Audio Engineering Society

Conference Paper 16

Presented at the AES International Conference on Machine Learning and Artificial Intelligence for Audio
2025 September 8–10, London, UK

This paper was peer-reviewed as a complete manuscript for presentation at this conference. This paper is available in the AES E-Library (http://www.aes.org/e-lib), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Simulating 3D Acoustic Radiation and Scattering in the Frequency Domain with Fourier Neural Operators (FNOs)

James Hipperson[1,2], Trevor Cox[1], and Jonathan Hargreaves[1]

[1] *University of Salford*
[2] *Funktion-One Research Ltd.*

Correspondence should be addressed to James Hipperson (j.r.hipperson@edu.salford.ac.uk)

ABSTRACT

Acoustic wave behaviour is described by the acoustic wave equation, which is a partial differential equation (PDE). PDEs are difficult to solve except for in the simplest cases, which necessitate the use of simplified models (such as geometrical approaches) or numerical methods (such as Finite Element) for real-world applications like room acoustics and loudspeaker radiation. Neural Operators are an emerging class of deep learning architectures that learn to approximate a solution to a PDE. Fourier Neural Operators (FNOs) project input functions onto Fourier modes, and parametrise the kernel operator in Fourier space. However, a large amount of data is still required, and the training process can be challenging. This paper presents a proof-of-concept using a U-shaped Fourier Neural Operator to solve the 3D Helmholtz Equation in air with pairs of simplified loudspeakers. The network is able to learn wave behaviour with varying accuracy across a range of frequencies, ranging from a Mean Absolute Error (MAE) of 0.12 at 500 Hz to 0.28 at 4 kHz. Once trained, inference is 15 to 3000 times faster than the Boundary Element Method (BEM). Although still at an early stage of development, Neural Operators show significant potential for a wide range of simulation applications.

1 Introduction

Most physical phenomena in the universe can be described by partial differential equations (PDEs). Unfortunately, most PDEs are difficult to solve. For acoustics, the PDE is the wave equation, and its time harmonic / frequency domain equivalent, the Helmholtz equation. In audio, this can be used to model scenarios such as loudspeaker radiation and room acoustics. The conventional methods for solving the Helmholtz equation for non-trivial cases are numerical, such as the Finite Element Method (FEM), Boundary Element Method (BEM) and Finite Difference Time Domain (FDTD) [1]. These are well established, but have two significant disadvantages. Any change to the model, no matter how small, typically requires the model to be solved again from scratch, and they become computationally intractable with very large numbers of elements / degrees of freedom.

Neural Operators are an emerging class of deep learning methods that seek to address these two problems. Traditional deep learning models attempt to learn a map between two finite-dimensional sets of data. Physics-informed neural networks (PINNs) learn to solve a single PDE instance by having a PDE residual in the

Table 1: Comparison of machine learning paradigms

Name	Training	Generalisation	Discretization
Convolutional Neural Network (CNN)	Data only	Image data	Fixed regular grid
Physics-Informed Neural Network (PINN)	Data and/or Physics	Single PDE instance	Meshless
DeepONet	Data only	General PDE class	Meshless
Fourier Neural Operator (FNO)	Data only	General PDE class	Any regular grid
Physics-Informed Neural Operator (PINO)	Data and/or Physics	General PDE class	Any regular grid

loss function (physics loss). Neural operators learn an approximation of the solution operator for a PDE class; a comparison of common paradigms is given in Table 1. The motivation for trying to learn the physics, rather than just the outcomes, is that in acoustics, the free-space solutions to the PDEs are well known and understood, so can be used as building blocks for other problems. A more significant problem is that without physical constraints, data-trained models can produce physically incorrect predictions.

Deep Learning has been applied to many problems in audio signal processing, but there are fewer applications to solving the wave equation directly. Most applications are from Geophysics, where the heterogenous wave equation and inverse problems introduce additional complexity that make neural networks an especially attractive solution compared to other numerical methods.

Some examples of airborne acoustics in physics-informed machine learning: A novel method for Nearfield Acoustic Holography (NAH) [2]. Using a CNN combined with a linear layer performing the Kirchoff-Helmholtz propagation, in effect, introducing a physics constraint via uniqueness (the output of the neural network must map to the correct result via the KH layer). A data-free PINN (only physics loss) for solving the acoustic wave equation [3]. A PINN to solve the wave equation in 3D in a simple cuboidal volume [4]. Neural operator methods applied to acoustics are not common, but [5] is an example of applying DeepONet to predict sound fields in complex 3D rooms with parametric source positions.

In this paper, we will explore the use of Fourier Neural Operators to predict the sound field near simplified loudspeakers (reflecting cubes with one radiating face) in free field.

