DOI: 10.61782/fa.2023.0047

forum acusticum 2023

THE APPLICATION OF FOURIER NEURAL OPERATOR NETWORKS FOR SOLVING THE 2D LINEAR ACOUSTIC WAVE EQUATION

Michael Middleton[1*] **Damian T. Murphy**[1] **Lauri Savioja**[2]
[1] AudioLab, School of Physics, Engineering and Technology, University of York, York, UK
[2] Department of Computer Science, Aalto University, Aalto, Finland

ABSTRACT

In recent years, data-driven operator approximation techniques have been explored as a means of solving physical problems described by ordinary and partial differential equations. In this paper, solutions to the linear 2D acoustic wave equation predicted by Fourier neural operator (FNO) networks are investigated in a square, free-field domain. The network's ability to generalise over variable excitation source positions in unseen locations is investigated. Furthermore, the network is tasked with learning progressively longer solutions in time to assess how the ratio of input to output data affects network prediction accuracy. Error between ground truth and predicted simulations is quantified and examined in an acoustics context.

Keywords: *Fourier neural operator, acoustic simulation, deep learning, FDTD, physics-informed neural network.*

1. INTRODUCTION

Wave-modelling methods for room acoustic simulation such as finite-difference time domain (FDTD) are desirable when simulating the acoustic response of a virtual domain given their inherent handling of complex wave phenomena like diffraction [1, 2]. FDTD is an inefficient process of numerical simulation as the domain must be highly over-sampled during discretisation to mitigate the effects of dispersion error [3]. As a result, it can take an impractical amount of time to run a single FDTD simulation when the domain sampling density is high [4]. Although recent advancements in computational power are addressing the issue for devices with large amounts of expendable working memory, this factor still limits the feasible use of FDTD for large-scale room acoustic simulation when compute resources are limited. For this reason, there is demand for a more time-efficient wave-modelling method of acoustic simulation that sidesteps the wasted computation present in an over-sampled FDTD scheme.

In this paper, deep learning techniques and their applicability to solving acoustic wave equations with variable excitation source positions are investigated. A novel system of wave-modelled acoustic simulation in two spatial dimensions using Fourier Neural Operator networks (FNOs) is presented [5, 6]. An improvement in computational efficiency over FDTD is achieved by representing the partial differential equation (PDE) that governs wave behaviour within the free-field domain implicitly as an FNO network, eliminating the wasted computation of an over-sampled FDTD scheme. The trained FNO therefore maps a space of input functions (these being domain excitation followed by an arbitrary number of simulated time-steps) to a space of output functions (the continued wave propagation until the simulation is complete) [5].

2. ACOUSTIC FDTD SIMULATION

The 2D wave equation is a second-order hyperbolic PDE describing air pressure u as a function of position (x, y) and time t [3]. c is a constant describing wave propagation speed in the medium. An impulse response (IR) can be sampled from an FDTD simulation by recording the pressure value at a point on the grid as it evolves from the moment of excitation until it returns to its fully mixed (or its ambient) state.

$$\frac{\partial^2 u}{\partial t^2} = c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{1}$$

*Corresponding author: michael.middleton@york.ac.uk.

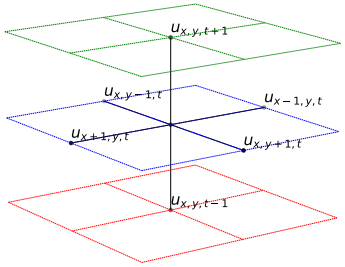

Figure 1. 4-point rectilinear 2D FDTD scheme to compute pressure at the grid point (x, y) at the next time step index $t + 1$.

FDTD has become one of the most widely researched approaches to acoustic simulation given its ability to simulate a broadband impulse response in a single pass [1, 2]. In the 2D-plus-time case explored in this paper, the acoustic wave equation (eq. 1) is solved numerically by discretising the time and space dimensions of the target domain. Space and time are represented as a three-dimensional matrix of the form (x, y, t). The air pressure at a position in the grid is computed from neighbouring indices in space and time as illustrated in figure 1. This process is repeated for all indices to complete a single update in time, which is in turn repeated until a desired number of time steps have elapsed.

