for MATLAB and PYTHON, comprising the neural network structure and our implementation of the other evaluated multi-slope decay analysis method.

The remainder of this paper is organized as follows. Section II states the problem formulation, and Sec. III provides an overview of prior work. Section IV describes the proposed neural network and its training in detail. Section V

[a]Electronic mail: georg.gotz@aalto.fi
[b]Also at: Media Lab, Department of Art and Media, Aalto University, Otakaari 5, 00250 Espoo, Finland.

https://doi.org/10.1121/10.0013416

presents an evaluation of the proposed network on two large datasets of real-world measurements and compares its performance to other state-of-the-art approaches. Section VI discusses the results. Section VII details the publicly available decay analysis toolbox, and Sec. VIII concludes the paper.

II. PROBLEM FORMULATION

Smooth EDFs can be obtained from RIRs via the backward integration procedure proposed by Schroeder.[2] The EDF $d(t)$ of a RIR $h(t)$ is calculated by

$$d(t) = \frac{1}{E}\sum_{\tau=t}^{L} h^2(\tau) \quad \text{with} \quad E = \sum_{\tau=1}^{L} h^2(\tau), \tag{1}$$

where t is the sample index and L is the number of samples in the EDF, which is also called the upper limit of integration (ULI).

EDFs can be modeled as a sum of multiple exponential decays and a noise term.[14] The model $d_K(t,\boldsymbol{\theta})$ of the EDF $d(t)$ is then given as[14,15]

$$d_K(t,\boldsymbol{\theta}) = N_0(L-t) + \sum_{i=1}^{K} A_i\left[e^{-13.8t/(f_s T_i)} - e^{-13.8L/(f_s T_i)}\right], \tag{2}$$

where $\boldsymbol{\theta} = [T_1, T_2, \ldots, T_K, A_1, A_2, \ldots, A_K, N_0]$ summarizes all of the decay parameters, T_i and A_i are the decay time and amplitude of the ith exponential decay, respectively, N_0 is the amplitude of the noise term, the constant $-13.8 = \ln(10^{-6})$ ensures that the sound energy has decayed to -60 dB after T_i seconds, $\ln(\cdot)$ denotes the natural logarithm, f_s is the sampling frequency of the RIR, and K is the model order, i.e., the number of exponential decays in the model. The constant second term in the square brackets accounts for the finite ULI and can be neglected for large L.[15] In the following, we usually refer to the decay model as $d_K(t)$, thus, dropping the decay parameters, $\boldsymbol{\theta}$, from the notation to improve readability.

Estimating the parameters, $\boldsymbol{\theta}$, is a crucial task for various problems in room acoustics. For example, the reverberation time can be determined by estimating the parameter T_1 for an EDF model with $K = 1$. Decay models of higher order have successfully been used to measure the absorption coefficients of materials[16] or characterize the sound decay of coupled spaces.[17,18]

III. PRIOR WORK

A. Sound energy decay analysis

Previous approaches for estimating the parameters, $\boldsymbol{\theta}$, differ mainly regarding the underlying model order. For model order $K = 1$, linear regression is commonly used to determine the reverberation time as a straight-line fit to the band-limited logarithmic EDF.[3,4] In this case, the noise term of the model is neglected, i.e., $N_0 = 0$. To get accurate estimates for T_1, the effect of the noise has to be countered by noise subtraction[5] or truncation of the RIR before backward integration.[3,7,8] The noise term, N_0, can be included in the model by using nonlinear regression.[9,10]

The sound decay of coupled rooms or rooms with considerably nonuniform absorption material distribution can usually not be modeled with a single decay rate.[1,11–13] In such cases, model orders $K > 1$ need to be considered. Xiang and Goggans proposed a Bayesian framework to determine the model parameters, T_i, A_i, and N_0, for $K \geq 1$.[14] The Bayesian formulation can also determine the most probable model order K given the measured EDF.[19] Numerous works have advanced the approach by investigating more accurate and efficient algorithms for estimating the parameters or determining the model order.[16,19,20]

Previous studies have evaluated the performance of single-slope EDF fitting software.[21–23] Katz[23] used a single measured lecture theater RIR, which was also used in a later study by Cabrera *et al.*,[21] together with artificial single-slope responses. Álvarez-Morales *et al.*[22] studied the software behavior on a slightly larger set of RIRs (15 receiver positions × 2 source positions, measured in a single auditorium). While Katz[23] still reported considerable differences among reverberation time estimation software in 2004, the later studies[21,22] found that reverberation time is consistently estimated within perceptual limits by the more recent software implementations in most cases. In all of the studies, the most inconsistent reverberation time estimates were obtained for low-frequency bands.[21–23]

B. Convolutional neural networks (CNN) in acoustics

The recent advances in machine learning and artificial intelligence have pushed the state-of-the-art in many different fields. Machine learning has traditionally been of interest in speech recognition,[24] natural language processing,[25] and music information retrieval.[26] More recent applications in acoustics include source localization and tracking,[27] blind room acoustic parameter estimation,[28] sound field scattering from three-dimensional (3D) objects,[29,30] and the inverse problem of object geometry regression from the scattered sound field.[31] For more details, the reader is referred to the thorough review by Bianco *et al.*[32] Although the applications, models, and task definitions vary considerably, most of the previously mentioned works share a common approach: a machine learning model, typically a neural network, extracts features and predicts the parameters of a parametric model.

Most of these applications can be divided into either classification or regression tasks, where the main difference is the output domain.[33] For classification, the goal is to select one or several categories, given a specific input. For regression, the goal is to predict one or several continuous values. The study by Fernández-Delgado *et al.*[34] includes a detailed analysis of many regression methods and applications. Depending on the network architecture and training procedure, some tasks can be formulated as regression or classification tasks. In this paper, the decay parameter

prediction is treated as a regression task, where we estimate the decay parameters as continuous values. On the other hand, the model order prediction is treated as a classification task with the categories "1 active decay slope," "2 active decay slopes," and so forth.

IV. PROPOSED METHOD

In this paper, we propose a neural network structure for estimating the parameters, $\boldsymbol{\theta}$, of the model in Eq. (2). We show that such a network can be trained on a fully synthetic training dataset to make predictions on real-world measurements. In Sec. IV A, we will first describe an example of such a network architecture. Subsequently, we provide details on the synthetic training dataset, the utilized loss functions, and the hyperparameters that can be used to train the network.

A. Network architecture

An example of the proposed neural network structure for estimating the sound decay parameters of the model in Eq. (2) is depicted in Fig. 1. We refer to this particular architecture as DecayFitNet in the remainder of this paper. The network takes EDFs as its input and returns preliminary estimates $\widetilde{T}_i$, $\widetilde{A}_i$, and $\widetilde{N}_0$ for the decay times, decay amplitudes, and noise term, respectively. Furthermore, it outputs the values $P_{K=i}$, which quantify the model order prediction. For example, by applying the logistic function

$$f(x) = \frac{1}{1+e^{-x}}, \tag{3}$$

we could get the probabilities $f(P_{K=i}) = [0.75, 0.15, 0.10]$, which are bound between 0 and 1, and indicate that the network predicts the model order $\widehat{K}$ to be $1, 2$, or 3 with the probabilities 0.75, 0.15, and 0.10, respectively. In the current implementation, we restrict the maximum model order to $\widehat{K}_{\max} = 3$, although higher model orders could also be supported in the future.

The network consists of a common base and individual output branches for the different estimated parameters. The base contains a sequence of three one-dimensional convolutional layers with intermediate max-pooling layers[35] along the time-axis and three fully connected layers. The output branches consist of two fully connected layers each. Rectified linear units (ReLU)[36] are used as activation functions after each network layer, excluding the output layers.

