

Wavefield solutions from machine learned functions constrained by the Helmholtz equation



Tariq Alkhalifah^{a,*}, Chao Song^a, Umair bin Waheed^b, Qi Hao^c

^a Physical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal, 23955, Saudi Arabia

^b Department of Geosciences, King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia

^c Center for Integrated Petroleum Research, King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia

ARTICLE INFO

Keywords:

Helmholtz equation
Wavefields
Modeling
Neural networks
Deep learning

ABSTRACT

Solving the wave equation is one of the most (if not the most) fundamental problems we face as we try to illuminate the Earth using recorded seismic data. The Helmholtz equation provides wavefield solutions that are dimensionally reduced, per frequency, compared to the time domain, which is useful for many applications, like full waveform inversion. However, our ability to attain such wavefield solutions depends often on the size of the model and the complexity of the wave equation. Thus, we use here a recently introduced framework based on neural networks to predict functional solutions through setting the underlying physical equation as a loss function to optimize the neural network (NN) parameters. For an input given by a location in the model space, the network learns to predict the wavefield value at that location, and its partial derivatives using a concept referred to as automatic differentiation, to fit, in our case, a form of the Helmholtz equation. We specifically seek the solution of the scattered wavefield considering a simple homogeneous background model that allows for analytical solutions of the background wavefield. Providing the NN with a reasonable number of random points from the model space will ultimately train a fully connected deep NN to predict the scattered wavefield function. The size of the network depends mainly on the complexity of the desired wavefield, with such complexity increasing with increasing frequency and increasing model complexity. However, smaller networks can provide smoother wavefields that might be useful for inversion applications. Preliminary tests on a two-box-shaped scatterer model with a source in the middle, as well as, the Marmousi model with a source at the surface demonstrate the potential of the NN for this application. Additional tests on a 3D model demonstrate the potential versatility of the approach.

1. Introduction

A fundamental part of using surface seismic recorded data to illuminate the Earth is solving the wave equation (Claerbout, 1985). Solving the wave equation numerically constitutes the majority of the computational cost and complexity in applications like seismic modeling, imaging, and waveform inversion. Time-domain solutions of the wave equation dominate seismic applications as they are often efficient and comply with our natural understanding of wave evolution (Alterman and Karal, 1968; Richards and Aki, 1980). However, frequency-domain solutions, providing a reduction in dimensionality, recently gained additional attention with the rise of waveform inversion (Pratt, 1999; Sirgue and Pratt, 2004). Such solutions are obtained by inverting the stiffness matrix of the Helmholtz wave equation. However, the cost and

complexity of such a matrix inversion are intolerable as the model size increases, like for high frequencies or 3D applications (Clément et al., 1990), or the wave equation is complex, like those in anisotropic media (Wu and Alkhalifah, 2018a). This led (Sirgue et al., 2008) to suggest using time-domain modelling to obtain wavefields in the frequency domain for waveform inversion applications. However, such solutions are vulnerable to dispersion and stability errors (Wu and Alkhalifah, 2018b).

In recent years, researchers in our field have utilized machine learning algorithms to predict and classify fault locations, horizons, salt boundaries, facies, as well as, velocity models (Röth and Tarantola, 1994; Wrona et al., 2018; Araya-Polo et al., 2019; Holm-Jensen and Hansen, 2020). Whether supervised or semi or unsupervised training, neural networks have shown incredible flexibility in adapting to various

* Corresponding author.

E-mail address: tariq.alkhalifah@kaust.edu.sa (T. Alkhalifah).

<https://doi.org/10.1016/j.aiig.2021.08.002>

Received 2 June 2021; Received in revised form 13 August 2021; Accepted 17 August 2021

Available online 8 September 2021

2666-5441/© 2021 The Authors. Publishing Services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>).

geophysical tasks. Supervised learning was instrumental in predicting low frequencies to help full waveform inversion (FWI) converge to an accurate solution (Ovcharenko et al., 2019). Deep learning was also utilized to develop “a priori” models from well information to be used in FWI (Mosser et al., 2018; Zhang and Alkhalifah, 2019). Even wave propagation and wave equation solutions were facilitated using deep neural networks (Sorteberg et al., 2018; Hughes et al., 2019).

Within the framework of utilizing deep neural networks as universal function approximators (Liu and Nocedal, 1989) and under the banner of physics-informed neural networks (PINN), Raisse et al. (2019) demonstrated the network’s flexibility in learning how to extract desired functional solutions to nonlinear partial differential equations, utilizing the concept of automatic differentiation (Baydin et al., 2018). PINN has found considerable traction in solving partial differential equations (linear and nonlinear ones) ranging from cardiac activation mapping (Sahli Costabal et al., 2020) to steady-state Navier-Stokes equation (Dwivedi et al., 2021). Even within the framework of one dimensional wave propagation, PINN was utilized to establish flexible domain solutions of the wave equation (Kissas et al., 2020). In all of these applications, the predicted solutions were smooth, which is a requirement of NN as a universal function approximator (Pinkus, 1999). Wavefields are generally smooth, but they are often more complex in nature than other physical phenomena. The complexity of the wavefield increases at the source, as it represents a singularity in the solution. Thus, to use ML to predict wavefield solutions will require larger neural networks, which will eventually require larger computational resources. It will also require, like most numerical methods, careful sampling of the source region. As a result Song et al. (2021) suggested that we seek such NN functions for the scattered wavefield instead of the full wavefield. Specifically, to avoid the need for adaptive training points for the neural network (NN) to handle the expected source singularity bias, they solve for the scattered wavefield in the frequency domain and, thus, utilize the corresponding Lippmann–Schwinger equation as the loss function to train a deep fully connected neural network with inputs given by (randomly chosen) points in space (within the domain of interest) and outputs given by the complex scattered wavefield at these points. In their implementation, they focus on the application of their method on anisotropic media. Our objective in this paper is to evaluate what exactly an NN can predict of the wavefield solution, especially at a reasonable cost that can be utilized in practical applications.

