forum acusticum 2026
8-12 September · Graz, Austria

REAL-TIME BROADBAND SOUND FIELD REGRESSION VIA MASSIVELY PARALLEL FREQUENCY STACKING NEURAL NETWORKS

Yuanxin Xia[1], Matteo Calafà[1], Cheol-Ho Jeong[1]

[1] *Acoustic Technology, Department of Electrical and Photonics Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark*

Abstract

Neural implicit fields are a powerful framework for sound field regression, but their application to broadband problems has been challenged by the prohibitively expensive computational cost of training neural networks. To break this barrier, this paper proposes a massively parallel strategy via frequency stacking. This architecture factorizes the broadband problem into independent, per-frequency branches that are trained in parallel, leveraging efficient GPU batching and parallel execution. Our results show that the proposed stacking architecture outperforms the conventional frequency conditioning architecture, with accelerations greater than two orders of magnitude and lower regression errors, which demonstrates a real-time potential for broadband sound field regression in small spaces. It is also demonstrated that incorporating a partial differential equation (PDE) residual loss further enhances the accuracy, particularly in the low to mid-frequency regime, despite at a cost of slightly longer training times. These findings offer practical guidelines for efficient, potentially real-time, yet high-fidelity sound field regression.

Keywords: *neural implicit field, sound field regression, frequency stacking, physics-informed neural networks*

1. Introduction

Regression of sound pressure from sparse measurements is central to virtual acoustics, spatial audio, and active noise control, yet remains difficult under realistic sampling budgets and/or challenging reverberation conditions [1], [2], [3], [4]. Data-driven estimators have shown progress on sound field reconstruction at low frequencies, but often require prohibitively large training datasets or fail to generalize across rooms and sensor layouts [5]. Physics-Informed Neural Networks (PINNs) have recently emerged as an appealing alternative by embedding the underlying physical equations into the loss, enabling volumetric reconstruction from sparse observations and supporting grid-free field queries [6], [7], [8], [9]. However, learning in the broadband frequency ranges is typically prohibitively slow with a single standard PINN, and convergence to the loss minimum is challenging.

Several strategies for multi-frequency learning have been explored in other domains. In geophysics, for example, frequency conditioning has been proposed, where a single network takes the frequency as a continuous input variable, sometimes enhanced with Fourier features or multiscale sinusoidal activations [10], [11]. Yet it remains unclear how well such conditioning strategies transfer to enclosed room acoustics, particularly at higher frequencies.

To address this challenge, we propose a frequency stacking strategy. Inspired by advances in operator learning and modern GPU capabilities [12], [13], [14], this architecture factorizes the broadband problem into independent, per-frequency branches. Each branch has dedicated weights, but all are fused into a single computational graph and executed concurrently on the GPU using optimized batched tensor operations. We present a direct comparison between the proposed stacking architecture and conventional frequency conditioning on a broadband room acoustic benchmark case. Specifically, we evaluate (i) regression error versus training time, (ii) convergence behavior across the spectrum, and (iii) memory and parameter efficiency.

2. Method

2.1. Problem formulation

This work addresses the problem of broadband sound field regression, namely, learning a continuous function that represents the complex sound pressure field

***Corresponding author**: yuxia@dtu.dk.*

$P(\mathbf{x}, k)$ from a sparse set of measurements. Here, $\mathbf{x} = (x, y, z)$ denotes the spatial coordinates and $k = 2\pi f/c$ is the wavenumber associated to the frequency f and speed of sound in air c. The sound field is governed by the Helmholtz equation

$$\nabla^2 P(\mathbf{x}, k) + k^2 P(\mathbf{x}, k) = 0. \quad (1)$$

We model the sound field as an implicit neural representation, where a neural network $\mathcal{N}_\theta$ with parameters θ maps input coordinates directly to the complex sound pressure. The neural network (NN) is optimized by minimizing a loss function that enforces fidelity to the measurement data. The data fidelity term, $\mathcal{L}_{\text{data}}$, ensures the NN's output converges to the known measured values at a sparse set of coordinates $\mathcal{D}_{\text{data}}$:

$$\mathcal{L}_{\text{data}} = \frac{1}{|\mathcal{D}_{\text{data}}|} \sum_{\mathbf{x} \in \mathcal{D}_{\text{data}}} \|\mathcal{N}_\theta(\mathbf{x}, k) - P_{\text{true}}(\mathbf{x}, k)\|^2. \quad (2)$$