This type of network architecture is ubiquitous for many different applications and tasks. In theory, a network that only consists of fully connected layers and nonlinearities can be seen as a universal approximator.[37] Such networks are also called multilayer perceptron (MLP). In practice, the flexibility of MLPs is limited by the amount of training data, size of the network, and training procedure. Most recent deep learning approaches favor convolutional layers to reduce the number of weights needed.[33] Furthermore, models with subsequent blocks of convolutional layers, nonlinearities, and pooling layers are biologically inspired by the visual processing system found in many living beings.[38] The convolutional layers act as feature extractors by learning filters that process the input signal in such a way as to maximize the information required

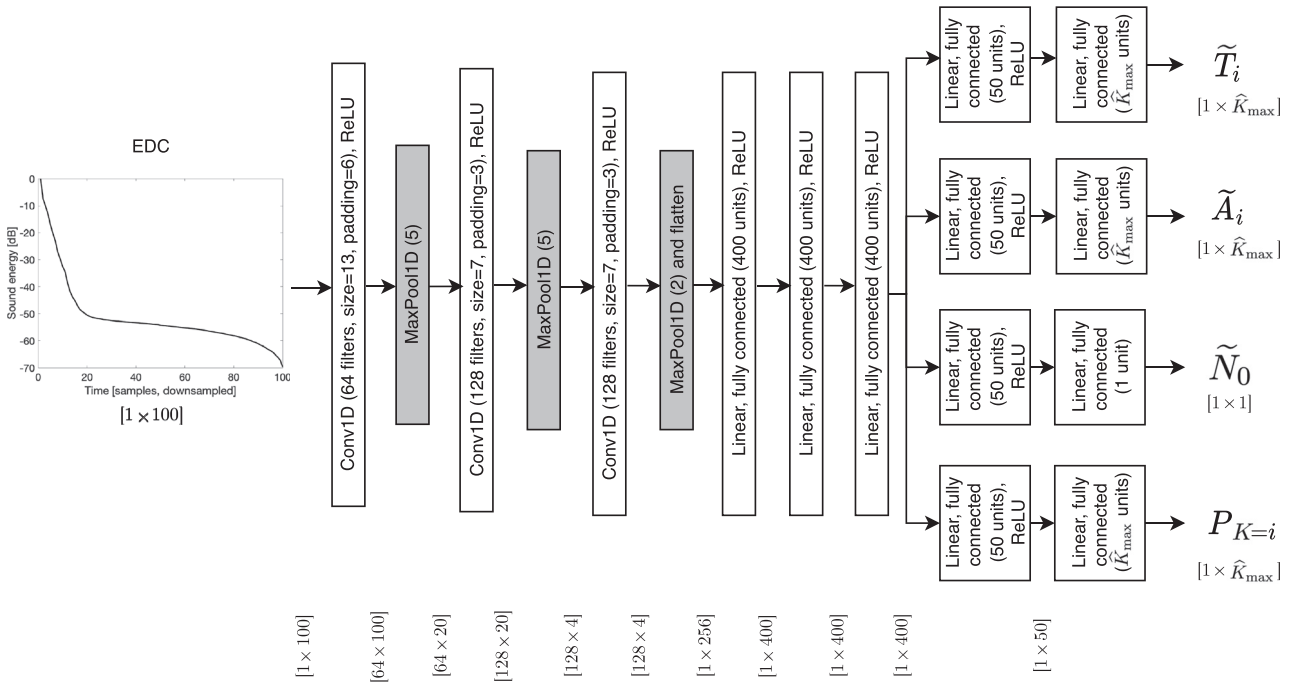

FIG. 1. An example of the proposed neural network structure for the parameter estimation of the sound decay model in Eq. (2). Throughout this paper, we refer to this particular architecture as DecayFitNet. The network outputs are preliminary values for the decay times, $\widetilde{T}_i$, decay amplitudes, $\widetilde{A}_i$, and noise term, $\widetilde{N}_0$. Additionally, the network returns the values $P_{K=i}$, which quantify the EDF model order prediction. The input length and maximum model order for the current implementation are $M = 100$ samples and $\widehat{K}_{\max} = 3$, respectively.

https://doi.org/10.1121/10.0013416

for the task. The max-pooling layers behave like a downsampling operation that focuses on the most active features, thus, reducing dimensionality and redundancies. Consecutive blocks represent progressively higher-level features of the input signal. The shared fully connected layers recombine the extracted features in a nonlinear way. Finally, the task-specific fully connected layers act as independent regressors.

Using a shared common core for all of the tasks has several advantages over using independent networks for each task. First, this reduces the system complexity for training and inference as the number of computations is reduced. More importantly, the shared core will tend to learn useful features across all of the tasks and not overfit any of them.[39] This property can improve generalization and robustness to noisy data.

B. Preprocessing

A neural-network-based structure for decay analysis requires some preprocessing steps. In the following, we detail the preprocessing steps for our example implementation DecayFitNet.

As the network requires a fixed-length input, we propose three preprocessing steps on the EDF $d(t)$ to be analyzed:

(i) The last 5% of the EDFs are excluded, i.e., the samples $d(t > 0.95L)$ are discarded. This step is motivated by the statistical uncertainty of the last EDF samples. This uncertainty is inherent to the Schroeder backward integration method because at the end of the EDF, only a few RIR samples are integrated.

(ii) The result is resampled to a fixed length, M. In our example implementation DecayFitNet, we set $M = 100$ samples. We apply fractional resampling. In fractional resampling, a sample rate conversion by a factor α/β is achieved by first upsampling by a factor α, followed by downsampling by a factor β.[40] In our implementation, $\alpha = M$ and $\beta = L$.

(iii) The EDF is converted into logarithmic scale (in dB) and normalized with the biggest absolute EDF sample value of the entire training dataset. The normalization ensures that every EDF sample of the training dataset lies in the interval $[-1, 1]$. The required normalization factor must be saved to a normalization file after the training procedure is completed. This way, the normalization can also be applied during inference.

C. Postprocessing. of estimates

The preliminary parameter estimates are processed with the following output transformations to yield the final parameter estimates $\widehat{T}_i$, $\widehat{A}_i$, and $\widehat{N}_0$:

$$\widehat{K} = \mathrm{argmax}_i(P_{K=i}), \tag{4a}$$

$$\widehat{T}_i = \frac{\widetilde{T}_i^2 + 1}{M}\frac{L}{f_s}, \tag{4b}$$

$$\widehat{A}_i = \begin{cases} \widetilde{A}_i^2 & \text{for } i \le \widehat{K}, \\ 0 & \text{for } i > \widehat{K}, \end{cases} \tag{4c}$$

$$\widehat{N}_0 = \frac{M}{L}\, 10^{-\widetilde{N}_0}. \tag{4d}$$

We use these transformations to ensure that the final predictions $\widehat{T}_i > 0$ s, $\widehat{A}_i \ge 0$, and achieve better numerical stability during the training while covering a large dynamic range of background noise levels $\widehat{N}_0$. Whereas $\widehat{T}_i$ is the estimated decay time in seconds, the preliminary estimate $\widetilde{T}_i$ is the decay time in samples. Therefore, the value $(\widetilde{T}_i^2 + 1)/M$ is the decay time relative to the neural network input length, M, where squaring and adding one results in $\widehat{T}_i > 0$, thus, avoiding division by zero in the exponential terms [cf. Eq. (2)]. Due to the resampling in the preprocessing stage, the decay time estimates must be readjusted to the original timescale by multiplying with L/f_s. For the same reason, the noise value predictions are scaled by M/L. The amplitudes of all of the exponential terms that have a higher order than the predicted model order $\widehat{K}$ are set to zero, hence, effectively removing their contribution to the predicted EDF fit. Although the amplitude values, A_i, can cover a large dynamic range, our preliminary experiments found that the training converges to better results when we predict $\widehat{A}_i$ on a linear scale as opposed to the logarithmic scale that we use to predict the noise values $\widehat{N}_0$.