1.1 Neural Operators

Neural Operators are motivated by the idea that if neural networks can be universal function approximators, they could also work as universal operator approximators. An early example being [6] with DeepONet in 2019.

1.2 Fourier Neural Operators (FNO)

Neural Operators were introduced in [8, 9] in 2020-21, utilising the principles that convolution is a special type of integral operator, and that spatial convolution is multiplication in the Fourier domain. This can be substantially accelerated by the Fast Fourier Transform (FFT). [7] introduced the Fourier Neural Operator (FNO) in 2021. Eq. 1 summarises the approach. The result of the integral operator K acting on function v_t at point x is approximated by the inverse Fourier transform F^{-1} of R_ϕ (parameterised by the network) multiplied by the Fourier transform F of v_t (convolution in the spatial domain) for every point x in domain D. For a complete explanation of the approach, refer to [7].

$$\big(K(\phi)v_t\big)(x) = F^{-1}\Big(R_\phi \cdot F(v_t)\Big)(x) \qquad \forall x \in D \quad (1)$$

Fig. 1b shows the general structure of the Fourier layer, and Fig. 1a shows how many layers may be connected to form a deep network. The first fully connected layer P projects the input data to a higher-dimensional latent space. One or more Fourier layers F transform the input into the Fourier domain and truncate the higher modes before performing the inverse Fourier transform F^{-1}. A skip connection has a linear transformation/bias W and σ is a non-linear activation function. A final fully connected layer Q projects back to

the target dimension. Truncating high-order modes has similarities to image compression, exploiting the fact that in most physical systems, higher-frequency modes contain relatively little information about the system. This improves training and generalisation, at the cost of some small details. Parallels can be seen to the modal expansion approach in [10].

1.3 U-Shaped Neural Operators (U-NO)

The U-NO is inspired by similar structures in Convolutional Neural Networks (CNNs), Autoencoders and U-Nets for image-to-image applications. This approach was necessitated by the structure of images - even a 64x64 image has 4096 pixels. A deep, fully connected network operating on such high-dimensional data would have extreme memory requirements and be very difficult to train. CNNs dramatically reduced the number of parameters while still providing good performance, making them a viable solution for computer vision. [12] demonstrated the now-classic convolutional neural network in 1998 which famously outperformed previous computer vision techniques by using convolutional layers and hierarchical feature extraction. [13] introduced the U-Net structure in 2015 for medical imaging applications, where the dimensionality is reduced, then expanded again, with skip connections. [14] introduced the U-NO in 2023. Just like in [13], the contracting and expanding structure means that the dimensionality and number of parameters are reduced. This enables models of systems that would be difficult or impossible to solve using numerical methods (fluid

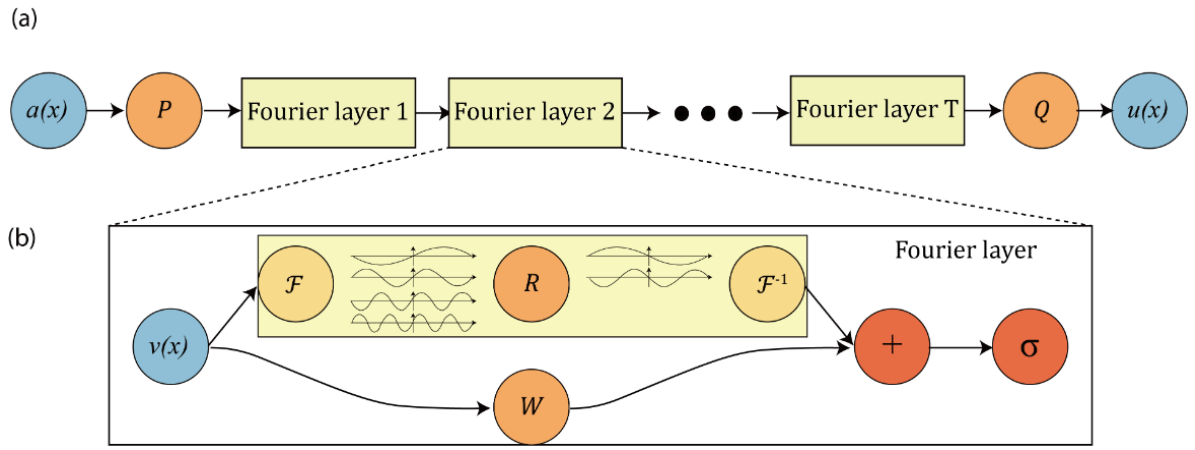


Fig. 1: (a) The use of multiple Fourier layers to map from a to u in a deep network (b) One Fourier Layer. From [7]

.

