Binaural room impulse responses interpolation using physics-informed neural networks in three dimensions

Yazhou Li, Lin Wang, Joshua Reiss

Center for Digital Music, Queen Mary University of London, Email: yazhou.li/lin.wang/joshua.reiss@qmul.ac.uk

Introduction

Binaural audio is needed for headphone reproduction in virtual reality applications. A common way to generate binaural audio is to use head-related impulse responses (HRIRs), known as head-related transfer functions (HRTFs) in the frequency domain, to add spatial cues, and use room impulse responses (RIRs) to add room reverberation to the audio. Binaural room impulse responses (BRIRs) are the combination of HRIRs and RIRs, which contain both reflections in RIRs and the head and pinnae's effects in HRIRs. For dynamic applications, BRIRs at different positions are needed, which are usually synthesized and may not sound authentic. Additionally, accurate BRIRs at different positions are needed for loudspeaker reproduction applications. However, measurements at different positions can be challenging. This calls for the need to interpolate BRIRs to more positions from a small set of measurements.

Traditional DSP methods for BRIRs interpolation use Dynamic Time Warping to find corresponding reflection peaks of two adjacent BRIRs, then do linear time interpolation of the onset times of two peaks, and linear magnitude interpolation of the shapes of two peaks [1]. However, the onset time and magnitude of the reflection peaks may not change linearly, and modeling the nonlinear composition is necessary.

Recently, deep learning has been applied to impulse response interpolation, for both RIRs interpolation and HRIRs interpolation, including generation models, HRTF field [2], and implicit neural representation [3], in both the time domain [4], the time-frequency domain [5], and the combination of the two domains [6]. [7] estimates HRTFs using implicit neural representation in the frequency domain. [8] applies implicit neural representation to BRIRs interpolation. Other works add physics constraints in the loss function to enforce the representation to conform to the wave equation, for both RIRs interpolation [9, 10] and HRIRs interpolation [11]. These methods are called physics-informed neural networks (PINNs).

BRIRs interpolation is more challenging than HRIRs interpolation in that we need to consider longer reflections in the room. BRIRs interpolation is more challenging than RIRs interpolation because the head at different positions might change the room acoustics, and it is difficult to explicitly model the head diffraction effects and incorporate them into the loss function. However, [12] shows even if the physics information is incomplete, the network is still able to learn the implicit physical equation based on the training data.

Also, current PINN implementations focus on 1D and 2D datasets [9, 10] due to memory consumption problems. However, in this case, the sound propagation from the third direction is not included in the wave equation, thus might affect the results.

In this work, we apply PINNs to BRIRs interpolation to evaluate the effectiveness of PINN when all the physics information is not explicitly given. We also extend the implementation to three dimensions and compare the performance using 1D, 2D and 3D datasets to see if training the network in 3D can improve the performance.

Method

Problem formulation Our goal is to estimate BRIRs of N time samples at M spatial positions from measurements of BRIRs of N time samples at $\tilde{M}$ spatial positions using a neural network. The ground truth BRIR is represented as $h(n, \boldsymbol{r})$, and the estimated BRIR is represented as $\hat{h}(n, \boldsymbol{r})$, where $\boldsymbol{r}$ is the spatial position of the measurement microphone in the room, and n is the time instance in the impulse response. $\boldsymbol{r} = (x, y, z)$ when the dataset is in 3D, $\boldsymbol{r} = (x, y)$ when the dataset is in 2D, and $\boldsymbol{r} = x$ when the dataset is in 1D. The network input is $(n, \boldsymbol{r})$, and the network output is $\hat{h}(n, \boldsymbol{r})$. This is shown in Figure 1 and Figure 2.

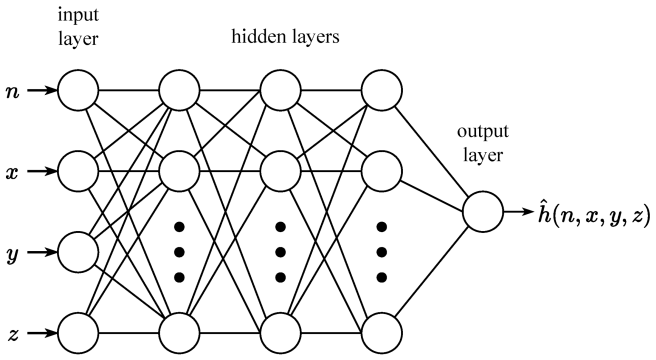

Figure 1: Network architecture when the dataset is in 3D. The input of the network is (n, x, y, z), and the output of the network is h. The input is n, x when the dataset is in 1D and the input is n, x, y when the dataset is in 2D.