The governing physical equation can also be incorporated as an additional residual loss term, $\mathcal{L}_{\text{res}}$, which serves as a learning regularizer. The residual of the Helmholtz equation is penalized over a set of collocation points, $\mathcal{D}_{\text{res}}$, which consist of coordinates where no measurement data is available and is defined by:

$$\mathcal{L}_{\text{res}} = \frac{1}{|\mathcal{D}_{\text{res}}|} \sum_{\mathbf{x} \in \mathcal{D}_{\text{res}}} \|\nabla^2 \mathcal{N}_\theta(\mathbf{x}, k) + k^2 \mathcal{N}_\theta(\mathbf{x}, k)\|^2. \quad (3)$$

The final training objective becomes a dynamically weighted sum

$$\mathcal{L}_{\text{total}} = \lambda_{\text{data}} \mathcal{L}_{\text{data}} + \lambda_{\text{res}} \mathcal{L}_{\text{res}},$$

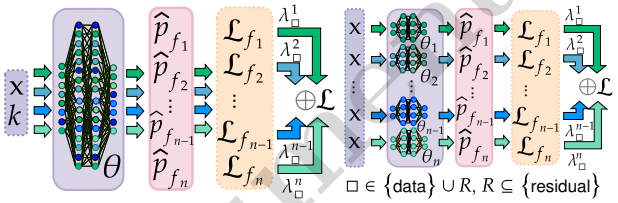
where $\lambda_{\text{data}}, \lambda_{\text{res}} > 0$ are scalar hyperparameters.

2.2. Network architecture for broadband regression

Two network architectures are compared: the conventional frequency conditioning approach and the proposed frequency stacking architecture, as shown in Figure 1. Both strategies are implemented as implicit neural representations using sinusoidal activation functions (SIRENs) with the modified MLP structure [13], [15].

The conventional method, referred to as frequency conditioning, treats frequency as a continuous input coordinate. As shown in Figure 1(a), this monolithic network takes as input $\{\mathbf{x}, k\}$ and learns a single, unified function over the joint space-frequency domain. While conceptually simple, this forces the network to learn a highly complex oscillatory behavior, creating a significant optimization barrier. In our experiments, we test this strategy with two configurations: 2 layers with 3072 neurons and 4 layers with 1792 neurons.

For the proposed frequency stacking method, a massively parallel architecture shown in Figure 1(b) is employed. This strategy factorizes the broadband problem into independent, per-frequency branches. It takes a single 3D spatial coordinate $\mathbf{x}$ and simultaneously outputs the complex pressure for all N frequency bins. This is achieved by implementing the parallel branches as a single fused GPU kernel, where a custom linear layer is optimized through the `opt_einsum` [14] module to efficiently execute batched matrix multiplications across the frequency dimension. This design is inherently GPU-friendly, mapping directly to the hardware's parallel nature and allowing all frequencies to be processed concurrently. For this architecture, we use in our experiments a compact configuration of two hidden layers with 128 neurons per layer. We designed two variants: one only with the data loss and the other combining the data and the PDE residual loss.



(a) Frequency conditioning (b) Frequency stacking

Figure 1: The two neural network architectures compared. (a) The conventional frequency conditioning approach uses a single monolithic network. (b) The proposed frequency stacking architecture factorizes the problem into parallel, per-frequency branches. The loss function must include data, whereas the PDE residual term is optional.