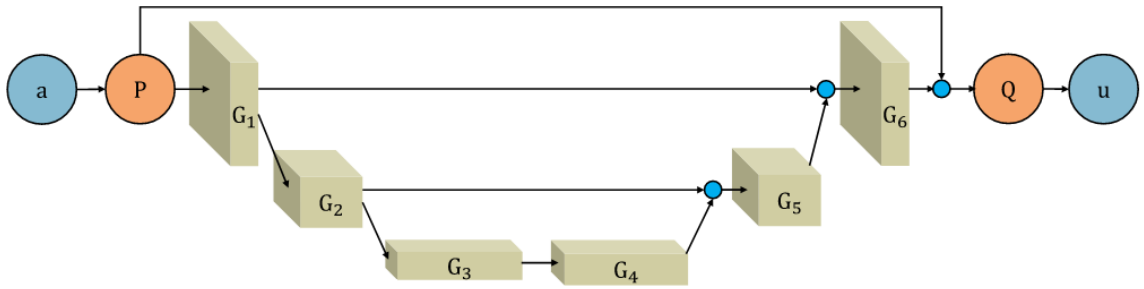


Fig. 2: U-NO structure (adapted from [11]).

dynamics, waveform inversion in geophysics) and, as demonstrated in this paper, enables significantly accelerated solution of acoustic models, opening up new applications and possibilities.

[11] demonstrates the effectiveness of a U-NO performing the challenging Geophysics task of seismic waveform inversion. Using a known source position and several frequencies, the network predicts the pressure field so that the velocity in the propagation medium can be estimated. Discontinuities in the velocity of the medium cause refraction, reflection and scattering. Correspondingly in airborne acoustics, reflection and scattering are frequently encountered in rooms and radiation from loudspeakers. Refraction occurs in outdoor sound propagation, where there are atmospheric temperature gradients. Fig. 2 shows the U-shaped network structure, with skip connections and contracting and expanding Fourier layer dimensions (G1-6). P and Q are fully connected layers as discussed in Section 1.2, and in practice are typically pairs of fully connected layers (Table 2).

2 Methods

The basic U-NO network from [11] was tested on a problem consisting of randomly generated pairs of solid reflecting cubes in free field. Each cube has one randomly chosen face radiating. Even with such a simple setup, the resulting sound pressure field can be very complex (See Fig. 3c)). The cubes were randomly positioned within the grid (without intersecting) and ranged from 2-20 cm in size. Fig. 3 shows three of the four network inputs. a) The 2D radiating faces within the 3D grid, represented by +1 or -1 depending on the orientation. b) Shows the volumetric binary mask of the cubes, represented by 1s in an empty grid of zeros. c) Shows the real component of the output from the Boundary Element Method (BEM) calculation, with the cube volumes masked off to zero pressure to remove any artefacts from the BEM calculation. The fourth input (not shown) is the frequency value scaled to a logarithmic spacing between 0 (0.5 kHz) and 1 (4 kHz). BEM is naturally suited to the problem of radiation in an infinite domain as only the boundaries must be meshed. Numerical methods are a convenient method for generating acoustic data for training. Disadvantages are computation time and that the data is perfectly clean, e.g. no noise, variation in acoustic parameters, temperature, measurement positions, etc. Neural networks operating on real measurement data would have all of these additional factors to contend with.

The network was implemented with Pytorch 2.6 [15]. The input data was generated programmatically with numpy and the output data (complex-valued pressure) was generated with BEM using bempp-cl ([16]). The mesh size was varied with frequency to use a (relatively coarse) minimum of 5 elements per wavelength. Standardisation was performed by scikit-learn [17]. In all cases, the predicted field was a 64x64x64 3D grid with real and imaginary components of sound pressure. Fig. 2 shows the U-shaped network structure.

An original dataset of 1000 pairs of cubes at 4 frequencies was augmented by spatially mirroring/flipping the data to expand the dataset to 2000 batches (8000 individual samples). This was split into a training set of 1600 batches, a validation set of 200 and a test set (unseen in training) of 200. The test set contained the same frequencies, but unseen configurations of size, position and radiating faces. The generalisability to