We use sinusoidal representation networks for BRIRs interpolation and add physics constraints to improve the performance of the model.

Sinusoidal representation networks Implicit neural representations are neural networks that represent a discrete signal as a continuous function. It can intrinsically

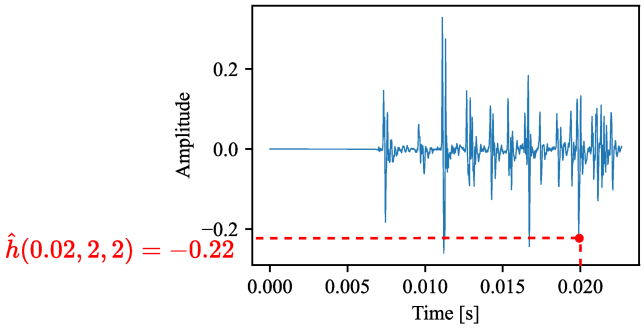

Figure 2: An example of the network input and output in 2D of the BRIR at position (2m, 2m). $\hat{h}(0.02, 2, 2) = -0.22$ is the amplitude at 0.02s in this BRIR.

perform interpolation of a spatially discrete signal and less memory is needed for data storage. By inputting the coordinate in the signal, the network outputs the value at that coordinate. The sine function is used as the activation function, as it can maintain the second derivatives when calculating partial differential equations (PDEs). The network architecture is denoted as sinusoidal representation network (SIREN) [3]:

$$\begin{aligned} \hat{h}(n, \boldsymbol{r}) &= \boldsymbol{W}_n(\phi_{n-1} \circ \phi_{n-2} \circ ... \circ \phi_0)(n, \boldsymbol{r}) + \boldsymbol{b}_n, \\ \phi_i(\boldsymbol{x}_i) &= \sin(\omega_0(\boldsymbol{W}_i \boldsymbol{x}_i + \boldsymbol{b}_i)), \end{aligned} \tag{1}$$

where ϕ_i is the i^{th} layer of the network, $\circ$ is the function composition symbol, $\boldsymbol{x}_i$ is the input to the i^{th} layer, $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ are weight matrices and bias vectors. ω_0 is a hyperparameter that controls the frequency scaling of the sinusoidal activation functions. Specifically, it determines the range of frequencies the network can represent, influencing its ability to model signals with varying levels of detail.

Physics informed neural networks The sound propagation in the room follows the wave equation [13], defined as

$$\frac{\delta^2 \boldsymbol{p}}{\delta t^2} = c^2 \left(\frac{\delta^2 \boldsymbol{p}}{\delta x^2} + \frac{\delta^2 \boldsymbol{p}}{\delta y^2} + \frac{\delta^2 \boldsymbol{p}}{\delta z^2}\right), \tag{2}$$

where $\boldsymbol{p} = p(t, \boldsymbol{r})$ is the sound pressure at time index t and spatial position $\boldsymbol{r}$, c is the sound propagation speed. The wave equation can be understood as the acceleration of the change of the sound field at a time spatial point is proportional to the curvation of the sound field in the spatial space.

PINNs ensure the network not only learns the measurement data, but also obeys the wave equation, so that the neural network is more robust to noise and can learn the data representation using less amount of data. The loss function L of PINNs includes data loss L_{data}, which is the distance between the output and the ground truth, and PDE loss L_{PDE}, which is the distance between the output and the wave equation:

$$\begin{aligned} L &= L_{data} + \alpha L_{PDE}, \\ L_{data} &= \frac{1}{N_d} \sum_{n_d=1}^{N_d} |\hat{h}(n_{n_d}, \boldsymbol{r}_{n_d}) - h(n_{n_d}, \boldsymbol{r}_{n_d})|, \\ L_{PDE} &= \frac{1}{N_f} \sum_{n_f=1}^{N_f} |\Delta^2 \hat{h}(n_{n_f}, \boldsymbol{r}_{n_f}) - \frac{1}{c^2} \frac{\delta^2}{\delta t^2} \hat{h}(n_{n_f}, \boldsymbol{r}_{n_f})|, \end{aligned} \tag{3}$$

where Δ is the Laplacian operator, $N_d = \tilde{M} \times N$ is the number of points for data loss calculation, $N_f = M \times N$ is the number of points for PDE loss calculation. α is a hyperparameter that controls the balance between the data loss and PDE loss.