3. Experimental setup

3.1. Data acquisition

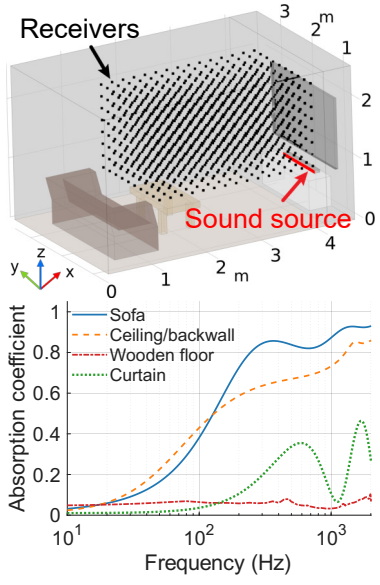


Figure 2: Room and acoustic boundary properties.

In Figure 2(a), we simulated a furnished rectangular room (4.5 × 3.0 × 2.6 m) containing a sofa, a table, and a TV cabinet using the COMSOL Pressure

Acoustics Module, with quadratic Lagrange elements and five elements per wavelength based on the highest study frequency [16]. The sound bar is used as a source, with its surface normal velocity defined as inversely proportional to the frequency, ensuring constant acoustic power output across the broadband spectrum. The sound field was sampled on a regular 3D grid of 15×10×8 points, corresponding to a total of 1200 receivers, with a uniform spacing of 0.2 m.

The acoustic properties of the surfaces are illustrated in Figure 2(b): the sofa was modeled as foam which has high absorption above 500 Hz, the side walls consist of several layers of curtains that exhibit resonance behaviors in their absorption, and the ceiling as a porous absorber. This configuration produced an overall reverberation time of about 0.5 s, which is common in actual living rooms. Accordingly, we used a frequency resolution of 2 Hz, with the lowest frequency of 70 Hz, which is the lower cutoff frequency of the 1/3 octave band centered at 80 Hz, and the highest frequency of 1120 Hz, i.e., the upper cut-off frequency of the 1/3 octave band centered at 1 kHz. The regression accuracy is analyzed in the 1/3 octave bands.

A sparse subset of 120 points, which corresponds to 10% of the receivers, was randomly selected from the training set, meaning that the remaining 1080 points are reserved as the validation set to evaluate the regression performances. During training, all complex pressure outputs were normalized at the receivers.

3.2. Training protocol

The training subset was generated three times with different random seeds to reduce sampling imbalance, while all networks shared the same random seed for reproducibility. All models were compiled and trained using the AdamW optimizer [17] with a maximum learning rate of 1×10^{-2}. The learning rate schedule included a 100 epoch warmup followed by cosine decay, both bounded at 0.1 of the maximum rate.

For both architectures, all frequencies have identical λ_{data} in the pure data-driven test. In the second scenario, where the residual loss is included, the gradient norm technique [18] is employed with a smoothing coefficient of 0.9 to balance the data and residual losses for each individual frequency. Here, the weighting is updated every 50 epochs, with the data loss weight λ_{data} initialized at 100 times λ_{res}. The residual loss was implemented using Jacobian–vector products and evaluated at all 1200 sampling point coordinates.

All models were trained for 250 epochs on a single Nvidia H100 GPU with PyTorch 2.8 and CUDA 12.8.

The two architectures were aligned in two key aspects to ensure a fair comparison. First, the number of network parameters is almost equivalent, as shown in Table 1. Second, the total number of coordinate-frequency pairs processed in a single training step is the same. We used the wave number instead of frequency because its scale is comparable to that of the coordinates.

3.3. Evaluation metrics

To evaluate the performance of the models, we define metrics for the physical accuracy and computational efficiency. The primary measure for accuracy is the SPL difference in the 1/3-octave band at the center frequency f_c, namely,

$$e_{f_c} := \left| SPL_{\text{pred,oct}}\left(f_c\right) - SPL_{\text{true,oct}}\left(f_c\right) \right| . \quad (4)$$

This metric evaluates the difference in the energy summed within the 1/3 octave bands and emphasizes the difference near the peaks in the frequency response functions, i.e., the energy near the natural frequencies. An average is performed across the receiver location and the frequency and named as $\bar{e}_{f_c}$, which defines a single-number global accuracy measure. The computational efficiency is quantified by three metrics recorded during training: (1) the total wall-clock training time excluding the compilation (s), (2) the average GPU utilization, i.e., Streaming Multiprocessor utilization percentage [19], SM (%), and (3) the average GPU memory usage (GB).