The estimated fit, $\widehat{d}_{\widehat{K}}(t)$, is obtained by inserting the network predictions, $\widehat{T}_i$, $\widehat{A}_i$, and $\widehat{N}_0$, into Eq. (2).

D. Synthetic training dataset

A large quantity of EDFs with various combinations of the decay parameters, $\boldsymbol{\theta}$ [cf. Eq. (2)], is required to train the proposed neural network structure. Synthetic EDFs are an efficient way of collecting many different EDFs that cover the variety of real-world measurements. In the following, we want to detail how such a training dataset can be synthesized.

A large fully synthetic dataset of 300 000 EDFs was generated to train the DecayFitNet. It can be split into three equally sized subsets, consisting of EDFs with model orders of 1, 2, and 3. The data generation included three steps. First, we randomly assigned values for the decay parameters, $\boldsymbol{\theta}$. Second, for model orders larger than 1, we checked if the drawn values were different enough to produce a proper multi-slope EDF and redraw values if necessary. Last, Gaussian noise was octave-band filtered, shaped, and backward-integrated to obtain a synthetic EDF with the previously drawn $\boldsymbol{\theta}$ values. Details about these steps are elaborated in the following.

Values for the decay times, T_i, were drawn from the uniform distribution,

$$T_i \in \mathcal{U}(0.1T_{\mathrm{EDF}},\ 1.5T_{\mathrm{EDF}}), \tag{5}$$

where $T_{\mathrm{EDF}} = L/f_s$. In other words, the drawn T_i values are between 10% and 150% of the input EDF length, T_{EDF}.

For our training dataset, we set $T_{\mathrm{EDF}} = 10$ s and $f_s = 48$ kHz. Due to the resampling of arbitrary length EDFs during inference, our dataset allows predicting different T_i value ranges for different input EDF lengths. For instance, for a 2.5 s long input EDF, the network can predict T_i values with lower and upper limits corresponding to 0.1×2.5 s $= 0.25$ s and 1.5×2.5 s $= 3.75$ s, respectively. In other words, the resampling allows our network to operate on various reverberation time ranges, depending on the length of the input EDF. A brief discussion on the generalization of network predictions to EDFs with decay times outside of the range specified in Eq. (5) can be found in Appendix A 1. After drawing the values, they were ordered such that $T_{i+1} \geq T_i$.

In our training dataset, we wanted to cover an extensive dynamic range. Therefore, we assigned

$$N_0 = 10^{a_{\mathrm{noise}}} \quad \text{with} \quad a_{\mathrm{noise}} \in \mathcal{U}(-14, -2), \tag{6}$$

corresponding to noise values between -140 and -20 dB.

As the decay amplitudes, A_i, typically cover a large dynamic range as well, we assigned the values as

$$A_i = 10^a \quad \text{with} \quad a \in \mathcal{U}(-4.5, 0), \tag{7}$$

corresponding to amplitude values between -45 and 0 dB. The A_i values were normalized such that they sum up to unity, i.e., $\sum_{i=1}^{K} A_i = 1$. Finally, they were ordered such that $A_i \geq A_{i+1}$.

If the desired model order, K, was larger than 1, the initial T_i and A_i values were checked for a sufficient multi-slope characteristic. This step was crucial because the random assignment could result in almost identical T_i and A_i values for the different slopes, thus, generating a single-slope EDF, although a multi-slope EDF was desired. Therefore, we introduced the constraints

$$T_{i+1} \geq 1.5\, T_i, \tag{8a}$$

$$\frac{A_i}{A_{i+1}} \geq 10. \tag{8b}$$

The constraint of Eq. (8a) ensured that the different slopes have considerably different decay rates. By applying the constraint of Eq. (8b), we aimed to distribute the amplitude values, A_i, over the entire range from -45 to 0 dB, thus, preventing very similar values. New values for T_i and A_i were randomly drawn from the above distributions until both constraints were fulfilled. A visual inspection of some resulting EDFs showed that both constraints together ensured a distinct multi-slope character. Both constraints are supposed to improve the neural network training because the resulting EDFs should teach the neural network what multi-slope EDFs look like. A discussion on the generalization of network predictions to EDFs that do not satisfy these constraints can be found in Appendix A 2.

Instead of directly inserting the resulting $\boldsymbol{\theta}$ values into Eq. (2), the final synthetic EDF was generated by applying the backward integration [cf. Eq. (1)] on decaying Gaussian noise. This additional step introduced small random fluctuations into the synthetic EDFs to improve generalization after training the neural network. Four steps were necessary to obtain the final synthetic EDF. First, the A_i and N_0 values had to be scaled to account for the backward integration,

$$A_{i,\mathrm{synth}} = 13.8 \cdot T_{\mathrm{EDF}} \frac{A_i}{T_i}, \tag{9a}$$

$$N_{0,\mathrm{synth}} = M N_0. \tag{9b}$$

Second, a synthetic energy response was generated as

$$h^2_{\mathrm{synth}}(t) = N_{0,\mathrm{synth}} \cdot (f_s \cdot T_{\mathrm{EDF}} - t) \cdot g_0^2(t) + \sum_{i=1}^{K} A_{i,\mathrm{synth}} \cdot g_i^2(t) \cdot e^{-13.8t/(f_s T_i)}, \tag{10}$$

where $g_0(t), g_1(t), \ldots, g_K(t)$ is Gaussian noise that is filtered with a random octave-band filter and re-normalized to zero mean and unit variance. For the data generation, we assumed that $f_s = 48$ kHz and $T_{\mathrm{EDF}} = 10$ s. Third, the response, $h_{\mathrm{synth}}(t)$, was backward-integrated according to Eq. (1). Last, just as described in Sec. IV B, the last 5% of the EDF samples was discarded, and the result was resampled to a length of $M = 100$ samples.

E. Loss function

We propose to use a loss function consisting of three parts for training the proposed neural network structure.

The first part is the *EDF loss*, defined as the mean absolute error (MAE) between the analyzed EDF, $d_{\mathrm{dB}}(t)$, and the estimated fit, $\widehat{d}_{\widehat{K},\mathrm{dB}}(t)$,

$$\mathcal{L}_{\mathrm{EDF}} = \frac{1}{M} \sum_{t=0}^{M-1} |d_{\mathrm{dB}}(t) - \widehat{d}_{\widehat{K},\mathrm{dB}}(t)|, \tag{11}$$

where the last 5% of EDC samples is excluded for both EDFs (cf. Sec. IV B), both EDFs are converted to a logarithmic scale (in dB), and $|\cdot|$ denotes the absolute value.

The second part of the loss is the *noise loss*, defined as the absolute error between the ground truth and the estimated noise exponent,

$$\mathcal{L}_{\mathrm{noise}} = |\log_{10}(N_0) - \log_{10}(\widehat{N}_0)|. \tag{12}$$

The third part of the loss is the *model order loss*, defined as the cross-entropy loss

$$\mathcal{L}_{\mathrm{order}} = -P_{K=\widehat{K}} + \ln\left(\sum_{i=1}^{\widehat{K}_{\max}} e^{P_{K=i}}\right). \tag{13}$$

In Eq. (13), $P_{K=i}$ quantifies which probability the network assigns to the model order, i, where K is the true model order. We include the model order loss to teach the network

https://doi.org/10.1121/10.0013416

to predict the correct number of slopes in an EDF. This measure should prevent the network from outputting multiple similar slopes in cases where a single slope would fit the EDF sufficiently well.

Finally, the total loss, $\mathcal{L}$, for training the proposed neural network is

$$\mathcal{L} = \mathcal{L}_{\text{EDF}} + \mathcal{L}_{\text{noise}} + \mathcal{L}_{\text{order}}. \tag{14}$$

F. Training

We train the DecayFitNet for 200 epochs using the Adam optimizer[41] with an initial learning rate of 2×10^{-3} and a weight decay[42] of 6×10^{-5}. Additionally, we apply cosine annealing with warm restarts[43] using a schedule of 40 epochs between the restarts.