Thus, here, we focus on the role of the model size, the solver, and frequency of the wavefield in predicting such scattered wavefield solutions. We will first compare solutions for the Helmholtz equation (McFall and Mahan, 2009) to those obtained for the scattered version of the Helmholtz equation for the same network size and hyperparameters. We will compare solutions at two different frequencies to assess the ability of the NN model in handling higher frequencies. This is followed by investigating the role of the NN model size in smoothing the wavefield solutions by evaluating the corresponding velocity for the predicted wavefields. We test the performance of the NN on a two box-shaped scatterer model, as well as, the Marmousi model, and in the process show the sensitivity of the approach to model size and frequency. Further testing on 3D will demonstrate the role of the NN parameters optimizer.

2. The Helmholtz equation

The wave equation is often solved in the time domain, and such solutions are attained by extrapolating the wavefield in time simulating what happens in nature (Richards and Aki, 1980). Wavefields in the time domain, however, are large, as they are given by a four dimensional function in 3D media or a three dimensional function in 2D media for a given source. In addition, the time axis, often, requires fine sampling to avoid aliasing, and an even finer sampling of time is required to avoid instability when solving the wave equation using finite-difference methods (Courant, 1928).

As a result of its linear nature, the wave equation can be easily formulated in the frequency domain. In this case, the resulting Helmholtz equation can be solved per frequency, with no requirements on frequency sampling, admitting a reduction in dimensionality of the wavefield solution. The Helmholtz equation in an acoustic, isotropic, constant density medium, described by the velocity, v , is given by:

$$(\nabla^2 + k^2)u(\mathbf{x}) = f(\mathbf{x}), \text{ where } k = \frac{\omega}{v}. \quad (1)$$

In this case, the solution of such an equation is a complex wavefield, $u = \{u_r, u_i\}$, defined in the Euclidean space, with $\mathbf{x} = \{x, y, z\}$, and a function of the angular frequency, ω . As a result, our time-domain solution is nothing but a superposition of frequency-domain solutions.

The point source nature of the source function, f , admits a singularity in the wavefield solution at the point source location. Such a singularity often causes inaccuracies in numerical solutions of the Helmholtz equation near the source. As suggested by Song et al. (2021), such a limitation can be addressed by solving the Lippmann–Schwinger form of the wave equation (Lippmann and Schwinger, 1950), instead. This equation is exact as we do not apply the Born approximation, as we maintain the true velocity on the right hand side of the equation. Thus, to somewhat mitigate the source singularity, we solve for the scattered wavefield, $\delta u = u - u_0$, where u_0 is the background wavefield satisfying the same wave equation (equation (1)) for the background velocity v_0 . Defining the velocity model perturbation, $\delta m = \frac{1}{v^2} - \frac{1}{v_0^2}$, the scattered wavefield satisfies

$$\left(\nabla^2 + \frac{\omega^2}{v^2}\right)\delta u = -\omega^2 \delta m u_0. \quad (2)$$

For the scattered wavefield, the source function is no longer confined in space, like the point source. It now depends on the perturbation model, which may extend the full space domain. To allow for efficient evaluation of the background wavefield, we consider the background velocity, v_0 , to be constant. For marine acquisition, we may choose this constant velocity to equal the water velocity to reduce the effect of the source singularity even further. The wavefield in an acoustic isotropic medium in 3D for a constant velocity and a point source located at \mathbf{x}_s , is given by:

$$u_0(\mathbf{x}) = \frac{e^{i\frac{\omega}{v_0}|\mathbf{x}-\mathbf{x}_s|}}{4\pi|\mathbf{x}-\mathbf{x}_s|}, \quad (3)$$

where i is the imaginary identity. For 2D applications, the solution for acoustic isotropic media is given by

$$u_0(\mathbf{x}) = \frac{i}{4}H_0^{(2)}\left(\frac{\omega}{v_0}|\mathbf{x}-\mathbf{x}_s|\right), \quad (4)$$

where $H_0^{(2)}$ is the zero-order Hankel function of the second kind, here $\mathbf{x} = \{x, z\}$ (Richards and Aki, 1980).

Solving for the scattered wavefield will allow us later to utilize random samples of the space domain to train the neural network to provide the functional solution representing the scattered wavefield in the frequency domain. The analytical solution for the background wavefield allows us to evaluate the wavefield instantly at any random point in the domain of interest. Next, we will see exactly how these formulations help the training of a functional neural network (NN).