4. Results and analysis

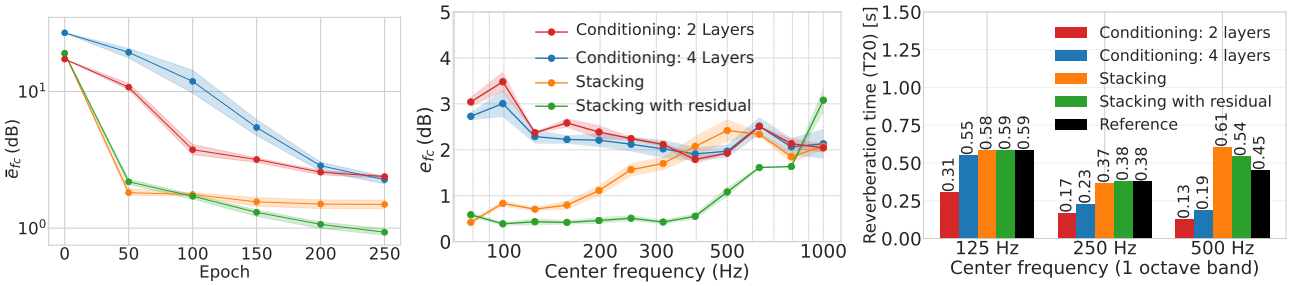
A summary of the accuracy and efficiency metrics for the four network configurations tested is presented in Table 1. Results show a clear out-performance by the stacking strategy. The stacking architectures is over two orders of magnitude faster than the conditioning ones. In particular, the pure data-driven regression takes less than 3 seconds, compared to about 1475 seconds for the conditioning models. The stacking model with residual loss requires about 60 seconds due to frequent calls to automatic differentiation. The stacking models reach much higher SM utilization, up to 85.30% with residual loss, while the conditioning model reaches only about 1%. In terms of accuracy, the stacking model with residual loss achieves the lowest $\bar{e}_{f_c}$ of 0.9 dB, outperforming all other configurations.

Table 1: Performance and efficiency with the best model in bold.

	Frequency conditioning		**Frequency stacking**	
Loss	Data	Data	Data	Data+residual
Architec	[4, 3072x2, 2]	[4, 1792x4, 2]	526x[3,128x2,2]	526x[3,128x2,2]
N_{para}	9.49 M	9.67 M	9.62 M	9.62 M
SM (%)	1.0	1.1	20.0	**85.3**
Mem.(GB)	48.7	48.7	**21.6**	36.9
Time (s)	1475.4	1471.5	**2.6**	61
$\bar{e}_{f_c}$ (dB)	2.4	2.3	1.5	**0.9**

The training dynamics are shown in the convergence plot in Figure 3(a). Consistent with the results in the Table 1, the stacking models converge faster, reaching a lower error within about 50 epochs. The pure data-driven model shows only minor improvement with further training up to 250 epochs, whereas the model with the residual loss continues to improve. In contrast, the conditioning models converge at a significantly reduced rate, confirming their lower parameter efficiency for this task.

The error e_{f_c} at the final epoch is shown in Fig-



(a) Convergence of e_{f_c} that evaluate at the validation points.

(b) e_{f_c} of the final epoch.

(c) Comparison of reverberation time.

Figure 3: Comparison of models' convergence and evaluation error.