V. EVALUATION

In our evaluation, we use the DecayFitNet on two large datasets and compare its EDF fitting performance with a publicly available toolbox and our own implementation of the Bayesian decay analysis framework. This section describes the details of our evaluation.

A. Baseline methods

We compare the DecayFitNet with two other decay analysis approaches. The first baseline method is based on a nonlinear regression model of a single exponential and a noise term.[10] The second baseline method is based on slice sampling for Bayesian decay analysis.[20] Details about the implementation of both methods are presented in the following.

1. Nonlinear regression

This baseline method uses the publicly available toolbox implemented by Karjalainen *et al.*[10] In initial experiments, we found that the performance of the toolbox depends considerably on the choice of the fitting scale. This issue was already observed by Karjalainen *et al.*, which is why they proposed to fit the EDF, $d(t)$, on a power scale.[10] This means that the nonlinear regression is carried out on the scaled EDF, $d_{\text{scale}}(t) = d^s(t)$, thus, adding the adjustable hyperparameter, s, to the method. In our evaluation, we use $s = 0.5$ as suggested by the developers of the toolbox.[10] Additionally, we use an improved variant, where a grid search is carried out over the interval $s \in [0.2, 0.8]$ to find the best fit regarding the mean squared error (MSE). In both of the variants, we only use the EDF below -5 dB, which is common practice for reverberation time estimation.

2. Bayesian decay analysis

For the second baseline method of our evaluation, we implemented a slice-sampling-based Bayesian decay analysis[20] in MATLAB and PYTHON. The code is contained in our decay analysis toolbox (cf. Sec. VII). Our implementation is based on the fully parameterized Bayesian formulation using the likelihood, $\ell_K(\boldsymbol{\theta})$, defined as[20]

$$\ell_K(\boldsymbol{\theta}) = \Gamma\left(\frac{L}{2}\right) \frac{[2\pi\mathcal{E}_K(\boldsymbol{\theta})]^{-L/2}}{2}, \tag{15}$$

where $\Gamma(\cdot)$ is the gamma function, L is the number of EDF samples, and $\mathcal{E}_K(\boldsymbol{\theta})$ quantifies the error between the measured EDF, $d(t)$, and the model, $d_K(t, \boldsymbol{\theta})$, as defined in Eq. (2),

$$\mathcal{E}_K(\boldsymbol{\theta}) = \frac{1}{2} \sum_{t=0}^{L-1} [d(t) - d_K(t, \boldsymbol{\theta})]^2. \tag{16}$$

No prior information about the parameter values, $\boldsymbol{\theta}$, is available before the decay analysis. Therefore, we assign a uniform prior and estimate the model parameters by maximizing the likelihood, $\ell_K(\boldsymbol{\theta})$, over the parameter space.

For this purpose, we apply the slice sampling algorithm[20] because a grid search over all of the parameter combinations would be computationally infeasible. In our analysis, we let the slice sampling algorithm run for at most 1000 iterations or until the first and second moment of the decay parameters have converged. More precisely, we determine convergence as proposed by Jasa and Xiang, i.e., as the iteration when the first and second moment of the decay parameters change less than 0.1% compared to the previous iteration.[20] We restrict the search space analogously to the training dataset of the DecayFitNet, i.e., as

$$T_i \in \mathcal{U}(0.15 \text{ s}, 3.75 \text{ s}), \tag{17a}$$

$$A_i = 10^a \quad \text{with} \quad a \in \mathcal{U}(-4.5, 0), \tag{17b}$$

$$N_0 = 10^{a_{\text{noise}}} \quad \text{with} \quad a_{\text{noise}} \in \mathcal{U}(-14, -2), \tag{17c}$$

where each dimension is discretized into 200 points (equally spaced in T_i, logarithmically spaced in A_i and N_0). In the first iteration, the decay parameters are initialized with random values from the search space. The algorithm proceeds by repeatedly sampling each decay parameter in turn.

In our evaluation, we use this framework to fit models of orders $K = 1, 2, 3$ to the measured EDFs. As suggested in previous work,[19,44] we determine the lowest possible model order that fits the data well by evaluating the Bayesian evidence, $\mathcal{Z}_K$, for different model orders,

$$\mathcal{Z}_K = \int_{\boldsymbol{\theta}} \ell_K(\boldsymbol{\theta})\, \Pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}, \tag{18}$$

where the likelihood is integrated over the entire parameter space, and we assume a uniform prior $\Pi(\boldsymbol{\theta})$. This formulation leverages the full potential of the slice-sampling algorithm because a large number of parameter-likelihood-combinations are determined during the search process, which can be used to calculate the evidence. By choosing the model with the largest evidence, the Bayesian framework balances the degree of fit and a potential over-parameterization.[19,44]

B. Datasets

We use two large RIR datasets for our evaluation. The first dataset is the *Motus dataset*,[45,46] which contains 3320 RIRs measured in various acoustic conditions. More precisely, the RIRs were measured in a single room (approximate volume of 60 m^3), where the furniture amount and placement were varied between measurements to generate 830 unique furniture combinations. The dataset features reverberation times between 0.5 and 2 s at 1 kHz. Due to the complex geometries, the dataset also features acoustic wave phenomena such as scattering. While most of its RIRs have a single-slope characteristic, only approximately 40 RIRs have pronounced multi-slope EDFs due to nonuniform absorption material distributions.

The second dataset is the *Room Transition dataset*.[47,48] It contains measurements of four room transitions with a spatial resolution of 5 cm, including positions with occluded line-of-sight between source and receiver. Due to the room coupling, a large number of EDFs in the dataset exhibit a multi-slope characteristic, where the amplitudes of the slopes vary considerably with the receiver position between the rooms. Furthermore, it was shown that the Room Transition dataset features complex acoustic phenomena of coupled room transitions, such as the portaling effect and distinctive direct-to-reverberant ratio transitions.[47] Our evaluation excluded the transition "office to anechoic chamber," thus, using only 1212 RIRs of the dataset. The remaining room transitions are "meeting room to hallway," "office to kitchen," and "office to stairwell." We cut away the last 0.1 s of all of the RIRs because they include a Hanning fade-out window that disturbs the fitting process.[49]

Both datasets contain higher-order Ambisonic RIRs. The following analyses are based on the omnidirectional channel and the six octave bands from 125 to 4000 Hz. A preliminary analysis of the estimated $\widehat{N}_0$ values showed that both datasets have similar average signal-to-noise ratios of approximately 65 dB.

C. Results

We evaluated the DecayFitNet and Baseline methods with respect to their fitting performance. The results are presented in this section.

1. Example fits

Figure 2 shows example fits obtained with the proposed DecayFitNet for two measured EDFs and also includes the corresponding fits obtained with the previously described grid-search variant of the Karjalainen toolbox and our implementation of the slice-sampling-based Bayesian decay analysis. All of the approaches fit the single-slope EDF of Fig. 2(a) equally well. Figure 2(b) depicts the resulting fits for a multi-slope EDF. The fits obtained with the proposed DecayFitNet architecture and Bayesian decay analysis show good agreement with the measured EDF. In contrast, the Karjalainen toolbox is based on a single exponential plus noise model, thus, being unable to fit EDFs with more than one slope. In the depicted scenario, the Karjalainen toolbox returns a slope between the two distinct slopes. Additionally, it overestimates the noise floor.

2. Error analysis

Table I summarizes the fitting results on the entire datasets and shows medians and 99% quantile values of the decibel-based mean squared error (dB-MSE) between measured octave-band filtered EDFs, $d_{\mathrm{dB}}(t)$, and obtained fits, $\widehat{d}_{\widehat{K},\mathrm{dB}}(t)$. In this paper, we define the dB-MSE as

$$\text{dB-MSE} = \frac{1}{L}\sum_{t=0}^{L-1}\left[d_{\mathrm{dB}}(t) - \widehat{d}_{\widehat{K},\mathrm{dB}}(t)\right]^2, \tag{19}$$

where both EDFs are converted to a logarithmic scale (in dB) for calculating the dB-MSE. A dB-MSE value of 0 dB corresponds to a perfect fit between modeled and analyzed EDF.