3. The neural network solution

Based on the physics-informed neural network (PINN) framework introduced by Raissi et al. (2019), we utilize a neural network architecture using fully connected layers to approximate a function. This function is the scattered wavefield solution of equation (2). Hornik et al. (1989) have shown the ability of neural networks in approximating functions

that are smooth, like what we would expect from solving the wave equation. The input to the network, like a function, is a location in space, given in 2D by x and z coordinate values, and in 3D by x , y , and z coordinate values. The output of the network consists of the real and imaginary values of the complex scattered wavefield at the input location. Fig. 1 shows, in detail, the PINN concept for our application. We use the network to evaluate the wavefield and its second-order partial derivatives in x and z , which is needed to evaluate the Laplacian operator and the loss function. Thus, to train the network with equation (2), we use the following loss function:

$$f = \frac{1}{N} \sum_{j=1}^N \left[\omega^2 m^{(j)} \delta u_r^{(j)} + \nabla^2 \delta u_r^{(j)} + \omega^2 \delta m^{(j)} u_{r0}^{(j)} \right]^2 + \left[\omega^2 m^{(j)} \delta u_i^{(j)} + \nabla^2 \delta u_i^{(j)} + \omega^2 \delta m^{(j)} u_{i0}^{(j)} \right]^2, \quad (5)$$

where N is the number of training samples, and j is the training sample index. The two terms in the loss function correspond to the losses for the real (δu_r) and imaginary (δu_i) parts of the scattered wavefield, using the real (u_{r0}) and imaginary (u_{i0}) parts of the background wavefield. For the loss function, we chose the background model to be simple enough (homogeneous) so that the background wavefield can be evaluated analytically on the fly. The details of the fully connected deep network will be shared in the examples. The activation function between layers, other than the last layer, in all the examples is an inverse tangent. The last hidden layer connected to the output layer is linear. We chose to optimize the loss function using an Adam optimizer followed by limited memory BFGS iterations, all full-batch, gradient-based optimization algorithm (Liu and Nocedal, 1989). The L-BFGS admits smoother and more robust updates at a higher cost. We will show later the performance of both optimizers separately for comparison.

The NN functional provides a continuous representation of the wavefield, as opposed to a grid based representation, and such a continuous representation offers many benefits. We can attain the solution at any point, no interpolation is needed, and the domain of coverage can be of any shape. This can be beneficial in the presence of topography. However, this continuous functional representation has its limitations that appear mainly when the wavefield is complex, requiring larger networks and more advanced training. This appears to be the case when we have strong scattering and high frequencies. As a result, in the following tests, and as we introduce the approach, we will focus on lower frequencies and smoothed models. We will, nevertheless, also demonstrate these limitations as we compute the velocity model that corresponds to the predicted scattered wavefield. This can be achieved by solving the wave equation for the velocity model, and from equation (1), this implies:

$$v_{approx}^2(\mathbf{x}) = \Re \frac{\omega^2 u(\mathbf{x})}{f(\mathbf{x}) - \nabla^2 u(\mathbf{x})}, \text{ where } u(\mathbf{x}) = u_0(\mathbf{x}) + \delta u_{NN}(\mathbf{x}), \quad (6)$$

δu_{NN} is the predicted neural network scattered wavefield solution, and \Re corresponds to the real part. The process of solving equation (6) must be handled with care and there are many ways to do so including multiplying the numerator and denominator with the complex conjugate of the denominator and adding a small positive number to the denominator to avoid dividing over zero (Song and Alkhalifah, 2020).

4. Testing the NN

We will test this NN framework initially on two 2D examples trying to highlight some of its features and weaknesses. In the first example, we use a two box-shaped scatterer model with the source in the middle and we look at the dependency of the prediction on the frequency. Then, we apply the approach on the Marmousi model with the source on the surface, and we test the dependency of the solution on the size of the neural network. Finally, we apply the approach on a small 3D model and focus on the role of the optimizer. The objective of these tests is to study the ability of an NN to learn to a functional solution of the wave equation for the scattered wavefield as opposed to the wavefield itself.

4.1. A two-scatterer model

In the first model, we place two box-shaped perturbations in an otherwise homogeneous background as shown in Fig. 2(a). The model has 100 samples in both the x and z directions, with a sampling interval of 20 m. The corresponding (real part) of the 5 Hz wavefield for a point source (a delta function, one sample) in the center of the model is shown in Fig. 2(b). The background model is given by a constant velocity of 2 km/s, and the corresponding wavefield for the same source and frequency is shown in Fig. 2(c). If we subtract the two wavefields, we obtain the true scattered wavefield with the real part shown in Fig. 3(a), where the energy, as expected, reflects scattering from the two box-shaped scatterers. Using the loss function in equation (5), we train an 8-layer deep fully connected neural network with 20 neurons in each layer to represent the scattered wavefield solution. We randomly chose 5000 samples from the space domain (x_i, z_i) for the training, and train for 100000 epochs of Adam updates and 20000 of LBFGS updates. This number of samples used represents one fourth of the grid samples used to solve the Helmholtz equation and it was necessary to arrive to the scattered wavefield solution shown in Fig. 3(b). The difference between the true scattered wavefield and the NN predicted one is shown in Fig. 3(c). There are differences, but they are generally mild. The imaginary part of the scattered wavefield, not shown here, had similar accuracy.

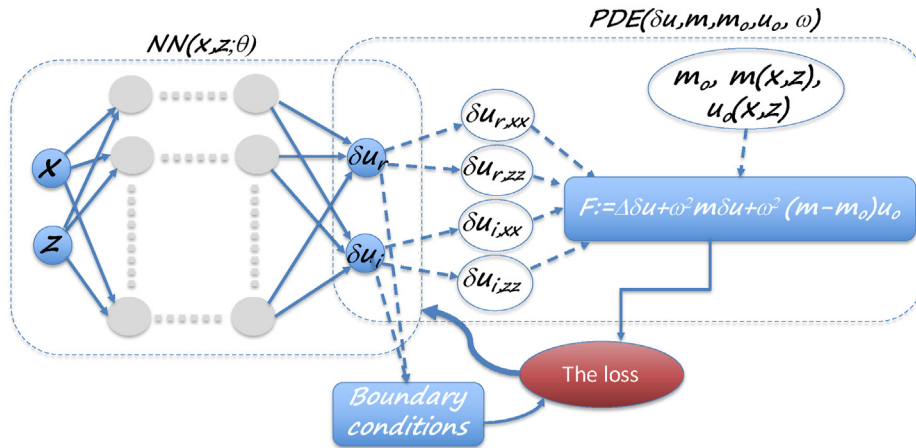


Fig. 1. The neural network architecture (left side dashed box) with inputs (x, z) and outputs given by the real and imaginary parts of the (scattered) wavefield. The network is trained using a loss function given by the scattered wave equation (right side dashed box), in which the Laplacian components ($\delta u_{r,xx}, \delta u_{r,zz}, \delta u_{i,xx}, \delta u_{i,zz}$) are evaluated using automatic differentiation of the NN. The loss function can be supported by boundary conditions.