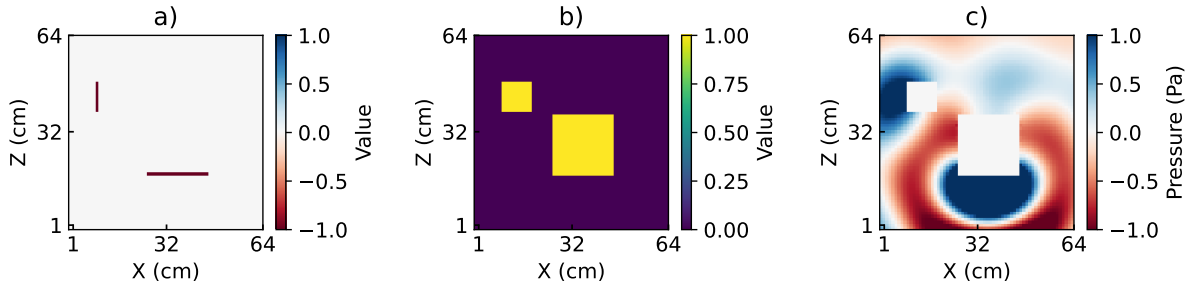


Fig. 3: Network inputs: a) Radiating Faces b) Binary Mask c) BEM Output (Frequency value channel not shown).

Table 2: Model Summary (width=8)

Layer	Width	Modes
Fully Connected 1	in,4	
Fully Connected 2	4,8	
Fourier 1	8,16	48,24
Fourier 2	16,32	32,16
Fourier 3	32,64	16,8
Fourier 4	64,32	32,8
Fourier 5	64,16	48,16
Fourier 6	32,16	64,24
Fully Connected 3	24,32	
Fully Connected 4	32,out	

unseen frequencies was not tested in this work, but is an important test for future work. The grid resolution was 1 cm per point, with octave band centre frequencies from 1 kHz to 4 kHz. The batch size was 4 (4 frequencies, 2 configurations = 8 total samples).

Width in the context of FNO has the same meaning as in a CNN: the base channel dimension, which is typically expanded as spatial dimensions are reduced, and vice versa (see Fig.2 and Table 2).

The learning rate was initially 0.001 and decayed by 0.5 every 50 epochs. The model was run for 150 epochs. Early stopping was implemented by saving the model state with the lowest validation loss. Training of the width=8 model took 14 hours on a single NVidia GTX 1080Ti (11 GB of memory).

The model structure is presented in Table 2. Fourier modes were not truncated to ensure accurate prediction of high-frequency wave effects, but this may not have been essential. Training of the width=16 model took 7 hours on a single NVidia A100 (40 GB of memory). The model structure was the same as in Table 2 but with the width of each layer doubled. The modes remained the same. The initial learning rate was reduced to 0.0005, decaying by 0.7 every 40 epochs. The loss function for both models was $0.9 \cdot MAE + 0.1 \cdot MSE$. This proportion was chosen empirically to encourage the network to focus on a generally correct solution, rather than getting stuck on outliers. Further improvement to the loss function may be possible.

$$MAE = \frac{\sum_{i=1}^{N} |y_{predicted} - y_{true}|}{N} \tag{2}$$

$$MSE = \frac{\sum_{i=1}^{N} (y_{predicted} - y_{true})^2}{N} \tag{3}$$

3 Results

3.1 Training

Fig. 4 shows the contrast between training on a single frequency vs. several frequencies. Convergence is much slower in the multiple frequency case (b), and validation loss stalls at around 0.33. In contrast, the single frequency case (a) is able to reach a validation loss of 0.2. This suggests that further optimisations to the training setup could be made. Increasing the layer width to 16 (c) did not significantly improve the performance which supports this.

3.2 Error vs. frequency and layer width

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_{predicted} - y_{true})^2}{N}} \tag{4}$$

Table 3 shows the Mean Absolute Error (MAE) (Eq. 2), Standard Deviation (SD) and Root Mean Squared Error (RMSE) (Eq. 4) vs. frequency, over the complete test set (200 samples of four frequencies each). The results clearly show the effects of spectral bias: the network is much more effective at learning lower-frequency solutions. The standard deviation shows that the variation in performance increases with frequency.

3.3 2D slices of 3D pressure fields

XZ plane slices at Y=32 cm of real and imaginary parts in Pascals are plotted in Figs. 5,6,7. Difference plotted as $P_{Predicted} - P_{True}$. Differences are highest at the reflecting boundaries. Both the pressure value and gradient are highest here. Fig. 5 and Fig. 6 show similar behaviour in that the general structure is correct. MAE throughout the domain can be less than 0.3, but the maximum absolute error can exceed 3.0 at the boundary or at very small singularities. Fig. 7 on the other hand demonstrates the difficulty of predicting the correct details as wavelength decreases and complexity of the pressure distribution increases. Different configurations are shown for each frequency to demonstrate the diversity of wavefields.