Experiment

Dataset The BRIRs dataset is created using RAZR simulator [14]. RAZR is a MATLAB toolbox that calculates spatial room impulse responses (SRIRs) by ray tracing acoustic simulation method, then convolves HRIRs from different directions with SRIRs to synthesize BRIRs. We choose the small KEMAR head's HRIRs from the CIPIC dataset [15] for BRIRs synthesis and only BRIRs of the left ear are considered. The head rotation is not considered, and only the head position is considered, so the directions of the head are always towards the right. The room is a shoebox room with the size of 5×4×3 m^3. The loudspeaker is placed at (3m, 1m, 1m). The sampling rate is 44100 Hz and the time length of the BRIR is 1000 instances, which is 22.7 ms.

1D, 2D and 3D datasets are created. For the 1D dataset, $\tilde{M} = 10$, the measurement microphones cover a 1 m line, from (1m, 1m, 1m) to (2m, 1m, 1m). The line is separated into 9 equal parts and two adjacent microphones' distance is $\frac{1}{9}$ m. $M = 20$, the evaluation microphones cover a 1m line, from (1m, 1m, 1m) to (2m, 1m, 1m). The 1m line is separated into 19 equal parts (so that there are two microphones on the two endpoints) and the two adjacent microphones' distance is $\frac{1}{19}$ m. The 2D and 3D datasets are defined similarly in Figure 3.

Training The network architecture is a 3-layer multilayer perceptron (MLP) with 128 neurons in each layer. The loss weight parameter α is 5×10^{-10}. The activation function initialization parameter ω_0 is 30. The network weights initialization scheme is the same as [3]. The inputs are first normalized to [0, 1] before inputting to the network. During the wave equation calculation, the inputs are scaled in order to maintain their physical dimension relationship. That is to say, the spatial input is scaled so that its unit is meter and the time input is scaled so that its unit is second. Pytorch autograd is used to calculate the derivatives of $\hat{h}(n, \boldsymbol{r})$ with respect to time and space.

To reduce the memory consumption by the network, we input only one time index and one spatial position to the network during training. This resolves the memory consumption problem, but on the other hand, it makes the network difficult to converge because the batch size

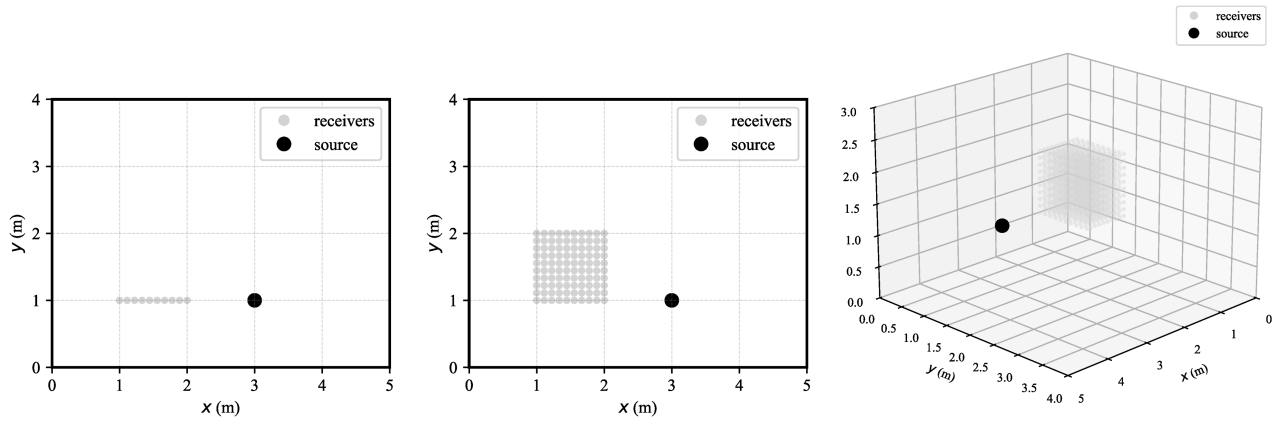

Figure 3: Visualization of the loudspeaker and microphone positions for dataset generation. The gray dots are the microphones and the black dot is the loudspeaker. Left: 1D dataset, $\tilde{M} = 10$, $M = 20$; Middle: 2D dataset, $\tilde{M} = 10 \times 10$, $M = 20 \times 20$; Right: 3D dataset, $\tilde{M} = 10 \times 10 \times 10$, $M = 20 \times 20 \times 20$.

is too small. Therefore, we use a batch size of B, which means B time instances are randomly chosen from the BRIR at one spatial position. Also, a smaller learning rate should be used compared to large-size data according to the linear scaling rule. After careful parameter tuning, the network can correctly work in 3D.