Although FDTD simulations can operate in parallel and the individual finite-difference operations are simple multiplications they are still considered to be computationally intensive tasks given the typically vast number of operations that must be performed [7]. FDTD schemes with more than one spatial dimension are also subject to dispersion error as a result of domain discretisation. There is an inherent loss of information when quantising the particles of air within a domain and its boundaries to a grid as the number of sampled points will be linearly-spaced and far fewer in number. Ideally, waves travelling in any direction through the grid should propagate equally far; the direction travelled and the time taken should be linear in all directions. However, depending on the angle of propagation, part of the wavefront will travel through more grid points than other portions of the wavefront. The result is anisotropic propagation of error through the FDTD system that becomes more apparent in higher frequency ranges [2].

Dispersion errors manifest as phase distortions, which in turn restricts the quality of IRs retrieved from the simulated data. This dispersion error can be audible between simulated and measured environments, limiting the physical accuracy of auralised signals [8]. Some FDTD schemes have been proposed that cause dispersion to manifest nearly isotropically, spreading the error out in all directions to reduce its impact within localised areas [9]. Dispersion error cannot be entirely avoided however as domain discretisation is always required for FDTD.

3. FOURIER NEURAL OPERATOR NETWORKS FOR ACOUSTIC SIMULATION

3.1 Background

Neural networks are capable of representing time-intensive functions as parallelisable linear algebraic systems, which are often faster to compute than the target function that was learned. Using deep learning techniques to solve acoustic wave problems is therefore an attractive prospect. If the aim is to create a general wave-modelling solver represented as a neural network for a given domain, numerical wave solutions produced by FDTD can be used for training and testing data. Convolutional neural network designs have been used to learn local features in problems by passing a trainable kernel over the target domain [10]. This provides the network with the ability to learn local features effectively but does not typically capture the entire domain unless the convolution kernel size is scaled up to match the domain size.

Universal approximation theorem asserts that a multilayer feed-forward neural network can model any function to an arbitrary degree of accuracy provided that the layers are populated with sufficient neurons [11]. This theorem extends to the mapping of one space of functions to another (as opposed to value-to-value mapping that characterises function approximation) [12]. Mapping between functional spaces is known as operator learning, where the operator to be learned defines the PDE behaviour within the domain concerned. It can be described as higher level learning by the network than function approximation, accounting for changing physical variables (such as excitation position) within the system being modelled.

Fourier Neural Operator (FNO) networks [5, 6] have demonstrated excellent prediction accuracy and convergence dynamics when tasked with solving a host of forward and inverse physical problems that were unseen by the network during training [13, 14]. Fourier operator layers transform incoming signals from the previous layer

using an 3D n-point FFT, where n is the length of the incoming signal from the preceding layer padded to the next highest power of 2 with zeros. At this point, it is standard practice to drop some frequency bins from the transformed signal as a regularisation measure. Bins are discarded from the highest frequency downwards as the high-frequency content present in the Fourier layer spectra can cause the network to overfit [5]. The spectrum is transformed by a matrix of weights of an equivalent length before the signal is transformed back into physical space via IFFT. It is then transformed by a Gaussian Error Linear Unit activation function (GELU, eq. 2) and passed to subsequent layers [15]. The efficient global convolution is performed by multiplying the signal spectra and weight matrices in the frequency domain, equivalent to a more computationally expensive convolution operation in the time domain.

$$GELU(x) = 0.5x(1 + \tanh(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3))) \quad (2)$$

3.2 FNO for acoustic wave simulation

In acoustics, FNO networks have been used to solve the elastic wave equation for geophysical modelling, predicting wave propagation emanating from an excitation source in a free-field domain [16]. This example is also the most similar in context and scope to the experiments presented in this paper out of all FNO literature explored. Here, FNO prediction accuracy degraded as the estimated simulation evolved over time. This suggests that FNO is not suited for modelling wave propagation problems with many time steps to be predicted. The experimental work presented later in this paper is designed to test this hypothesis. So far, there have been no implementations of FNO solving the linear wave equation within the context of room acoustic simulation.