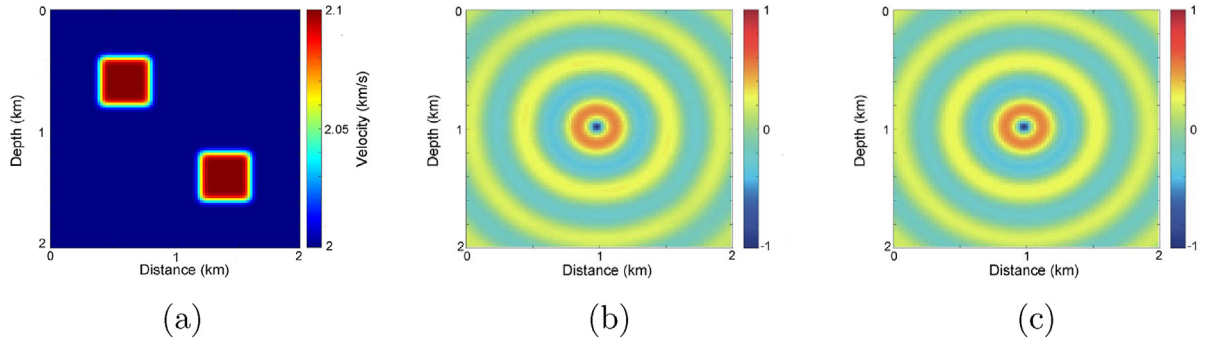


Fig. 2. a) A two-scatterer model. b) The real part of a 5 Hz wavefield for the velocity in (a) for a source in the middle, computed numerically, and considered true. c) The real part of the 5 Hz wavefield for the background model given by a velocity of 2 km/s, computed analytically.

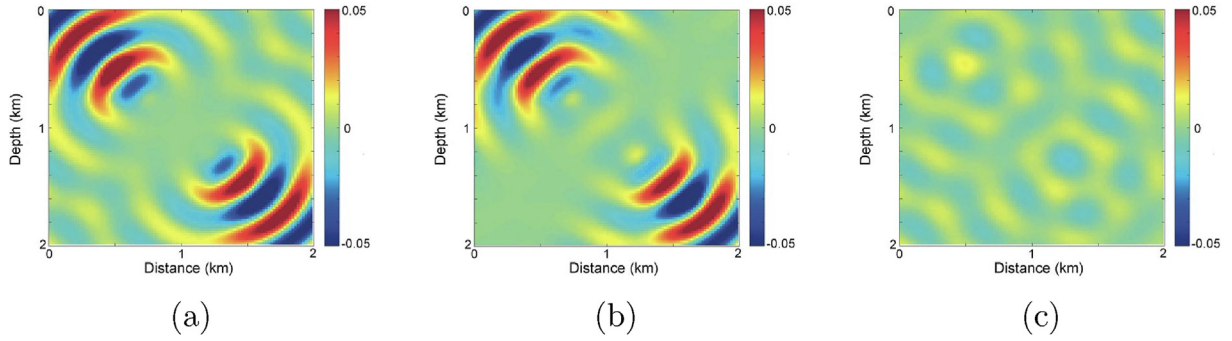


Fig. 3. a) The scattered wavefield given by the difference between the two wavefields in Fig. 2(b) and (c) (True and background wavefields). b) The NN predicted scattered wavefield on a regular grid. c) The difference between the actual and predicted scattered wavefields.

To justify inverting for the scattered wavefield instead of the wavefield directly using the Helmholtz wave equation, we repeat the exact experiment with the same number of randomly chosen training samples. The loss function, in this case, is given by the Helmholtz wave equation and to lessen the effect of a point source bias, we use an isotropic Gaussian source with a variance of 2.5. Fig. 4(a) and (b) show the real and imaginary parts, respectively, of the true wavefield for the two-scatterers model shown in Fig. 2(a). Fig. 4(c) and (d) show the real and imaginary parts, respectively, of the NN predicted wavefield for the same model. The difference is large and this is attributed to the source singularity in the Helmholtz equation, which requires better sampling of the source area in the training data.

For an 8 Hz wavefield, we use a larger network given by 40 neurons in each of the 8 layers and we use 10000 samples in the training. The real part of the predicted scattered wavefield is shown in Fig. 5(a). The difference between this predicted wavefield and the considered true numerical solution, plotted at the same scale as in Fig. 5(a), is small as shown in Fig. 5(b). To arrive to this solution, we used 150000 epochs of Adam updates and 20000 of LBFGS updates as demonstrated in Fig. 5(c). We use LBFGS at the end as it admits smoother updates we can rely on, but it is generally more expensive. The sudden change in the behaviour of the loss curve reflects the transition from Adam to LBFGS. Despite the larger network, compared to the 5 Hz case, and additional epochs, the cost increase was less than 100%, and that is much smaller than the additional cost we experience in solving for high frequencies using finite difference methods, which tend to increase exponentially.