Table 3: Error over test set

	Width: 8			Width: 16		
Frequency	MAE	SD	RMSE	MAE	SD	RMSE
0.5 kHz	0.135	0.217	0.256	0.120	0.180	0.216
1 kHz	0.155	0.251	0.295	0.134	0.193	0.235
2 kHz	0.291	1.240	1.274	0.292	1.244	1.277
4 kHz	0.269	1.160	1.191	0.281	1.591	1.616

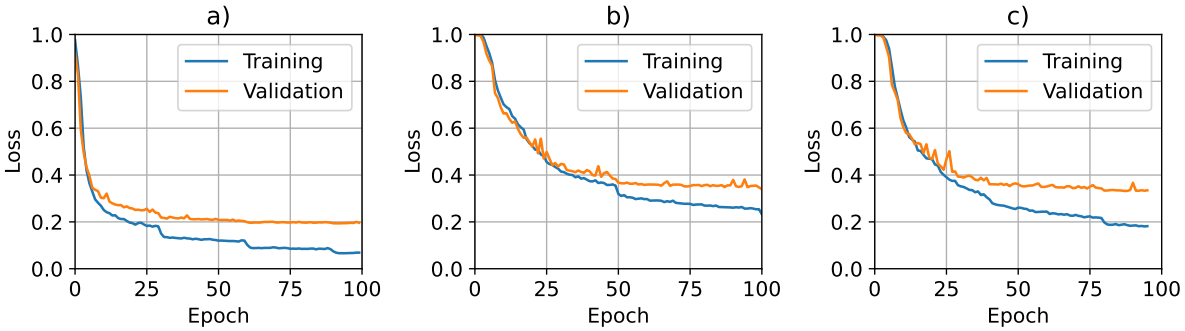


Fig. 4: Training curves: a) Width 8, trained on a single frequency (1 kHz) b) Width 8, trained on four frequencies (0.5 kHz, 1 kHz, 2 kHz, 4 kHz) c) Width 16, trained on four frequencies (0.5 kHz, 1 kHz, 2 kHz, 4 kHz).

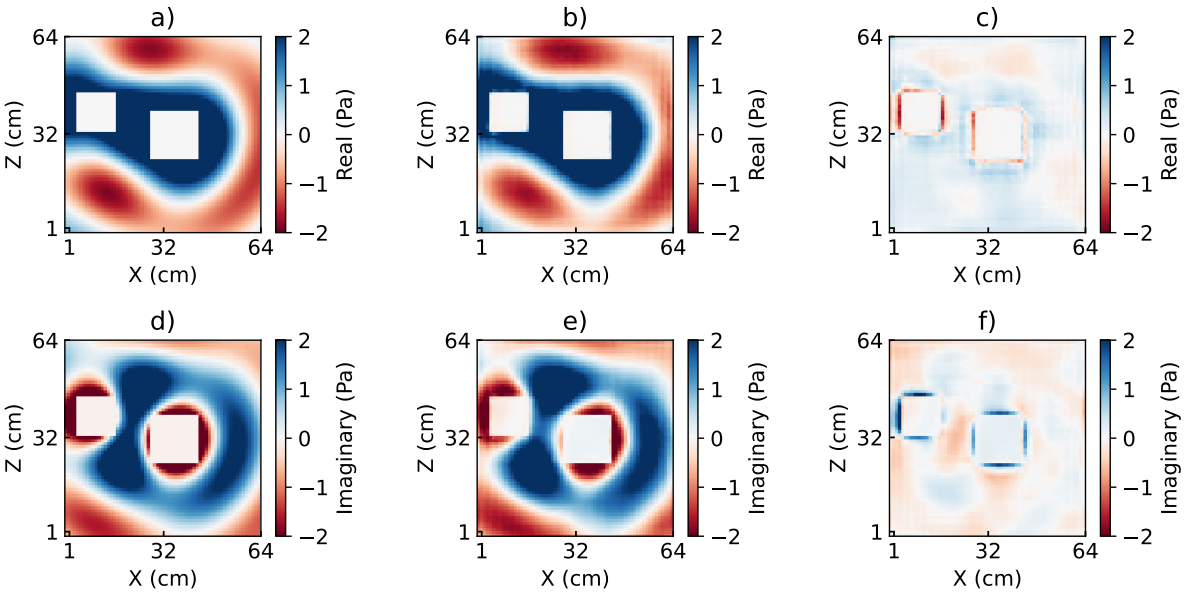


Fig. 5: Acoustic pressure at 1 kHz, Width=8 model: a) Ground Truth (real part) b) Prediction (real part) c) Difference (real part). d) Ground Truth (imaginary part) e) Prediction (imaginary part) f) Difference (imaginary part)