FNO network architecture excels under specific conditions. As FFT assumes signal periodicity, Fourier operator layers can only intrinsically handle periodic boundary conditions. Non-periodic boundaries are solved by padding the domain around its edges with zeros and incorporating techniques like skip connections, convolutional layers or two-step U-net paths into the Fourier layers [5, 17, 18]. Nonetheless, FNO has been shown to struggle when solving problems with complex boundary conditions, with predicted effects from boundary interactions lacking the finer detail present in the ground truth solutions [13]. For this reason, free-field solutions are explored in this paper. Whilst the boundary conditions remain non-periodic, there is no superposition of waves caused by reflections in the training data. In theory, this results in an easier problem for the network to model, with the aim of simulating more complex acoustic problems using FNO networks in future experiments in mind.

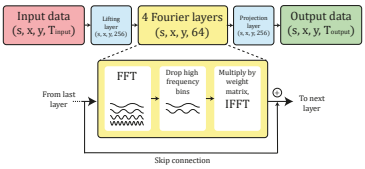

Figure 2. Data flow through the acoustic FNO network.

A diagram of an FNO network configured for acoustic wave simulation is presented in figure 2. The input functions used for training and inference $a(x, y)$ are a collection of FDTD simulations excited at a coordinate (x, y), rendered up to a given number of input time steps T_{input}. In data form, these functions can be represented as a 4-dimensional matrix of size (s, X, Y, T_{input}), where s is the number of input simulations. Likewise, the predicted output functions $u(x, y)$ are the continued evolution of the wavefield from the time step $T_{input} + 1$ to the end of the simulation. The output simulations are represented as a 4D matrix of size (s, X, Y, T_{output}), where T_{output} is the number of time steps to be predicted by the network to complete the solution. An additional fully-connected layer (known as the "lifting" layer) is added after the input and prior to the first Fourier layer to vectorise the inputs, representing them as a matrix of neural network weights and a layer of neurons. Another fully-connected layer is added before the network output (the "projection" layer) which returns the signal from the Fourier layers back to the physical domain required for the solution [5].

4. METHODOLOGY

The following tests were conducted to assess the prediction accuracy and learning dynamics of FNO networks applied to 2D acoustic linear wave problems in a free-field domain. It is often desirable to calculate many time-steps of a solution for acoustic simulation to allow for wave propagation and boundary interaction across as much of

the domain as possible. This formed the motivation behind the following experiments where the ratio of input to output data provided to a FNO network during training and prediction was increased successively. The data ratio was controlled by increasing the number of predicted time steps (T_{output}) whilst keeping the number of input time steps (T_{input}) constant. Error present in predicted transfer functions was quantified using Mean-Squared Error (MSE) in decibel scale, provided in eq. 3 where N is the number of samples in the true and predicted data sets.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(TrueData_i - PredictedData_i),$$
$$MSEdB = 20\log 10(|MSE|) \quad (3)$$

4 FNO networks were trained to predict solutions to the wave equation, with each modelling solutions with more time-steps than the last. Each FNO network was tasked with learning an operator defining wave behaviour within the domain when excited at a random position. The domain was square, measuring 128 samples in the x and y dimensions. Each simulation produced was 264 time-steps long, although each FNO network was trained on a different number of output time-steps according to table 1. Furthermore, each simulation was excited at a random position within the domain. To model free-field behaviour, the domain was symmetrically zero-padded by $ceil(\frac{T}{2})+1$ samples where T is the length of the entire time domain in samples. Padding occurred only in the spatial dimensions and solutions were un-padded after the simulation had completed. By exciting the domain only in the central region, waves were allowed to propagate beyond the area of interest. Any boundary reflections and wave superposition therefore occurred within the padded area, which was discarded to leave only the free-field propagation.

$$u_{x,y,t+1} = \frac{1}{2}(u_{x+1,y,t} + u_{x-1,y,t} + u_{x,y+1,t} + u_{x,y-1,t}) - u_{x,y,t-1} \quad (4)$$

A standard 4-point rectilinear stencil was used for the FDTD computation as shown in figure 1. The FDTD update equation used to produce wave simulation data is provided in eq. 4. A Dirac pulse was used as the excitation signal. This combination of update scheme and excitation signal was chosen to establish FNO learning potential from data sourced from a simple FDTD system. Simulation quality could be improved by using a different excitation signal (such as a Gaussian or Ricker wavelet) or by using an interpolated or higher-order FDTD stencil. The effects of improving the physical accuracy of training data on FNO optimisation can be benchmarked against the following results in future experiments.