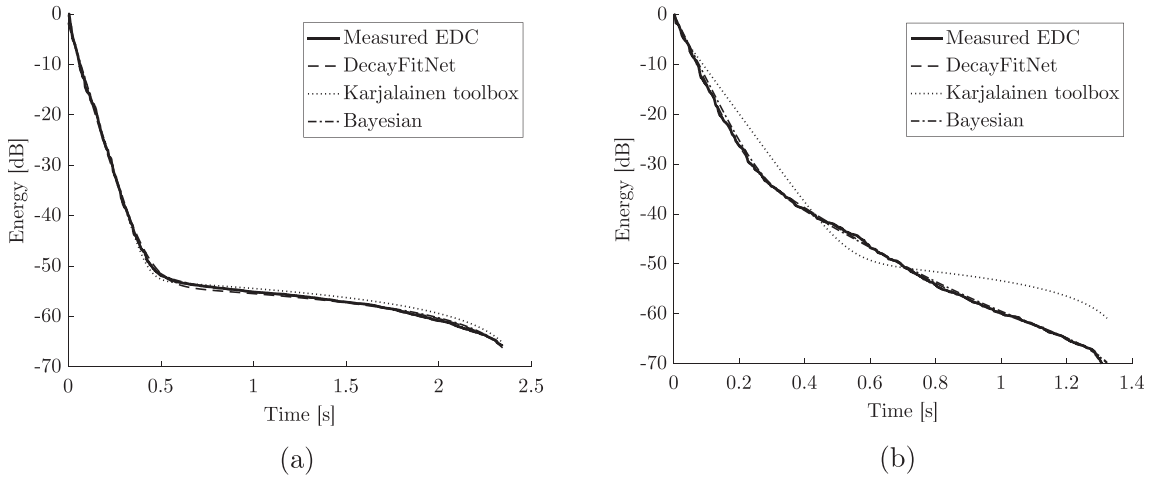

FIG. 2. Examples of fitting (a) a single-slope (from Motus dataset, measurement No. 500, loudspeaker 1, 1 kHz octave-band) and (b) a multi-slope EDF (from Room Transition dataset, meeting room to hallway, source in room, no line-of-sight, 25 cm position, 1 kHz octave-band). The measured EDFs are fitted with the proposed DecayFitNet, the grid-search variant of the Karjalainen toolbox (Ref. 10), and our implementation of the slice-sampling-based Bayesian decay analysis (Ref. 20). Analogously to the rest of our evaluation, the last 5% of EDF samples are excluded from the plot.

https://doi.org/10.1121/10.0013416

TABLE I. The median and 99% quantiles of the dB-MSE [cf. Eq. (19)] between analyzed EDF and estimated fits. A dB-MSE value of 0 dB corresponds to a perfect fit between modeled and analyzed EDF. We compare the proposed DecayFitNet with the publicly available toolbox by Karjalainen *et al.* (Ref. 10) and our own implementation of the slice-sampling-based Bayesian decay analysis (Ref. 20). The different fitting approaches are evaluated on two large, publicly available datasets.

	Karjalainen (standard)		Karjalainen (grid)		Bayesian (slice sampling)		DecayFitNet	
Dataset	median	99% q.	median	99% q.	median	99% q.	median	99% q.
Motus	0.68 dB	6.43 dB	0.56 dB	5.12 dB	0.20 dB	1.13 dB	0.15 dB	1.02 dB
Room Transition	0.89 dB	45.56 dB	0.73 dB	22.53 dB	0.37 dB	2.08 dB	0.23 dB	1.63 dB

The last 5% of EDF samples is excluded from the dB-MSE calculation, following the reasoning described in Sec. IV B. Furthermore, 11 RIRs of the Motus dataset were excluded from the analysis because they featured transient noise artifacts. Such artifacts introduce large discontinuities into the corresponding EDFs, which violate the model of Eq. (2). Consequently, all of the tested fitting approaches had large dB-MSEs for EDFs with such artifacts, allowing us to detect and exclude these measurements.

Table I indicates that the Karjalainen toolbox, its grid-search variant, the slice-sampling-based Bayesian analysis, and proposed DecayFitNet accurately fit the Motus dataset EDFs with median dB-MSEs of 0.68 dB or lower. However, the 99% quantile values indicate that the spread of achieved dB-MSE values is higher for the Karjalainen toolbox than for the proposed DecayFitNet and Bayesian analysis. This suggests that the proposed DecayFitNet achieves slightly more robust fitting on a dataset with mostly single-slope EDFs than a well-established single-slope fitting toolbox. Furthermore, its performance, in terms of median errors, is comparable to an existing multi-slope fitting approach.

The results for the Room Transition dataset show a similar trend. While median values do not differ considerably between the four approaches, the standard Karjalainen toolbox and its grid-search variant exhibit increased 99% quantile values of 45.56 and 22.53 dB, respectively. This considerable variability in fitting performance can be attributed to the insufficient model order for fitting multi-slope EDFs. In contrast, the 99% quantile value for fitting the Room Transition dataset with the proposed DecayFitNet is 1.63 dB and, therefore, only slightly higher than for the Motus dataset. The Bayesian analysis achieves a similar performance. This result suggests that the DecayFitNet can robustly fit large quantities of multi-slope EDFs.

Figure 3 supports these findings. It shows violin plots of the dB-MSE values that were the basis for the calculations of Table I. Figure 3(a) shows that all of the approaches achieve low dB-MSE values for the Motus dataset across all of the frequency bands. Most dB-MSE values lie below 10 dB, and the spread of values below 10 dB is slightly bigger for the Karjalainen toolbox than for the DecayFitNet and Bayesian analysis. Figure 3(b) shows the results for the Room Transition dataset. The plots exhibit larger dB-MSE values for the two Karjalainen toolbox approaches with many data points well above 10 dB. A more thorough analysis of the high dB-MSE values revealed that they can be attributed to multi-slope EDFs, for which the model order of the Karjalainen toolbox is too low. In contrast, the proposed DecayFitNet and Bayesian analysis achieve similarly low fitting errors for the Room Transition dataset with all of the dB-MSE values below 10 dB.

VI. DISCUSSION

The results presented in Sec. V C indicate that the proposed neural network structure can robustly fit single-slope and multi-slope EDFs on large datasets without prior parameter tuning or supervision.

Our evaluation was based on two large datasets of more than 1000 RIRs each, corresponding to more than 20 000 EDFs across six octave bands in total. They feature various acoustic conditions such as varying amounts and placements of absorptive materials, diffraction and scattering from the room geometry, and room coupling. On both evaluated datasets, the DecayFitNet and Bayesian decay analysis outperform the toolbox by Karjalainen *et al.*,[10] which is a well-established toolbox for fitting EDFs with a single slope and a noise term. It is important to note that the Karjalainen toolbox cannot fit multi-exponential decays, thus, explaining the degraded performance on the Room Transition dataset, which contains many multi-slope EDFs. In contrast, the Karjalainen toolbox performed well on the Motus dataset, which primarily contains single-slope decays. Nevertheless, it exhibited a considerably larger spread of errors than the DecayFitNet and Bayesian decay analysis. The latter two approaches performed similarly well on both datasets.