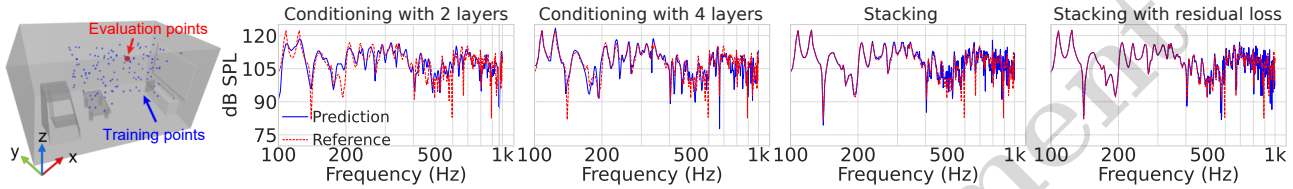


Figure 4: Prediction vs. reference frequency response function at receiver [2.7, 1.7, 2] m.

ure 3(b). The stacking network with residual loss achieves the lowest error across most of the spectrum, although its effectiveness is strongly frequency-dependent. Compared to the pure data-driven stacking network, adding the residual loss substantially reduces the error below 1 kHz, where longer wavelengths make the residual loss an effective regularizer. Above 1 kHz, however, it leads to a larger error, as the fixed collocation spacing in $\mathcal{D}_{\text{res}}$ cannot resolve shorter wavelengths and the additional constraint makes the loss harder to balance. Among the conditioning models, deeper networks yield slightly smaller errors. Furthermore, both conditioning models reach lower error at the 1 kHz 1/3-octave band than the stacking network with residual loss. In Figure 3(c), the reverberation time is compared with the ground truth at a receiver point at [2.7, 1.7, 2] m, visualized in the first figure of Figure 4. As expected from Figure 3(b), the predicted reverberation times, T20, are very accurate for the stacking network up to the 250 Hz band. Above 500 Hz, the discrepancies in T20 are larger than one just noticeable difference (JND) of 5 % for reverberation time [20]. Although not shown, the predicted T20 converges within one JND from the reference after a longer training, approximately 1000 epochs, allowing users to choose between accuracy and computational speed depending on the purpose of the project. Finally, the frequency response function at the receiver used in Figure 3(c) is compared in Figure 4. The stacking network with the residual loss agrees best with the reference, particularly in capturing the modal behavior at low frequencies. However, since stacking models treat each frequency independently, they cannot capture inter-frequency continuity and can produce non-physical spectral fluctuations at higher frequencies. In contrast, conditioning networks enforce smooth evolution across frequencies by learning a unified function over the joint space–frequency domain, but they fail to extrapolate from sparser training data points and struggle to capture the full spectral details.

5. Discussion

The proposed frequency stacking strategy has proven to be more effective and accurate in sound field regression tasks than the frequency conditioning. The frequency stacking architecture offers a speed-up of over 500 times, with the pure data-driven variant compared to the frequency conditioning architectures. The training time of 2.6 s enables a nearly real-time sound field regression for the room and frequency range tested. Thanks to the impressive convergence rate shown in Figure 3(a), a reasonably accurate and perceptually plausible regression is possible by training only up to 50 epochs, corresponding to a 520 ms training time. Small rooms, such as car cabins, are potential applications for real-time sound field regression.

In terms of accuracy, incorporating the residual loss into the stacking networks yields the impressively low error of 0.9 dB, which is within one JND for broadband sound perception [21], highlighting the potential of PDE constraints as a strong regularizer. This advantage, however, comes with a trade-off: stacking models treat frequencies independently and may produce non-physical spectral fluctuations, whereas conditioning networks enforce smooth inter-frequency evolution at the cost of lower efficiency and accuracy.

6. Conclusion

We presented two neural network architectures that can efficiently perform spatial regression of sound fields for a broad frequency band. Specifically, we

compared frequency conditioning and frequency stacking strategies. We found that the frequency stacking strategy outperforms the conditioning network both in terms of accuracy and efficiency. The pure data-driven frequency stacking networks demonstrate near real-time potential having a reasonable accuracy. The physics-informed variant provides the best accuracy, particularly thanks to its exceeding performance at the low frequencies. These findings point to a fundamental trade-off between the speed of stacking and the continuity of conditioning, motivating future work on hybrid designs that combine their corresponding strengths.