4.2. The Marmousi model

Now, we test the utilization of the NN PDE in solving for the wavefield for a slightly smoothed Marmousi model (Fig. 6(a)). A point source is placed, this time, on the surface at location 4.5 km. We solve the Helmholtz equation numerically to obtain the 3 Hz frequency wavefield.

The background model is homogeneous with a velocity of 1.5 km/s in which we can solve the wavefield analytically. The difference between the true and the background wavefields, constituting the scattered wavefield is shown in Fig. 6(b) (real part) and 6(c) (imaginary part). The background wavefield and the model perturbations (difference between the true model and the homogeneous background) are used in the cost function given by equation (5) to invert for the NN parameters. We use, this time, a 10-layer network with {128, 128, 64, 64, 32, 32, 16, 16, 8, 8} neurons in the layers, respectively. We find that this configuration, given by larger dimensional layers early, is generally more effective. We use 10000 random sample points for the training and the resulting loss over 20000 epochs of training is shown in Fig. 7(a). The trained network is then used to evaluate the scattered wavefield on a regular grid and the resulting real part is shown in Fig. 7(b) and the imaginary part is shown in Fig. 7(c).

The difference between the true scattered wavefield and the NN predicted one is shown in Fig. 8(a) (real part) and 8(b) (imaginary part). The difference is generally small again, but here it seems to include more coherent energy corresponding to some of the scattering. In other words, the resulting NN predicted scattered wavefield is smoother than the true wavefield. This is an expected feature of NN when we avoid overfitting, the network acts as a smoother (Neal et al., 2018). We can further verify this smoothness feature by using the wave equation (equation (6)) to compute the velocity model corresponding to the predicted wavefield as shown in Fig. 8(c).

If we use a smaller network of 8 layers with {64, 64, 32, 32, 16, 16, 8, 8} neurons in the layers from left to right (we dropped the first two layers from the previous network), and use the same number of epochs, the resulting real part of the predicted scattered wavefield tends to be smoother as shown in Fig. 9(a). The difference between the predicted and true scattered wavefield is shown in Fig. 9(b) plotted at the same scale. The difference includes more energy than before. The increased smoothness of the wavefield can be verified by the resulting velocity

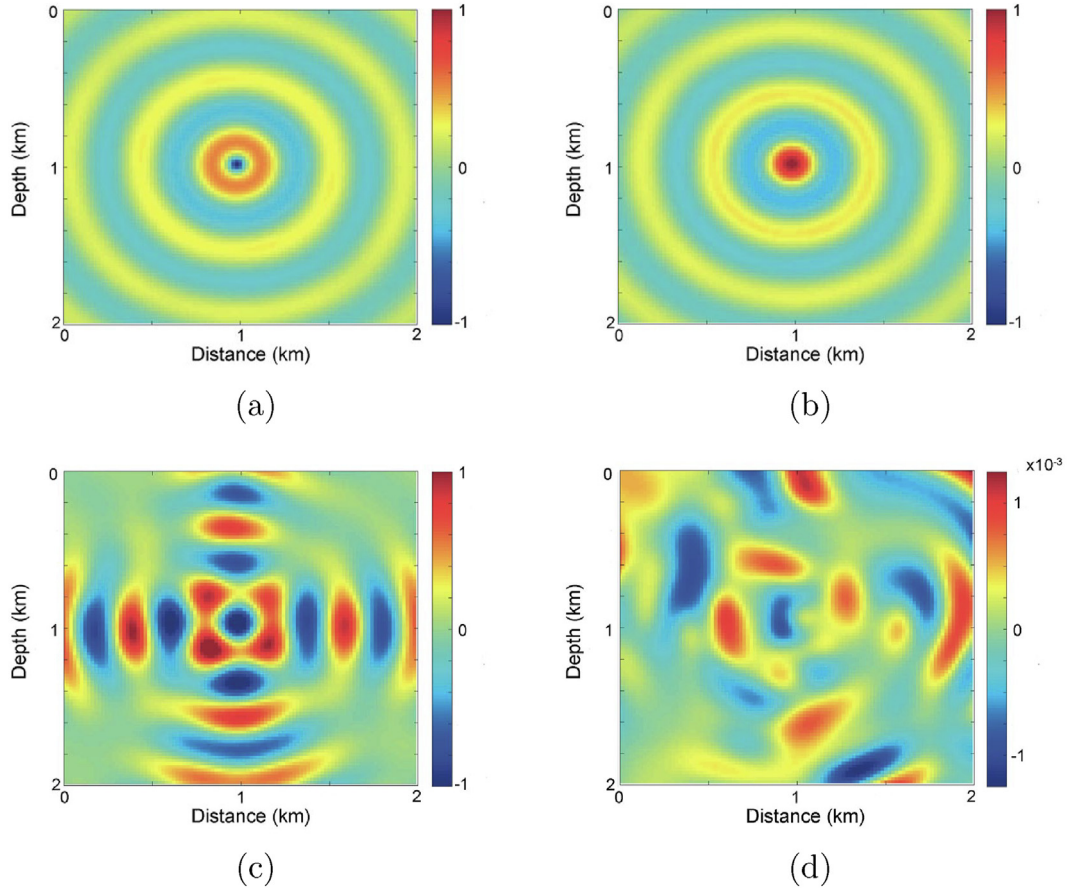


Fig. 4. a) The real part of the true wavefield. b) The imaginary part of the true wavefield. c) The real part of the NN predicted wavefield. d) The imaginary part of the NN predicted wavefield. The wavefields correspond to the velocity model in Fig. 2(a).