To the best of the authors' knowledge, at this point, there is no study examining the performance of state-of-the-art decay analysis algorithms on large amounts of data. We found that the slice-sampling-based Bayesian analysis performs well on large-scale datasets, despite the iterative nature of the approach. As a fully deterministic alternative approach, we have proposed a neural network structure. With the presented DecayFitNet, potential numerical difficulties have moved from the application (and the user) to the network training stage. The network training has to be performed only once, and the pre-trained network can subsequently be used by users without any additional effort. This shift decreases the required user tuning and also reduces the computational complexity at inference time. The efficiency and low user tuning of the neural network structure are balanced by the decreased possibilities of adjusting an incorrectly fitted EDF. Whereas other approaches have adjustable

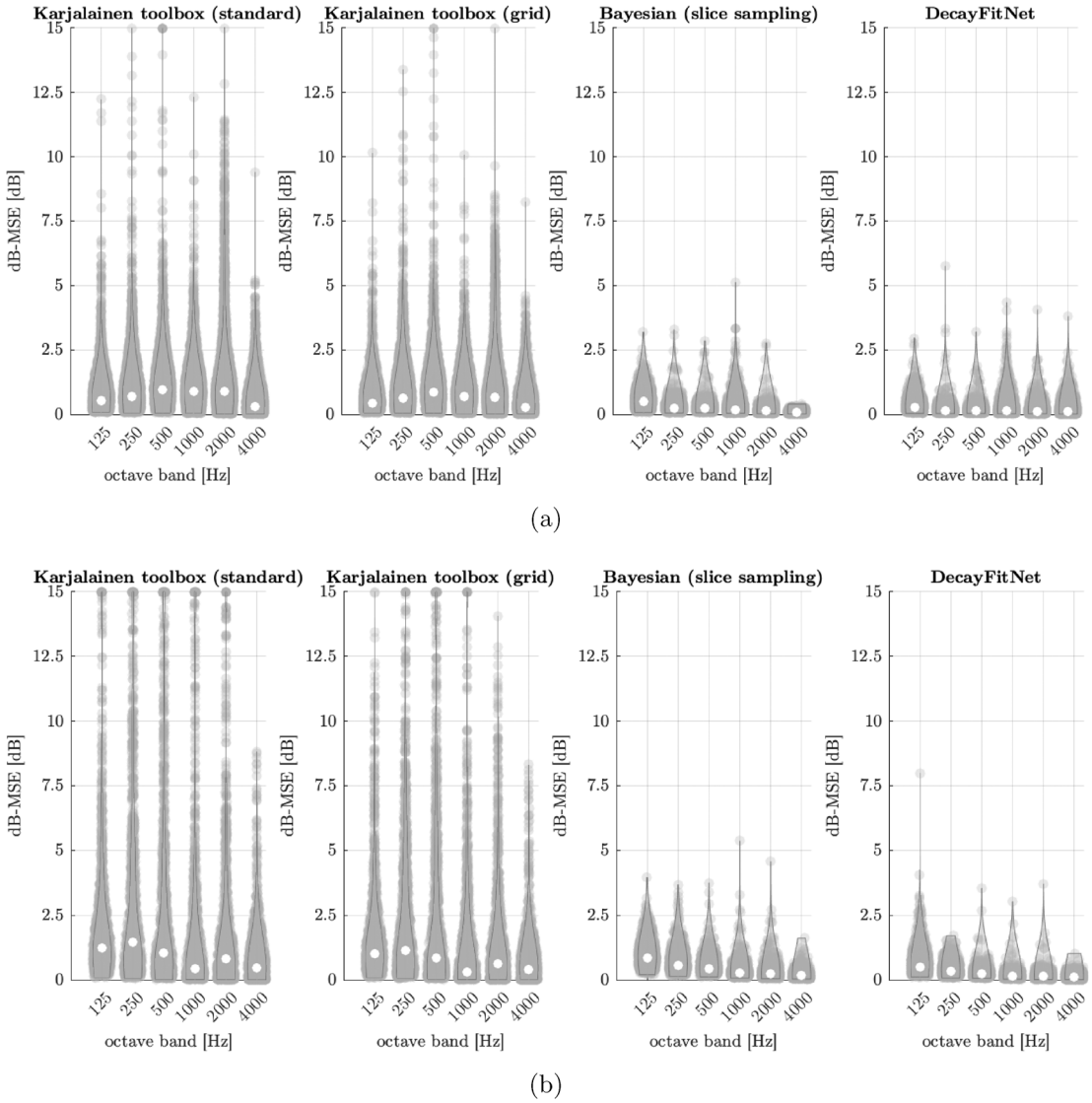

FIG. 3. Violin plots of the dB-MSE [cf. Eq. (19)] between measured octave-band-filtered EDFs and the corresponding fits obtained with different approaches. The evaluation is based on the (a) Motus dataset and the (b) Room Transition dataset. Light gray circles indicate individual data points, and white circles indicate the median. dB-MSE values greater than 15 dB have been excluded from the plot for clarity. A dB-MSE value of 0 dB corresponds to a perfect fit between modeled and analyzed EDFs.

parameters to fix a wrong EDF fit, the output of the neural network cannot be changed unless it is trained with new data.

Last, it is interesting to note that the DecayFitNet can analyze the entire Motus dataset with almost 20 000 EDFs in less than 30 s. The evaluation was carried out on a modern laptop computer (model year 2019) without a dedicated GPU. We obtained this result by averaging the elapsed time of 100 runs over the entire dataset to account for temporary drops of processing power that could bias the result. However, it is important to note that the presented number should only be understood as an orientation for the reader. It is obvious that optimized implementations are an interesting engineering task for future work, which could reduce the computation times even further.

For example, due to the CNN-based architecture, our approach can easily be scaled up and parallelized to efficiently process large amounts of data with modern GPUs. The execution time of the DecayFitNet depends largely on the number of model parameters, i.e., the learned weights and biases of the different layers. With approximately 677 000 parameters, the DecayFitNet is rather lightweight. The high computational efficiency is an important step toward bringing decay analysis algorithms to mobile devices. Furthermore, it benefits the processing of large datasets, where GPU-accelerated machines can leverage the full potential of our CNN-based architecture. Additionally, the DecayFitNet is completely deterministic at inference time because it always applies the same set of parameters to its inputs. This property limits the number of executed operations to a fixed value and ensures that the same results are obtained for repeated runs.

VII. DECAY ANALYSIS TOOLBOX

As part of this work, we provide an open-source toolbox for PYTHON and MATLAB. The toolbox includes a pretrained

https://doi.org/10.1121/10.0013416

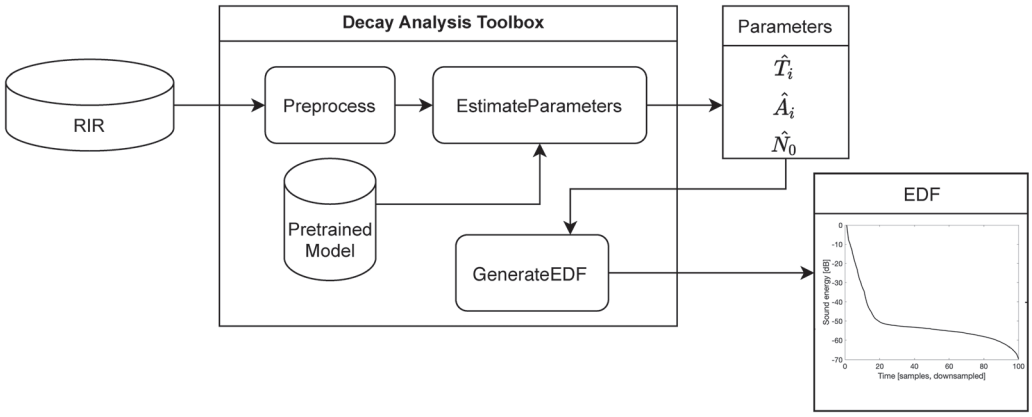

FIG. 4. Workflow and structure of the decay analysis toolbox to estimate the EDF parameters from a RIR.

neural network model, a code interface to interact with the model, our implementation of the slice-sampling-based Bayesian decay analysis, as well as some utility functions. The main goal of the toolbox is to present a simple package to estimate multi-slope EDFs that can be used with minimal code. Although a pretrained model is provided, the toolbox also includes the ability to train and export a model using a custom dataset. This could be useful in scenarios where the generalization of the pretrained model is not accurate enough. The source code and documentation for the toolbox are available online.[51]

The structure and workflow of the toolbox are presented in Fig. 4. The main functionality includes

- impulse responses preprocessing for the neural-network-based decay analysis (as described in Sec. IV B);
- estimation of parameters, $\boldsymbol{\theta}$, from Eq. (2) by doing a forward pass of the pre-trained model or by applying the slice-sampling-based Bayesian decay analysis;
- generation of the fitted EDF using the estimated parameters; and
- training and export of the DecayFitNet neural network model.