Simulated wave propagation speed c was set to 343m/s. Distances between sampled points in space and time (Δx and Δt) measured 42.875mm and 88.388µs respectively. The Courant number was $\frac{c\Delta t}{\Delta x} = 0.707$. System sample rate was ≈11.31kHz. The FDTD system was physically accurate up to a limit of 1,108Hz as the standard rectilinear stencil is physically accurate only to a normalised frequency limit of 0.196 [3]. Figure 3 shows an example of training data provided to the network

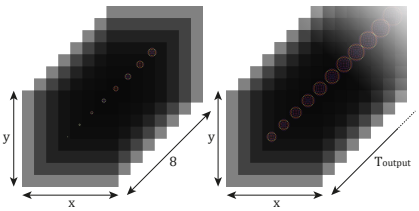

Figure 3. Example of training data. 8 input time steps are shown on the left which is mapped to the remaining T_{output} time steps.

Table 1. Relation between the volumes of input/output data in the tests conducted and the shape of the FDTD domains during and after simulation.

Test	T_{input} (samples)	T_{output} (samples)	Ratio	Domain size (x, y, t)	Padded domain size (x, y, t)
1	8	32	1:4	(128, 128, 40)	(170, 170, 40)
2	8	64	1:8	(128, 128, 72)	(202, 202, 72)
3	8	128	1:16	(128, 128, 136)	(266, 266, 136)
4	8	256	1:32	(128, 128, 264)	(394, 394, 264)

A unique tensorised FNO (TFNO) network with skip connections was trained for each experiment using the parameters described in table 2. TFNO architecture was used to keep performance overheads low by factorising the FFT computations within the Fourier layers and sharing weights between them [19]. These values were based on those used in the paper that proposed the FNO architecture [5] and then refined according to empirical results. The small change in model parameters between tests is

Table 2. FNO configuration.

Frequency bins preserved	(64, 64)
Num. Fourier layers	4
Fourier layer width	32
Lifting channel width	256
Projection channel width	256
Model parameters	Test 1: 7,382,080 Test 2: 7,390,304 Test 3: 7,406,752 Test 4: 7,439,648
Epochs trained	750
Learning rate	0.008
Learning rate scheduler	Cosine annealing ($T = 30$)
Optimiser	ADAM (default Pytorch config)
Loss functions	MSE, L2
Domain padding (x, y)	(128, 128)
Average GPU memory used	5.21GB
Factorisation type	Tucker (rank = 0.42)

explained by the changing number of output time steps; the shapes of hidden layers were kept constant between experiments.

Crucially, the signal was symmetrically zero-padded by 128 samples in the spatial dimensions after the lifting layer within FNO, tripling their effective width. Signals were then un-padded back to the original spatial domain size of $(128, 128)$ samples before the projection layer. This is a separate process to the zero-padding which occurred during FDTD simulation to produce free-field behaviour. As the boundary conditions present in the 2D free-field wave examples are non-periodic and the FFT operation within the Fourier layers assumes periodicity, the signal propagating through the FNO network must be padded to solve the boundary conditions properly.

Python 3.10 with the packages `Pytorch` and `neural-operator` were used to produce the neural network examples [5, 20]. Training was performed via CUDA on a Nvidia RTX 2060 Super GPU with 8Gb available working memory. 210 simulations were produced using the 2D FDTD scheme described above. 200 of these examples were used as training data whilst the remaining 10 were used for evaluation and were unseen by the network during training. A list of unique (x, y) excitation coordinates was produced using Latin Hyper-Cube sampling to mitigate clusters of points forming in localised areas that may occur when a purely stochastic sampling technique is used. The training data set is overall relatively sparse, with 200 locations on the 128 by 128 discretised grid comprising ≈1.22% of all measurable positions. Therefore, wave propagation behaviour emanating from ≈98.88% of possible excitation positions on the grid are unknown to the network.

To ensure a fair comparison and reproducibility between tests, all random seeds used in Python software packages were set to 0. Furthermore, the same set of training data was used to train each FNO network, although only Test 4 received the full 8 input time steps and 256 output time steps during training. For other tests where $T_{output} < 256$, fewer time steps were fed into the network for use as training data. At an average simulation time of of 775ms per FDTD solution, the training and testing data sets took a total of ≈2.71 minutes to compute.