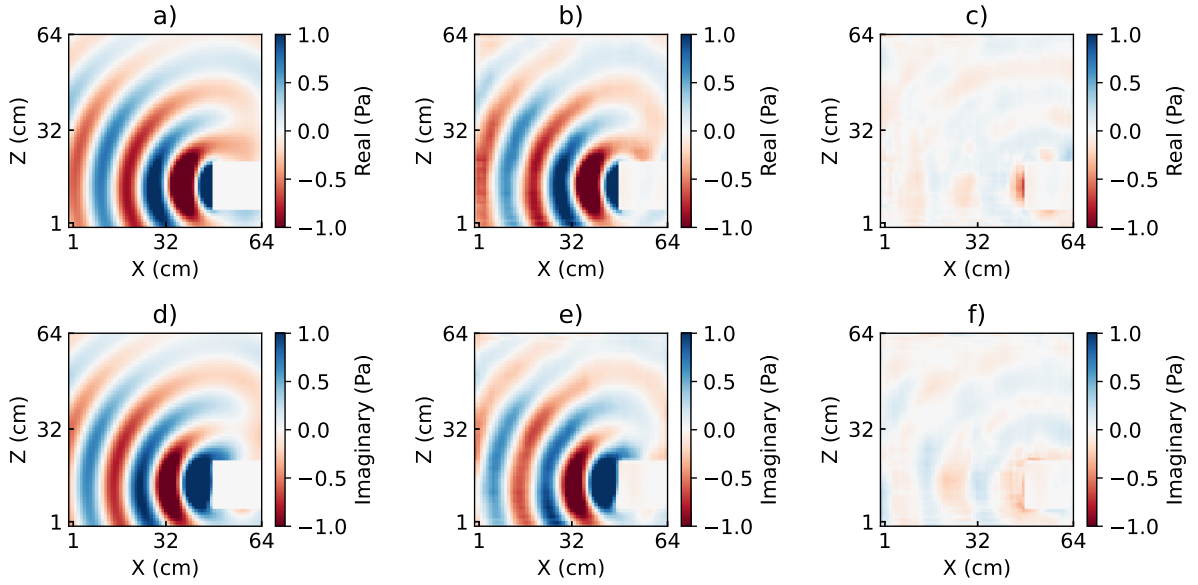


Fig. 6: Acoustic pressure at 2 kHz, Width=8 model: a) Ground Truth (real part) b) Prediction (real part) c) Difference (real part). d) Ground Truth (imaginary part) e) Prediction (imaginary part) f) Difference (imaginary part)

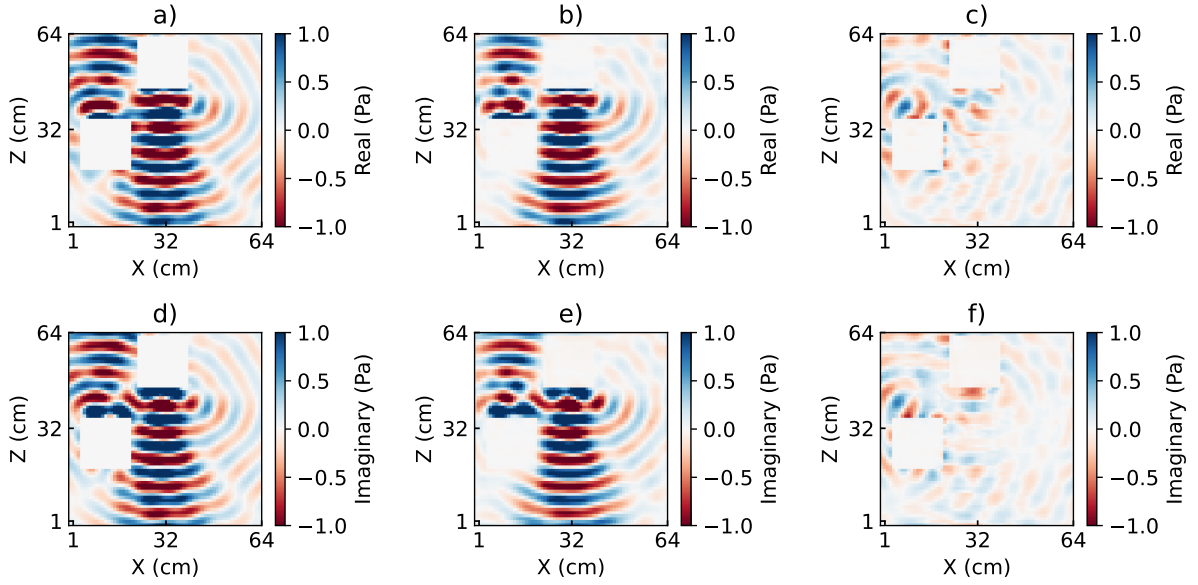


Fig. 7: Acoustic pressure at 4 kHz, Width=8 model: a) Ground Truth (real part) b) Prediction (real part) c) Difference (real part). d) Ground Truth (imaginary part) e) Prediction (imaginary part) f) Difference (imaginary part)