The toolbox exports the neural network model and operates using the Open Neural Network Exchange format (ONNX).[50] ONNX is an open and common format that allows for fully trained machine learning models to be distributed and utilized by a variety of frameworks and platforms. Although the toolbox only includes interfaces for PYTHON and MATLAB, a trained model exported to the ONNX format can be supported by many applications, requiring only the porting of the preprocessing and EDF generation code. The provided model was trained using the synthetic dataset described in Sec. IV D.

VIII. CONCLUSIONS

This paper proposed a neural network structure for fitting multi-exponential EDFs. It was shown that such a network can be trained on a dataset of synthetically generated EDFs. We presented the DecayFitNet as an example architecture of the proposed approach. A large-scale evaluation applied the DecayFitNet and two comparable state-of-the-art methods on two large datasets of real-world measurements with more than 1000 RIRs each, corresponding to over 20 000 EDFs across six octave bands. The analyzed datasets featured various acoustic conditions, such as varying amounts of absorptive material, diffraction and scattering from the room geometry, and room coupling.

The results of our evaluation indicate that the proposed neural network structure can robustly fit single-slope and multi-slope EDFs without prior parameter tuning or supervision by the user. Additionally, the presented DecayFitNet is fully deterministic during inference time and computationally efficient and capable of analyzing almost 20 000 EDFs in less than 30 s on a modern laptop computer (2019) without a dedicated GPU. Our evaluation indicates that the DecayFitNet robustly estimates the model parameters from large datasets of measured EDFs while achieving a comparable fitting performance compared to state-of-the-art multi-slope decay analysis algorithms. A decay analysis toolbox has been made publicly available for the audio community.

The DecayFitNet and its corresponding toolbox may benefit future research on room acoustic analysis and modeling. Data-heavy machine-learning-based approaches may leverage its full potential regarding computational efficiency and robustness.

ACKNOWLEDGMENTS

This research has received funding from the Aalto University Doctoral School of Electrical Engineering and the Academy of Finland, Project No. 317341. We acknowledge the computational resources provided by the Aalto Science-IT project.

APPENDIX A: NEURAL NETWORK GENERALIZATION

The DecayFitNet is trained with 300 000 different EDFs, split equally into EDFs with one, two, or three slopes. Among this large number of EDFs, there are EDFs with a variety of different decay parameters as outlined in Sec. IV D. The neural network training procedure teaches the

network to deal with all of the possible decay parameter combinations within the specified ranges, thus, making it capable of generalizing to unseen data during inference time. However, the neural network performance may degrade for unseen data with decay parameters far outside of the parameter ranges of the training dataset. In such cases, generalization cannot be guaranteed, but the training data can be easily adapted to make the analyzable parameter ranges larger if required. In these cases, the neural network training has to be performed again.

The generalization to extremely out-of-distribution samples is an interesting and common discussion topic in the machine learning literature. Thoroughly exploring the capabilities of the proposed model for this task is out of the scope of this work. Nevertheless, This appendix gives the reader some intuition about how the network performs on EDFs with decay parameters outside of the training data parameter ranges.

1. EDFs with decay parameters outside of the training data parameter ranges

This section reports on an experiment that investigates the neural network performance for EDFs with decay parameters outside of the training data parameter ranges specified in Sec. IV D. To this end, 10 000 synthetic double-slope EDFs were generated, the decay times of which were randomly drawn from a uniform distribution between 4 and 7 s. The synthetic EDFs had a length of $T_{\mathrm{EDF}} = 2.5$ s. Consequently, the randomly drawn decay times could become significantly larger than the largest decay times contained in the training dataset. The other parameter ranges and constraints remained identical to those specified in Sec. IV D.

Analogously to the other analyses in this paper, we evaluate the fitting performance based on the dB-MSE [cf. Eq. (19)] between the true and modeled EDFs. The median and 99% quantile dB-MSE value over all 10 000 EDF fits amount to 0.07 and 0.50 dB, respectively. This result indicates that the neural network still performs robustly for unseen EDFs that do not exactly fall inside of the parameter range of the training dataset. However, although many EDFs were evaluated in this experiment, it is important to note that the presented analysis should not be understood as a guarantee for successful generalization to such cases. This can also be seen when drawing decay times from parameter ranges even further outside of the training dataset. For example, in a follow-up experiment, we uniformly drew decay times between 7 and 13 s while keeping all of the other experimental parameters identical. In this case, the fitting performances slowly starts to deteriorate as indicated by median and 99% quantile dB-MSE values of 1.2 and 2.1 dB, respectively.

2. EDFs with decay parameters that do not satisfy the constraints of Eqs. (8a) and (8b)

This section reports on an experiment that investigates the neural network performance for EDFs with decay parameters that do not satisfy the constraints in Eqs. (8a) and (8b). To this end, 10 000 synthetic double-slope EDFs were generated, the decay parameters of which were randomly drawn from the parameter ranges specified in Sec. IV D. However, in contrast to the generation of the training dataset, the synthetic EDFs in this experiment were drawn such that they do *not* satisfy the constraints in Eqs. (8a) and (8b). This should provide some orientation on how the neural network reacts to EDFs that violate these constraints.

Analogously to the other analyses in this paper, we evaluate the fitting performance based on the dB-MSE [cf. Eq. (19)] between the true and modeled EDFs. The median and 99% quantile dB-MSE value over all of the 10 000 EDF fits amount to 0.04 and 0.29 dB, respectively. This result indicates that constraining the parameter value ranges of the training dataset does not significantly affect the neural network performance for unseen EDFs that do not satisfy the constraints of Eqs. (8a) and (8b). However, although many EDFs were evaluated in this experiment, it is important to note that the presented analysis should not be understood as a guarantee for successful generalization to such cases.