5. RESULTS

5.1 Prediction accuracy

For the following examples, one ground-truth FDTD simulation and its corresponding FNO predictions were chosen from the testing set of 10 for illustration purposes. This example was excited at (x, y) position $(83, 73)$ on the grid and was unseen by the network during training. Figure 4 illustrates evolving wave fields at different time steps t, whilst figure 5 portrays the relative error between true and predicted results. It can be seen that the wavefront has been modelled accurately in all examples except Test 4, which fails to produce a physical solution whatsoever. Much of the training data for Test 4 consisted of background noise and dispersion artifacts after the wave had propagated past the boundaries, leading to the FNO being trained on low-quality data compared to other examples. For these reasons, predictions made by the FNO network trained for Test 4 will be excluded from further analysis. Although wavefront propagation is modelled well in Tests 1 to 3, the relative error surrounding it appears to increase as T_{output} also increases. Maximum absolute error for Tests 1 to 3 are below several physics-informed machine learning approaches to solving the acoustic wave equation [21, 22].

Figure 6 illustrates MSE in decibels between true and predicted transfer functions. Relative to the excitation position (x, y), on-axis positions are measured at $(x, y + 10)$ and off-axis positions are measured at $(x + 10, y + 10)$. The change in network prediction quality noted in figs. 4 and 5 is further reflected in the transfer functions obtained from true and estimated simulations.

Transfer functions were then taken for every point on the true and predicted grids. MSE in decibels for tests 1 to 3 were then plotted as heatmaps to indicate areas of relatively high error. In all examples, error was most apparent close to the excitation source (illustrated in figure 7). Despite FNO architecture being unable to handle non-periodic boundary conditions implicitly, there was no in-

dication that error manifested in greater amounts towards domain boundaries relative to the rest of the domain. This suggests that padding the spatial domains with zeros prior to transforming the signal using Fourier layers within the FNO is a suitable approach for solving problems with non-periodic boundary conditions. Table 3 describes the error between true and predicted transfer functions for on and off-axis positions.

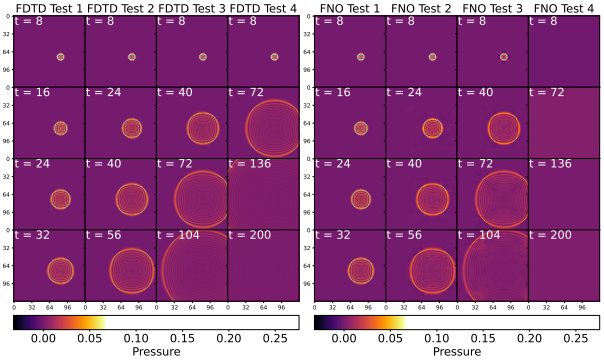

Figure 4. Wave propagation over time as simulated by FDTD (left) and predicted by FNO networks (right).

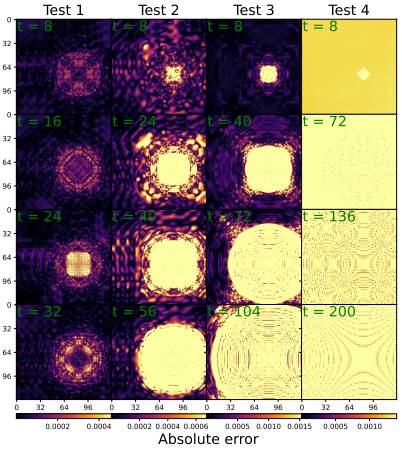

Figure 5. Absolute error between true and predicted examples.

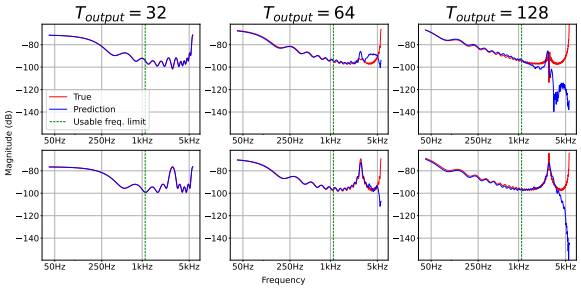

Figure 6. Transfer functions obtained from on-axis (top) and off-axis (bottom) positions relative to the excitation position.