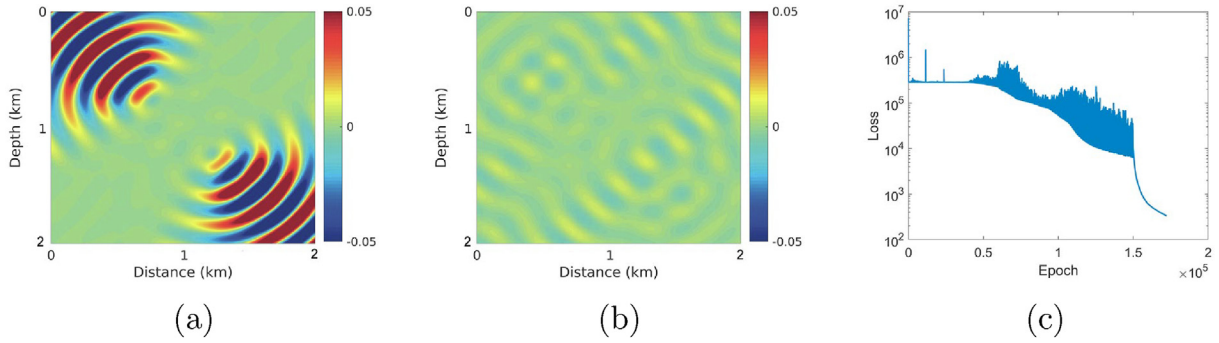


Fig. 5. a) The NN predicted scattered 8Hz wavefield for a source in the middle. b) The difference between the predicted scattered wavefield and the one computed numerically (true) plotted at the same scale as in a). c) The NN training loss function, which displays the loss using Adam followed by LBFGS.

model calculated from the wavefield and shown in Fig. 9(c). The velocity model is smooth compared to the true model, reflecting the smooth nature of the wavefield.

On the other hand, if we actually use a 12-layer network by adding two layers at the beginning of the original network with 256 neurons in each of them, we end up with a network given by {256, 256, 128, 128, 64, 64, 32, 32, 16, 16, 8, 8} neurons in the layers from left to right. Using the same number of epochs in the training of the same random samples in space, we end up with the predicted scattered wavefield with the real part shown in Fig. 10(a). The difference between the predicted and true scattered wavefield is shown in Fig. 10(b) plotted at the same scale. It contains less energy and that again can be verified by the resulting velocity model calculated from the wavefield and shown in Fig. 10(c). The

velocity model is clearly sharper and it is reasonably close to the true velocity model shown in Fig. 6(a).

Thus, a larger network can provide more accurate wavefields, but for applications in gradient calculation for velocity model update, a perfect scattered wavefield is not necessary. The cost of training the 12-layer network is 50% higher than the 10-layer one in spite that the number of network model parameters increased by 4. Meanwhile, the cost of the training the 8-layer network is two-third the cost of training the 10-layer network, while the number of network parameters is one-fourth of that of the 10-layer network. So, in summary, using this network architecture, the cost of the training will increase by about 50% with the addition of two layers of double the size of the first (largest) layer, and we end up with a higher resolution wavefield.

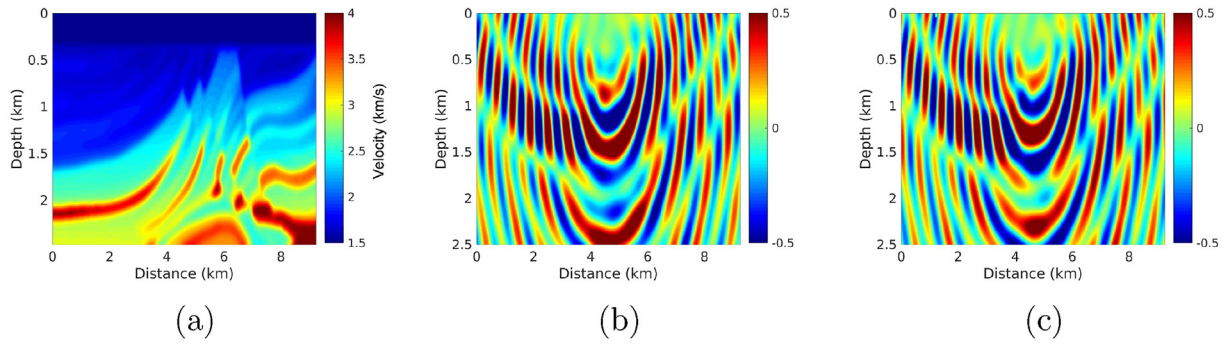


Fig. 6. a) The Marmousi model. b) The real part of the resulting 3 Hz wavefield for a source on the surface located in the middle. c) The imaginary part of the wavefield.