The training process is conducted on a single NVIDIA A5000 GPU. For 1D and 2D datasets, $B = 4$, the learning rate is 10^{-4}. The network is trained for 2000 epochs, which takes a few minutes. For the 3D dataset, $B = 1000$, and the learning rate is 5×10^{-6}. The network is trained for 40000 epochs, which takes 4 hours.

Metrics Normalized mean squared error (NMSE) is used to measure the difference between the estimated BRIRs and the ground truth BRIRs when we are comparing the results of three datasets. This is defined by

$$\text{NMSE} = 10 \log_{10} \frac{1}{M} \sum_{m=1}^{M} \frac{||\hat{h}(:, \boldsymbol{r}_m) - h(:, \boldsymbol{r}_m)||^2}{||h(:, \boldsymbol{r}_m)||^2}. \quad (4)$$

By normalizing the BRIR error at one spatial position by the signal energy of the BRIR, different signal energies at different spatial positions will be considered so that NMSE will not be dominated by the high-energy signal. This is especially useful when the datasets for comparison have different energy scales. To better compare the models trained on three datasets, all of them are evaluated using the 1D dataset and NMSEs are calculated in one dimension. So here $M = 20$, which is the evaluation data size of the 1D dataset.

Three methods for BRIRs interpolation are compared: PINN, SIREN (only data loss is used, no PDE loss in the loss function) and bilinear interpolation.

Results

NMSEs of models trained with 1D, 2D and 3D datasets are shown in Table 1. For all three datasets, PINN performs better than SIREN and bilinear interpolation. This shows that PINN is able to model BRIRs correctly in all three dimensions. SIREN performs the worst for the 1D case, but performs better than bilinear interpolation in 2D and 3D. PINN performs much better than SIREN in 1D and 2D, but PINN and SIREN's performances become more similar in 3D. This demonstrates PINN's ability to do the interpolation with a smaller data size. However, PINN's performance still improves from 1D to 2D to 3D, because PINN struggles to learn the data distribution if the data size is too small. The truth that PINN can work in 1D and 2D shows the ability of PINN to learn the physical properties from the data even if the PDE equation is not explicitly given.

Table 1: NMSE (dB) of the three methods when 1D, 2D and 3D datasets are used.

	1D	2D	3D
Bilinear	1.89	1.89	1.89
SIREN	3.93	-0.18	-2.86
PINN	**0.04**	**-3.42**	**-4.20**

To better explain the experiment results, we visualize the first 400 time instances of the estimated BRIRs, which is 9 ms, in the $x - t$ domain, as shown in Figure 4. In each BRIR's visualization, the two lines represent the direct sound and the first reflection in the BRIR. By comparing PINN and SIREN in 1D and 2D, we can see the BRIRs estimated by PINN contain fewer artifacts than SIREN, which indicates the effectiveness of incorporating the PDE loss. This shows SIREN generates a large amount of noise when the data amount is small. For 1D PINN, the network is unable to learn the continuous change of the first reflection, which might be due to the small data size. By comparing the results of three datasets for both PINN and SIREN, we can see the artifacts in both methods become fewer, and SIREN is also able to learn noise-free estimation for the 3D case when the data amount is large enough.