Table 3. MSE in decibels for on and off-axis transfer functions pictured in figure 6.

Test	On-axis TF MSE (dB)	Off-axis TF MSE (dB)
1	-259.457	-232.209
2	-170.511	-174.251
3	-166.485	-167.217

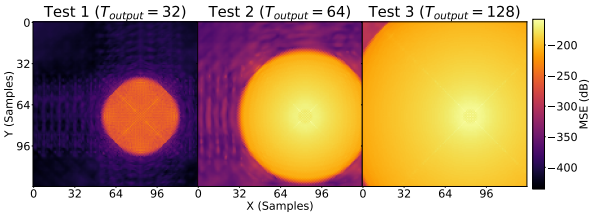

Figure 7. Heatmaps illustrating MSE for transfer functions extracted from every measurable position on the grid.

5.2 Training and inference time

Figure 5.2 illustrates the training losses of each FNO network over successive training iterations. The regular peaks in each loss curve are due to the cosine annealing learning rate scheduler. It can be seen that the initial loss value roughly doubles alongside T_{output}. This behaviour is also roughly mirrored in the final loss values for Tests 1 to 3. From this, it can be assumed that doubling the amount of required prediction data approximately halves

the effective convergence of the network on the training examples provided. In all examples, the testing loss gradients follow the training gradients closely, suggesting the network is generalising well and not showing symptoms of overfitting.

Test 4 appears to converge far more erratically than the other examples before the gradient "explodes" around epoch 550 to a value of over 14 million. It then manages to converge to a value just above the initial loss value before repeatedly plateauing, refusing to decrease beyond a lower threshold. The exploding gradient problem can be attributed to numerous factors including too low of a training batch size and insufficient or poor-quality data [23]. It can be assumed Test 4 failed to converge to a physical solution due to some combination of these factors. The issues with the FDTD training data produced for Test 4 are evident in the fluctuation of the loss gradient. Furthermore, the ratio of input to output data was causing convergence issues prior to the gradient explosion, suggesting that a 1:32 ratio is stretching the limits of FNO convergence under the conditions presented here.

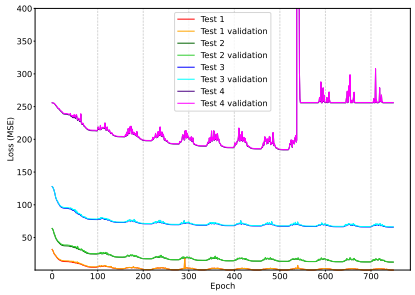

Figure 8. Training and testing loss descent for each network over successive iterations. The Y limit is set to 400 although Test 4 greatly exceeds this for several epochs.

As shown in table 4, FNO outperforms all equivalent FDTD computations by increasingly high margins. There is little meaningful difference between FNO inference times as the number of parameters remained broadly equal between models. On the other hand, the time taken to produce each simulation increased as the required number of simulated time steps increased. FNO networks, once trained, are therefore more time efficient at solving the 2D acoustic linear wave problem than an equivalent FDTD process under the conditions presented here.

Table 4. CPU-based FDTD computation time, FNO network inference time and FNO training time. FDTD computation time is only considered for T_{output} time steps. The time taken to compute the first 8 input time steps are not included in these figures.

Test	Domain size (x, y, t)	Padded domain size (x, y, t)	Avg. FDTD simulation time (per solution)	Avg. FNO prediction time (per solution)	FNO training time
1	(128, 128, 32)	(162, 162, 32)	10.8ms	1.3ms	2h 25m
2	(128, 128, 64)	(194, 194, 64)	62ms	1.2ms	2h 34m
3	(128, 128, 128)	(258, 258, 128)	258.4ms	1.3ms	3h 13m
4	(128, 128, 256)	(386, 386, 256)	704.2ms	1.4ms	4h 11m

6. CONCLUSION

It has been demonstrated that FNO networks are capable of learning and predicting high-quality solutions to the 2D linear acoustic wave equation. When the ratio of input to output data was 1:16 or lower, wavefield propagation was modelled accurately. However, the quality of the predicted results degraded as the number of output time steps required was increased. At a data ratio of 1:32, training the FNO network became entirely unstable. Whilst FNO predictions were faster to obtain than all equivalent FDTD solutions, the time taken to train the FNO network negates the improvement in computational efficiency as it would be faster to simulate all measurable positions on the grid numerically.