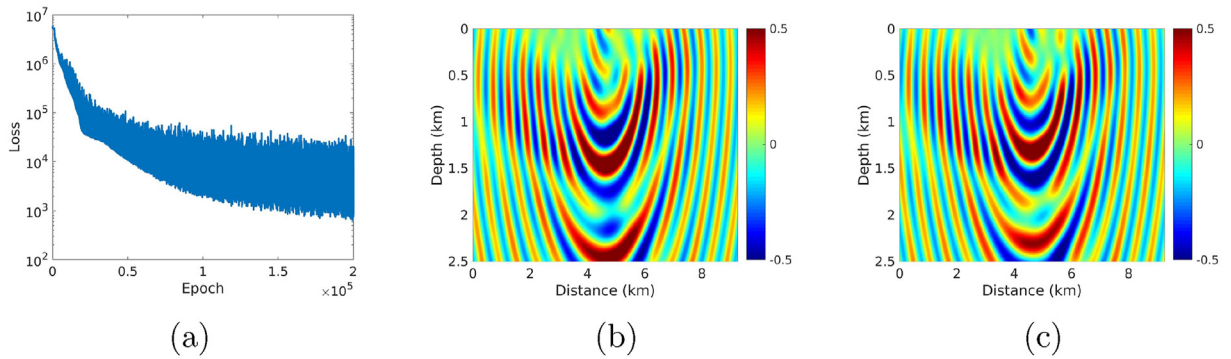


Fig. 7. a) The loss function for the training of the NN. b) The real part of the predicted scattered wavefield from the NN network. c) The imaginary part.

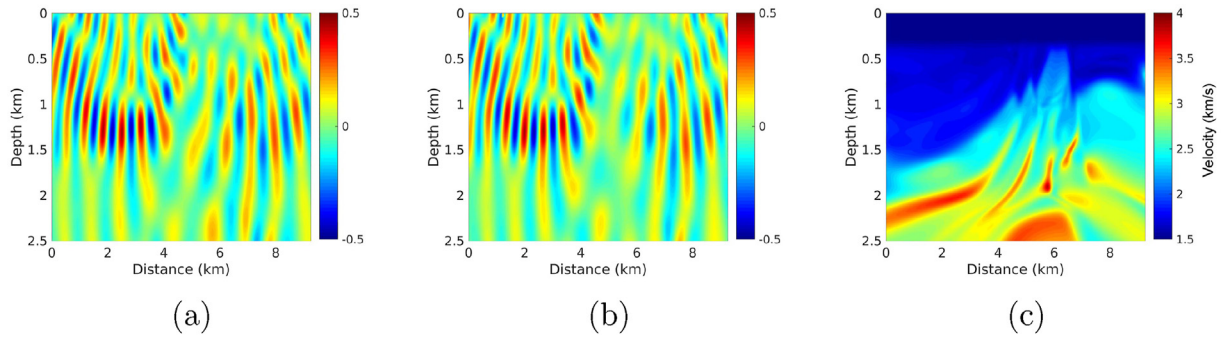


Fig. 8. a) The difference between Fig. 6(b) and 7(b) (True and predicted real parts of the scattered wavefields). b) The difference between Fig. 6(c) and 7(c) (True and predicted imaginary parts of the scattered wavefields). c) The velocity model computed from the predicted wavefield.

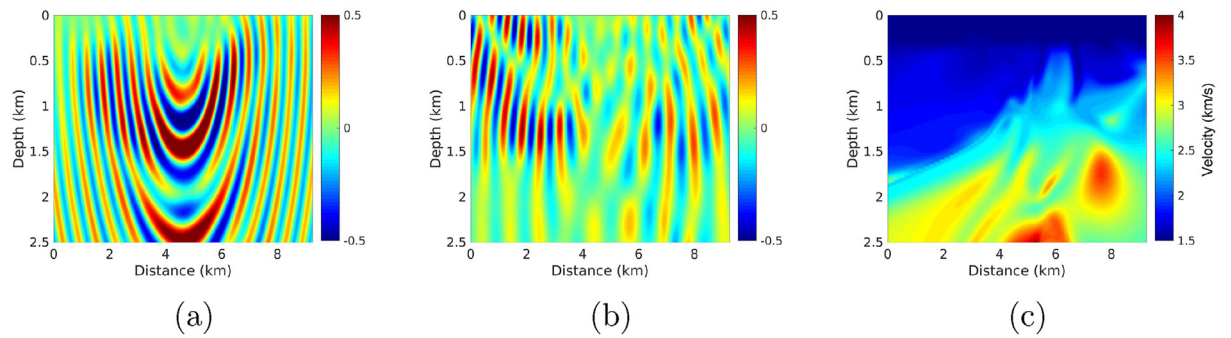


Fig. 9. a) The real part of the predicted scattered wavefield using an 8-layer network. b) The difference between Fig. 6(b) and 9(a) (True and predicted real parts of the scattered wavefields). c) The velocity model computed from the predicted wavefield.

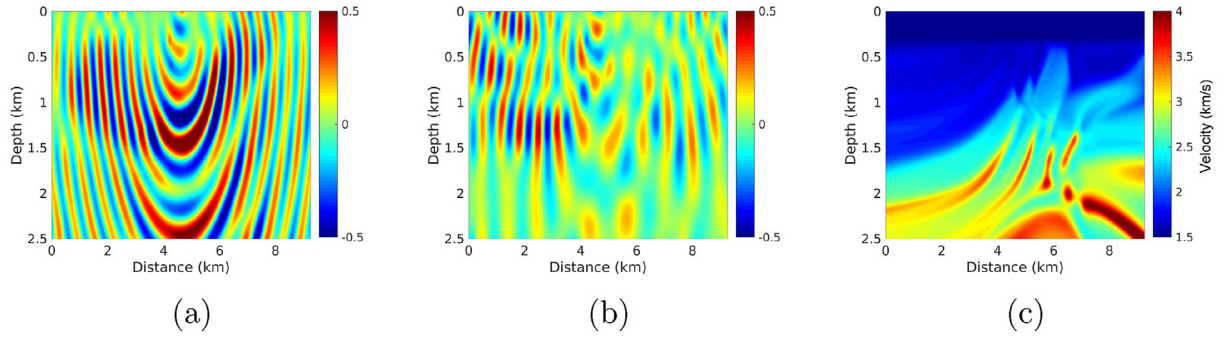


Fig. 10. a) The real part of the predicted scattered wavefield using a 12-layer network. b) The difference between Fig. 6(b) and 10(a) (True and predicted real parts of the scattered wavefields). c) The velocity model computed from the predicted wavefield.