3.4 Inference time

Each prediction from the width 8 model took 0.05 seconds for all four frequencies. In comparison, the BEM calculations ranged from 1.5 seconds (for small meshes/high frequencies) to over 150 seconds (for large meshes/low frequencies). This equates to a speed-up range of 30-3000 times. Each prediction from the width 16 model took 0.1 seconds, a minimum speed-up of 15 times. Inference for both models was performed on a GTX 1080Ti GPU. BEM calculations were performed on a CPU (AMD Ryzen 9 3900X). Some preliminary tests were performed running BEM calculations on a GPU in the interest of experimental fairness, but performance was typically worse than on a CPU. Two reasons for this are the main memory to GPU memory transfer bottleneck, and that solution of the dense global matrix equation is not easily parallelisable. [18]

4 Discussion

The results show that FNOs can learn acoustic wave behaviour in 3D across a range of frequencies, but with diminishing accuracy at higher frequencies. However, the error is not general across all samples, but concentrated in a small number of samples where the network fails significantly. GPU memory limited the layer width and batch size substantially, which limited the accuracy of the results and likely contributed to the training difficulties. However, the results show that inference on conventional (consumer) computing hardware is possible, and useful results with improved accuracy will likely become possible in the near future, as the available GPU memory in devices increases.

The most challenging aspect was hyperparameter tuning. The model readily learned single-frequency solutions, but in the multi-frequency case, the model failed to converge in most configurations. A full ablation study would be useful to understand whether the model can be further simplified for airborne homogenous acoustics problems. The training curves in Fig. 4 show the hyperparameter optimisation difficulty: Overfitting, which indicates excessive model complexity, but poor performance when learning multiple frequencies. The layer width was significantly reduced from the Geophysics model (from 32 to 8) but all six layers and all Fourier modes were retained, which may not have been necessary. Larger layer width but fewer layers or Fourier modes may produce better results more efficiently.

A significant limitation of FNOs is the requirement for a regular grid. [19] explores a method for deforming a geometry mesh into a regular grid for use with FNO layers, then transforming back to the original mesh geometry. Another limitation is that the network in this paper (and FNOs generally) only have indirect physics constraints in the form of inductive bias (trained on the output from a numerical method) and the parameterisation of the kernel operator in Fourier space. There may be better network architectures that embed the constraints of the wave equation, leading to easier training and more accurate results. It is remarked in [20] that compared to data-based machine learning (which has existed for longer), the network architectures in operator learning are still primitive.

In terms of further work, comparison of operator learning models with a conventional image-to-image CNN baseline for the same task would be valuable. Interpretability is also a key area of interest. Currently it is unclear exactly what FNOs learn, and whether the latent space has any similarity to known physics. One way of testing this would be to make predictions for problems with well-understood analytical solutions, and inspect the intermediate data passing between layers in the network.

A residual learning approach could have benefits. Rather than having the network learn and predict the full solution, it would learn and predict the difference between an approximate solution (point source model) and an accurate solution (BEM). This could train faster and produce more accurate results.

Another important direction for further work is how to efficiently and usefully represent complex and arbitrary geometry in operator learning. Additionally, architectural physics constraints should be explored to determine whether this can improve the ease of training and the accuracy of results. The dataset of 8000 samples was around 20 GB. Optimal datasets for more complex problems may be several times larger, becoming increasingly time-consuming to generate and train on. Making training more efficient with physics constraints is therefore a key area for further work.

5 Summary

FNO was applied to an acoustic radiation and scattering problem with simplified loudspeaker geometry, at medium to high frequencies. FNOs are capable of approximating a solution operator for the 3D wave equation. The accuracy of results is reduced at high frequencies, but inference is 15-3000x faster than BEM, as a forward pass through the network is a constant computational complexity (depending only on the model size) whereas numerical methods scale by $O(N)$ (FEM) to $O(N^2)$ (BEM), with higher frequencies (or larger domains) requiring more degrees of freedom and therefore more computation.

FNOs can be thought of as a generalisation of image-to-image CNNs, but because they are general, the network must learn the underlying physics from scratch. However, as mentioned in the introduction, the physics of acoustic waves is well known already. Therefore, the most important and obvious direction for further work is how to incorporate domain knowledge and known physics constraints to build operator learning networks that are more specific to acoustic problems.

6 Acknowledgements

This work was supported by Funktion-One Research Ltd. as part of the CDT in Sustainable Sound Futures.