For future research, it is suggested that FNO is trained to model superposition of waves by introducing another excitation source into the experiments presented here, also placed at a random position within the domain. As superposition is a quality that arises from reflections at wave boundaries, it would be fruitful to assess if FNO is capable of solving these more complex examples before expanding the process to work in a room acoustics context. It would also be beneficial to replace the simple 4-point rectilinear FDTD stencil with an interpolated one to study the effects of dispersion error and input data quality on FNO learning dynamics [9]. It could also be beneficial to continue investigating why the FNO network failed to train at a data ratio of 1:32.

7. REFERENCES

[1] L. Savioja, T. J. Rinne, and T. Takala, "Simulation of room acoustics with a 3-d finite difference mesh," in *ICMC*, 1994.

[2] D. Botteldooren, "Finite-difference time-domain sim-

ulation of low-frequency room acoustic problems," *The Journal of the Acoustical Society of America*, vol. 98, pp. 3302–3308, 12 1995.

[3] K. Kowalczyk and M. van Walstijn, "Room acoustics simulation using 3-d compact explicit fdtd schemes," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 34–46, 2011.

[4] J. J. López, D. Carnicero, N. Ferrando, and J. Escolano, "Parallelization of the finite-difference time-domain method for room acoustics modelling based on cuda," *Mathematical and Computer Modelling*, vol. 57, no. 7, pp. 1822–1831, 2013. Public Key Services and Infrastructures EUROPKI-2010-Mathematical Modelling in Engineering Human Behaviour 2011.

[5] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," 2020.

[6] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar, "Fourier neural operator with learned deformations for pdes on general geometries," 2022.

[7] C. J. Webb and S. Bilbao, "Virtual room acoustics: A comparison of techniques for computing 3d-fdtd schemes using cuda," *Journal of the Audio Engineering Society*, may 2011.

[8] J. Saarelma, J. Botts, B. Hamilton, and L. Savioja, "Audibility of dispersion error in room acoustic finite-difference time-domain simulation as a function of simulation distance," *The Journal of the Acoustical Society of America*, vol. 139, no. 4, pp. 1822–1832, 2016.

[9] K. Kowalczyk and M. Van Walstijn, "Wideband and isotropic room acoustics simulation using 2-d interpolated fdtd schemes," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, pp. 78 – 89, 02 2010.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, p. 84–90, may 2017.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[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] V. Gopakumar, S. Pamela, L. Zanisi, Z. Li, A. Anandkumar, and M. Team, "Fourier neural operator for plasma modelling," 2023.

[14] B. Li, H. Wang, X. Yang, and Y. Lin, "Solving seismic wave equations on variable velocity models with fourier neural operator," 2022.

[15] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2020.

[16] T. Zhang, D. Trad, and K. Innanen, "Learning to solve the elastic wave equation with fourier neural operators," *GEOPHYSICS*, vol. 88, no. 3, pp. T101–T119, 2023.

[17] J. Kossaifi, N. B. Kovachki, K. Azizzadenesheli, and A. Anandkumar, "Multi-grid tensorized fourier neural operator for high resolution PDEs," 2023.

[18] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, and S. M. Benson, "U-fno – an enhanced fourier neural operator-based deep-learning model for multiphase flow," 2022.

[19] A. Tran, A. Mathews, L. Xie, and C. S. Ong, "Factorized fourier neural operators," 2023.

[20] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, "Neural operator: Learning maps between function spaces," *CoRR*, vol. abs/2108.08481, 2021.

[21] M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis, "Physics-informed neural networks (pinns) for wave propagation and full waveform inversions," *Journal of Geophysical Research: Solid Earth*, vol. 127, no. 5, p. e2021JB023120, 2022. e2021JB023120 2021JB023120.

[22] N. Borrel-Jensen, A. P. Engsig-Karup, and C.-H. Jeong, "Physics-informed neural networks for one-dimensional sound field predictions with parameterized sources and impedance boundaries," *JASA Express Letters*, vol. 1, p. 122402, 12 2021.

[23] G. Philipp, D. Song, and J. G. Carbonell, "The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions," 2018.