[1]H. Kuttruff, *Room Acoustics*, 4th ed. (Spon, London, UK, 2000).
[2]M. R. Schroeder, "New method of measuring reverberation time," J. Acoust. Soc. Am. **37**(3), 409–412 (1965).
[3]ISO 3382-1, "Acoustics–Measurement of room acoustic parameters–Part 1: Performance spaces" (International Organization for Standardization, Geneva, Switzerland, 2009).
[4]ISO 3382-2, "Acoustics—Measurement of room acoustic parameters—Part 2: Reverberation time in ordinary rooms" (International Organization for Standardization, Geneva, Switzerland, 2008).
[5]W. T. Chu, "Comparison of reverberation measurements using Schroeder's impulse method and decay-curve averaging method," J. Acoust. Soc. Am. **63**(5), 1444–1450 (1978).
[6]D. R. Morgan, "A parametric error analysis of the backward integration method for reverberation time estimation," J. Acoust. Soc. Am. **101**(5), 2686–2693 (1997).
[7]A. Lundeby, T. E. Vigran, H. Bietz, and M. Vorländer, "Uncertainties of measurements in room acoustics," Acta Acust. Acust. **81**(4), 344–355 (1995).
[8]M. Guski and M. Vorländer, "Measurement uncertainties of reverberation time caused by noise," in *Proceedings of the International Conference on Acoustics (ICA), the 39th Annual Congress of DEGA, and the 40th Annual Congress of AIA* (DEGA, Merano, Italy, 2013), pp. 2067–2070.
[9]N. Xiang, "Evaluation of reverberation times using a nonlinear regression approach," J. Acoust. Soc. Am. **98**(4), 2112–2121 (1995).
[10]M. Karjalainen, P. Antsalo, A. Mäkivirta, T. Peltonen, and V. Välimäki, "Estimation of modal decay parameters from noisy response measurements," J. Audio Eng. Soc. **50**(11), 867–878 (2002).
[11]C. F. Eyring, "Reverberation time measurements in coupled rooms," J. Acoust. Soc. Am. **3**(2), 181–206 (1931).
[12]F. V. Hunt, L. L. Beranek, and D. Y. Maa, "Analysis of sound decay in rectangular rooms," J. Acoust. Soc. Am. **11**(1), 80–94 (1939).
[13]E. Nilsson, "Decay processes in rooms with non-diffuse sound fields Part I: Ceiling treatment with absorbing material," Build. Acoust. **11**(1), 39–60 (2004).
[14]N. Xiang and P. M. Goggans, "Evaluation of decay times in coupled spaces: Bayesian parameter estimation," J. Acoust. Soc. Am. **110**(3), 1415–1424 (2001).
[15]N. Xiang, P. M. Goggans, T. Jasa, and M. Kleiner, "Evaluation of decay times in coupled spaces: Reliability analysis of Bayeisan decay time estimation," J. Acoust. Soc. Am. **117**(6), 3707–3715 (2005).
[16]J. Balint, F. Muralter, M. Nolan, and C.-H. Jeong, "Bayesian decay time estimation in a reverberation chamber for absorption measurements," J. Acoust. Soc. Am. **146**(3), 1641–1649 (2019).

[17]H. Pu, X. Qiu, and J. Wang, "Different sound decay patterns and energy feedback in coupled volumes," J. Acoust. Soc. Am. **129**(4), 1972–1980 (2011).
[18]Z. Sü Gül, E. Odabaş, N. Xiang, and M. Çalışkan, "Diffusion equation modeling for sound energy flow analysis in multi domain structures," J. Acoust. Soc. Am. **145**(4), 2703–2717 (2019).
[19]N. Xiang, P. Goggans, T. Jasa, and P. Robinson, "Bayesian characterization of multiple-slope sound energy decays in coupled-volume systems," J. Acoust. Soc. Am. **129**(2), 741–752 (2011).
[20]T. Jasa and N. Xiang, "Efficient estimation of decay parameters in acoustically coupled-spaces using slice sampling," J. Acoust. Soc. Am. **126**(3), 1269–1279 (2009).
[21]D. Cabrera, J. Xun, and M. Guski, "Calculating reverberation time from impulse responses: A comparison of software implementations," Acoust. Aust. **44**(2), 369–378 (2016).
[22]L. Álvarez-Morales, M. Galindo, S. Girón, T. Zamarreño, and R. M. Cibrián, "Acoustic characterisation by using different room acoustics software tools: A comparative study," Acta Acust. Acust. **102**(3), 578–591 (2016).
[23]B. F. G. Katz, "International round robin on room acoustical impulse response analysis software 2004," Acoust. Res. Lett. Online **5**(4), 158–164 (2004).
[24]M. Alam, M. Samad, L. Vidyaratne, A. Glandon, and K. Iftekharuddin, "Survey on deep neural networks in speech and vision systems," Neurocomputing **417**, 302–321 (2020).
[25]T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," available at https://arxiv.org/abs/1708.02709 (Last viewed August 5, 2022).
[26]H. Purwins, B. Li, T. Virtanen, J. Schluter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," IEEE J. Sel. Top. Signal Process. **13**(2), 206–219 (2019).
[27]P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin, "A survey of sound source localization with deep learning methods," available at https://arxiv.org/abs/2109.03465 (Last viewed August 5, 2022).
[28]J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, "Estimation of room acoustic parameters: The ACE challenge," IEEE/ACM Trans. Audio, Speech, Lang. Process. **24**(10), 1681–1693 (2016).
[29]Z. Fan, V. Vineet, H. Gamper, and N. Raghuvanshi, "Fast acoustic scattering using convolutional neural networks," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2020), pp. 171–175.
[30]S. Wirler, S. J. Schlecht, and V. Pulkki, "Machine learning based auralization of rigid sphere scattering," in *International Conference on Immersive and 3D Audio (I3DA)* (2021).
[31]Z. Fan, V. Vineet, C. Lu, T. W. Wu, and K. McMullen, "Prediction of object geometry from acoustic scattering using convolutional neural networks," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2021), pp. 471–475.
[32]M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle, "Machine learning in acoustics: Theory and applications," J. Acoust. Soc. Am. **146**(5), 3590–3628 (2019).
[33]I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA), available at http://www.deeplearningbook.org (Last viewed August 5, 2022).
[34]M. Fernández-Delgado, M. Sirsat, E. Cernadas, S. Alawadi, S. Barro, and M. Febrero-Bande, "An extensive experimental survey of regression methods," Neural Networks **111**, 11–34 (2019).
[35]D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN), Part III*, Lecture Notes in Computer Science (Springer, Thessaloniki, Greece, 2010), pp. 92–101.
[36]V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Haifa, Israel (2010), pp. 807–814.
[37]A. Pinkus, "Approximation theory of the MLP model in neural networks," Acta Numer. **8**, 143–195 (1999).
[38]Y. LeCun and Y. Bengio, "Convolutional networks for images, speech and time series," in *The Handbook of Brain Theory and Neural Networks*, edited by M. A. Arbib (The MIT Press, Cambridge, MA, 1995), pp. 255–258.
[39]H. Bilen and A. Vedaldi, "Integrated perception with recurrent multi-task neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16 (Curran Associates Inc., Red Hook, NY, 2016), pp. 235–243.
[40]A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. (Prentice Hall, Upper Saddle River, NJ, 1999).
[41]D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, San Diego, CA (2015).
[42]I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, New Orleans, LA (2019).
[43]I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient descent with warm restarts," in *International Conference on Learning Representations (ICLR)*, Toulon, France (2017).
[44]N. Xiang, "Model-based Bayesian analysis in acoustics—A tutorial," J. Acoust. Soc. Am. **148**(2), 1101–1120 (2020).
[45]G. Götz, S. J. Schlecht, and V. Pulkki, "A dataset of higher-order Ambisonic room impulse responses and 3D models measured in a room with varying furniture," in *International Conference on Immersive and 3D Audio (I3DA)* (2021), pp. 1–8.
[46]G. Götz, S. J. Schlecht, and V. Pulkki, "Motus: A dataset of higher-order Ambisonic room impulse responses and 3D models measured in a room with varying furniture" (version 1.0), available at https://zenodo.org/record/4923187#.YuGSc3bMKUk (Last viewed August 5, 2022).
[47]T. McKenzie, S. J. Schlecht, and V. Pulkki, "Acoustic analysis and dataset of transitions between coupled rooms," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2021), pp. 481–485.
[48]T. McKenzie, S. J. Schlecht, and V. Pulkki, "A dataset of measured spatial room impulse responses for the transition between coupled rooms" (version 1.2), available at https://zenodo.org/record/4636068#.YuGT5nbMKUk (Last viewed August 5, 2022).
[49]Hanning windows are frequently used in various signal-processing scenarios (Ref. 40). In the Room Transition dataset, only the right half of the window is used to smoothly fade out the RIR toward the end. This step is useful for auralization purposes because it prevents abrupt offsets at the end of the RIR, which would cause audible artifacts. As this window violates the EDF model defined by Eq. (2), it is cut away for the analyses of this paper.
[50]J. Bai, F. Lu, and K. Zhang, "ONNX: Open neural network exchange," available at https://github.com/onnx/onnx (Last viewed August 5, 2022).
[51]See https://github.com/georg-goetz/DecayFitNet/ (Last viewed August 5, 2022).