References

[1] Sakuma, T., Sakamoto, S., and Otsuru, T., editors, *Computational Simulation in Architectural and Environmental Acoustics: Methods and Applications of Wave-Based Computation*, Springer, Tokyo, 1 edition, 2014, ISBN 978-4-431-54453-1, doi:10.1007/978-4-431-54454-8.

[2] Olivieri, M., Pezzoli, M., Antonacci, F., and Sarti, A., "A Physics-Informed Neural Network Approach for Nearfield Acoustic Holography," *Sensors*, 21(23), p. 7834, 2021, ISSN 1424-8220, doi:10.3390/s21237834.

[3] Sethi, H., Pan, D., Dimitrov, P., Shragge, J., Roth, G., and Hester, K., "Hard enforcement of physics-informed neural network solutions of acoustic wave propagation," *Computational Geosciences*, 27(5), pp. 737–751, 2023, ISSN 1573-1499, doi:10.1007/s10596-023-10232-3.

[4] Schoder, S., "Physics-Informed Neural Networks for Modal Wave Field Predictions in 3D Room Acoustics," *Applied Sciences*, 15(2), 2025, ISSN 2076-3417, doi:10.3390/app15020939.

[5] Borrel-Jensen, N., Goswami, S., Engsig-Karup, A. P., Karniadakis, G. E., and Jeong, C.-H., "Sound propagation in realistic interactive 3D scenes with parameterized sources using deep neural operators," *Proceedings of the National Academy of Sciences*, 121(2), p. e2312159120, 2024, ISSN 0027-8424, 1091-6490, doi:10.1073/pnas.2312159120.

[6] Lu, L., Jin, P., and Karniadakis, G. E., "DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators," *Nature Machine Intelligence*, 3(3), pp. 218–229, 2021, ISSN 2522-5839, doi:10.1038/s42256-021-00302-5.

[7] Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A., "Fourier Neural Operator for Parametric Partial Differential Equations," in *International Conference on Learning Representations*, 2021.

[8] Kovachki, N. B., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A. M., and Anandkumar, A., "Neural Operator: Learning Maps Between Function Spaces," *CoRR*, abs/2108.08481, 2021.

[9] Anandkumar, A., Azizzadenesheli, K., Bhattacharya, K., Kovachki, N., Li, Z., Liu, B., and Stuart, A., "Neural Operator: Graph Kernel Network for Partial Differential Equations," in *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2019.

[10] Xu, B. and Sommerfeldt, S. D., "A hybrid modal analysis for enclosed sound fields," *The Journal of the Acoustical Society of America*, 128(5), pp. 2857–2867, 2010, ISSN 0001-4966, 1520-8524, doi:10.1121/1.3493429.

[11] Zou, C., Azizzadenesheli, K., Ross, Z. E., and Clayton, R. W., "Deep neural Helmholtz operators for 3-D elastic wave propagation and inversion," *Geophysical Journal International*, 239(3), pp. 1469–1484, 2024, ISSN 0956-540X, 1365-246X, doi:10.1093/gji/ggae342.

[12] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86(11), pp. 2278–2324, 1998, ISSN 00189219, doi:10.1109/5.726791.

[13] Ronneberger, O., Fischer, P., and Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," in N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Springer International Publishing, Cham, 2015, ISBN 978-3-319-24574-4.

[14] Rahman, M. A., Ross, Z. E., and Azizzadenesheli, K., "U-NO: U-shaped Neural Operators," *Transactions on Machine Learning Research*, 2023, ISSN 2835-8856.

[15] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *CoRR*, abs/1912.01703, 2019.

[16] Betcke, T. and Scroggs, M., "Bempp-cl: A fast Python based just-in-time compiling boundary element library." *Journal of Open Source Software*, 6(59), p. 2879, 2021, ISSN 2475-9066, doi:10.21105/joss.02879.

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, pp. 2825–2830, 2011.

[18] Molina-Moya, J., Martínez-Castro, A. E., and Ortiz, P., "An Iterative Parallel Solver in GPU Applied to Frequency Domain Linear Water Wave Problems by the Boundary Element Method," *Frontiers in Built Environment*, 4, p. 69, 2018, ISSN 2297-3362, doi:10.3389/fbuil.2018.00069.

[19] Li, Z., Kovachki, N. B., Choy, C., Li, B., Kossaifi, J., Otta, S. P., Nabian, M. A., Stadler, M., Hundt, C., Azizzadenesheli, K., and Anandkumar, A., "Geometry-Informed Neural Operator for Large-Scale 3D PDEs," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[20] Azizzadenesheli, K., "Machine Learning on Function Spaces," Tutorial at the International Conference on Machine Learning, 2024.