Conclusion

In this work, we use PINNs to interpolate simulated BRIRs. The results show PINNs perform better than baseline methods SIREN and bilinear interpolation. This

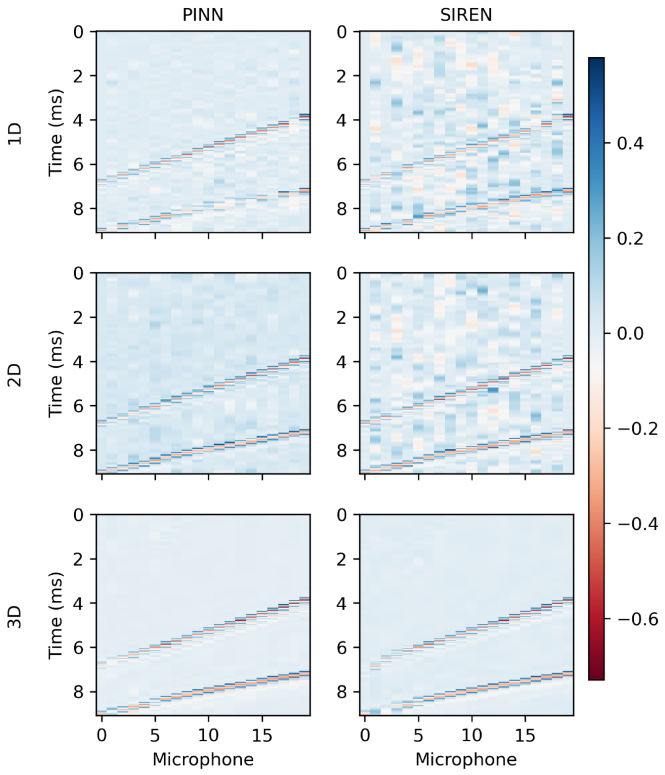

Figure 4: 2D $x-t$ plane visualization of the first 400 time instances of BRIRs estimated using SIREN and PINNs using 1D, 2D and 3D training datasets.

proves that PINNs can learn the head effects in BRIRs. By comparing the performances in 1D, 2D, and 3D, we find that PINNs can learn the reflections from all three dimensions even if the wave equation is only defined in 1D or 2D. These all prove that PINNs can learn the physical information from the training data even if it is not explicitly given in the PDE loss. Future work includes using a large dataset for training to learn data features across different rooms to make PINNs generalize to different acoustic environments.

Acknowledgement

Yazhou Li is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by the China Scholarship Council and Queen Mary University of London.

References

[1] V. Garcia-Gomez and J. J. Lopez. Binaural room impulse responses interpolation for multimedia real-time applications. In *Audio Engineering Society Convention 144*, Milan, 2018.

[2] Y. Zhang, Y. Wang, and Z. Duan. Hrtf field: Unifying measured hrtf magnitude representation with neural fields. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Rhodes Island, 2023.

[3] V. Sitzmann et al. Implicit neural representations with periodic activation functions. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Vancouver, 2020.

[4] K. Su, M. Chen, and E. Shlizerman. Inras: implicit neural representation for audio scenes. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, New Orleans, 2022.

[5] Andrew Luo et al. Learning neural acoustic fields. *Advances in Neural Information Processing Systems*, 35:3165–3177, 2022.

[6] Z. Ge, L. Li, and T. Qu. A hybrid time and time-frequency domain implicit neural representation for acoustic fields. In *Audio Engineering Society Convention 156*, Madrid, 2024.

[7] Diego Di Carlo et al. Neural steerer: novel steering vector synthesis with a causal neural field over frequency and direction. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New York, 2024.

[8] Y. Qiao and E. Choueiri. Neural modeling and interpolation of binaural room impulse responses with head tracking. In *Audio Engineering Society Convention 155*, New York, 2023.

[9] M. Pezzoli, F. Antonacci, and Sarti A. Implicit neural representation with physics-informed neural networks for the reconstruction of the early part of room impulse responses. In *Forum Acusticum*, Turin, 2023.

[10] X. Karakonstantis et al. Room impulse response reconstruction with physics-informed deep learning. *The Journal of the Acoustical Society of America*, 155(2):1048–1059, 2024.

[11] F. Ma et al. Physics informed neural network for head-related transfer function upsampling. *arXiv preprint arXiv:2307.14650*, 2023.

[12] A. M. Tartakovsky et al. Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research*, 56(5), 04 2020.

[13] Earl G Williams. *Fourier acoustics: sound radiation and nearfield acoustical holography*. Academic press, 1999.

[14] T. Wendt, S. Van De Par, and S. Ewert. A computationally-efficient and perceptually-plausible algorithm for binaural room impulse response simulation. *Journal of the Audio Engineering Society*, 62(11):748–766, 2014.

[15] V.R. Algazi et al. The cipic hrtf database. In *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, New Paltz, 2001.