References

[1] F. Ma, S. Zhao, and I. S. Burnett, "Sound field reconstruction using a compact acoustics-informed neural network," *The Journal of the Acoustical Society of America*, vol. 156, no. 3, pp. 2009–2021, 2024.

[2] D. Mylonas, A. Erspamer, C. Yiakopoulos, and I. Antoniadis, "A virtual sensing active noise control system based on a functional link neural network for an aircraft seat headrest," *Journal of Vibration Engineering & Technologies*, vol. 12, no. 3, pp. 3857–3872, 2024.

[3] Y. A. Zhang, F. Ma, T. D. Abhayapala, P. N. Samarasinghe, and A. Bastine, "An active noise control system based on soundfield interpolation using a physics-informed neural network," in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, pp. 506–510.

[4] S. Xu, J. A. Zhang, T. D. Abhayapala, A. Bastine, W.-T. Lai, and P. N. Samarasinghe, "Sparse sound field representation using complex orthogonal matching pursuit," in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, pp. 1336–1340.

[5] F. Lluís, P. Martínez-Nuevo, M. B. Møller, and S. E. Shepstone, "Sound field reconstruction in rooms: Inpainting meets super-resolution," *The Journal of the Acoustical Society of America*, vol. 148, no. 2, pp. 649–659, 2020.

[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[7] X. Karakonstantis, D. Caviedes-Nozal, A. Richard, and E. Fernandez-Grande, "Room impulse response reconstruction with physics-informed deep learning," *The Journal of the Acoustical Society of America*, vol. 155, no. 2, pp. 1048–1059, 2024. DOI: 10.1121/10.0024750.

[8] M. Olivieri, X. Karakonstantis, M. Pezzoli, F. Antonacci, A. Sarti, and E. Fernandez-Grande, "Physics-informed neural network for volumetric sound field reconstruction of speech signals," *EURASIP Journal on Audio, Speech, and Music Processing*, 2024. DOI: 10.1186/s13636-024-00366-2.

[9] F. Ma, S. Zhao, and I. S. Burnett, "Sound field reconstruction using a compact acoustics-informed neural network," *The Journal of the Acoustical Society of America*, vol. 156, no. 3, pp. 2009–2020, 2024. DOI: 10.1121/10.0029022.

[10] C. Song and Y. Wang, "Simulating seismic multifrequency wavefields with the fourier feature physics-informed neural network," *Geophysical Journal International*, vol. 232, no. 3, pp. 1503–1514, 2023.

[11] X. Chai, Z. Li, T. Alkhalifah, and Y. Wang, "Modeling multisource multifrequency acoustic wavefields by a multiscale fourier feature physics-informed neural network," *Geophysics*, vol. 89, no. 3, T79–T94, 2024.

[12] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, 2021.

[13] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, A3055–A3081, 2021.

[14] D. G. Smith and J. Gray, *Opt_einsum-a python package for optimizing contraction order for einsum-like expressions. j. open source softw. 3, 26 (2018), 753*, 2018.

[15] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.

[16] COMSOL, *Eigenmodes of a room*, COMSOL Application Gallery, Application ID: 63, 2025. Accessed: Sep. 17, 2025. [Online]. Available: https://www.comsol.com/model/eigenmodes-of-a-room-63.

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[18] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, "An expert's guide to training physics-informed neural networks," *arXiv preprint arXiv:2308.08468*, 2023.

[19] NVIDIA Corporation, *Nvidia h100 tensor core gpu architecture*, `https://resources.nvidia.com/en-us-hopper-architecture/nvidia-h100-tensor-c`, Accessed: 2025-09-17, 2022.

[20] ISO, "Iso 3382-1:2009 acoustics – measurement of room acoustic parameters – part 1: Performance spaces," ISO, Geneva, Switzerland, Standard, 2009.

[21] G. A. Miller, "Sensitivity to changes in the intensity of white noise and its relation to masking and loudness," *The Journal of the Acoustical Society of America*, vol. 19, no. 4, pp. 609–619, 1947.