4.3. A 3D example and the optimizer

We consider a 3D cube extracted from the SEG/EAGE Overthrust model (Aminzadeh et al., 1994) and slightly smoothed as shown in Fig. 11(a). In this test, we also test the performance of two NN optimization algorithms, specifically Adam and LBFGS. The background homogeneous model has a velocity of 3.2 km/s. The difference between the Helmholtz computed wavefield for 10 Hz and the background wavefield for the same frequency provides us with the true scattered wavefield for a source located in the middle, with the real part of this scattered wavefield shown in Fig. 11(b). Since the Adam optimizer admits, as we saw earlier, turbulent loss functions, for this example, we will test the performance of the two network optimization algorithms separately: The Adam optimizer and the limited memory BFGS algorithm. In this comparison, for the Adam optimizer we use 150000 epochs and for the LBFGS we use 50000 epochs in the training, and the neurons in each of the 8 hidden layers are {64, 64, 32, 32, 16, 16, 8, 8}, as shown in Fig. 11(c).

The loss function for the Adam optimizer is shown in Fig. 12(a) and, in average, the loss reduces per epoch, but with Adam we notice the loss function is bumpy and this has been realized by others in the application of PINN. Considering the log scale of the vertical axis, such alterations are slightly exaggerated in the Figure. The Adam update can be made smoother by using a smaller learning rate, but that increases the number of epochs. The real part of the resulting predicted scattered wavefield is shown in Fig. 12(b), and the difference between it and the true scattered wavefield is shown in Fig. 12(c). The difference, plotted at the same scale as the scattered wavefield, is small. On the other hand, using an LBFGS optimizer, the loss function is smoother as shown in Fig. 13(a), and seemingly admits a lower loss. However, the predicted scattered wavefield from the network, shown in Fig. 13(b), looks almost identical to the one obtained from the Adam optimizer. This can be further verified by observing the difference between the predicted scattered wavefield and the true one shown in Fig. 13(c), plotted again at the same scale. The errors seem to be similar to that observed with the Adam optimizer. So the effective differences in the loss has limited effect on the wavefield.

Considering that the results from using the Adam and the LBFGS optimizers are similar, and since the LBFGS optimizer is more expensive depending on the memory parameters, we suggest, as Raissi et al. (2019) suggested, using initially the Adam optimizer, which is extremely popular in ML optimizations.

5. Discussions

Machine learning provides a platform for predicting outputs by mainly recognizing the corresponding patterns of the inputs through a training process. Wavefields are by definition smooth and differentiable other than at the source, which is a requirement for a functional NN solution output (Hornik et al., 1989). The input to the proposed neural network is a location in space and the output is the wavefield (or the scattered wavefield) that satisfies a cost function given by the Helmholtz equation or its variant. These equations depend on the velocity and the source function, or in our implementation, the background wavefield and the velocity perturbations. So the NN weights and biases are expected to absorb the velocity and source information in their efforts to learn to predict the solution of the wave equation. As the NN tries to learn the wavefield, the source location is, especially, influential as it determines the epicentre of the wavefield. The velocity has generally a second-order effect on the wave shape, compared to the source location. Meanwhile, the frequency mainly controls the wavelength. These facts can help us decide on how to use the network for any successive wavefield solutions. For example, solutions for any additional velocity perturbations that maybe extracted from any velocity model update procedure like migration velocity analysis or full waveform inversion. Specifically, the current NN model can be used as an initial model for training on the updated velocity.

By using the Born (Lippmann–Schwinger equation) version of the wave equation instead of the Helmholtz solver, we avoided the bias required in better sampling the source region in the training of the network, necessary to mitigate the effect of the source singularity. So by using a homogeneous background model in which the wavefield can be

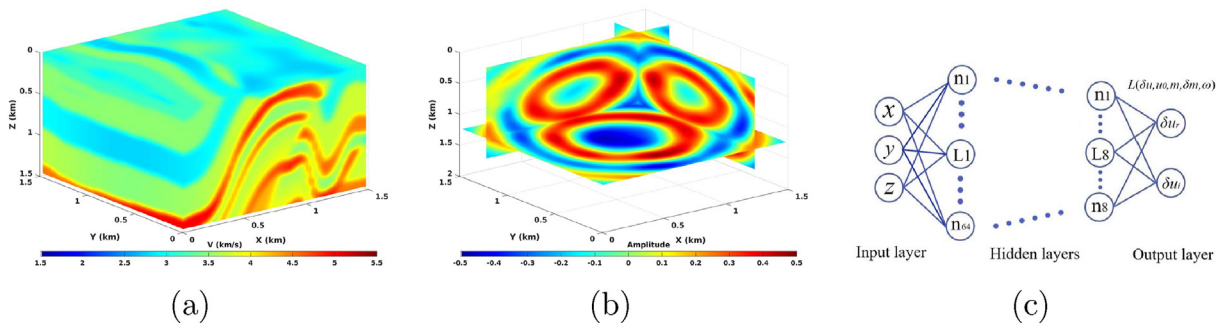


Fig. 11. a) A 3D model. b) The real part of the resulting 10 Hz wavefield for a source on the surface located in the middle. c) The NN architecture with dimensions of the 8 hidden layers given by (64,64,32,32,16,16,8,8) from shallow to deep.

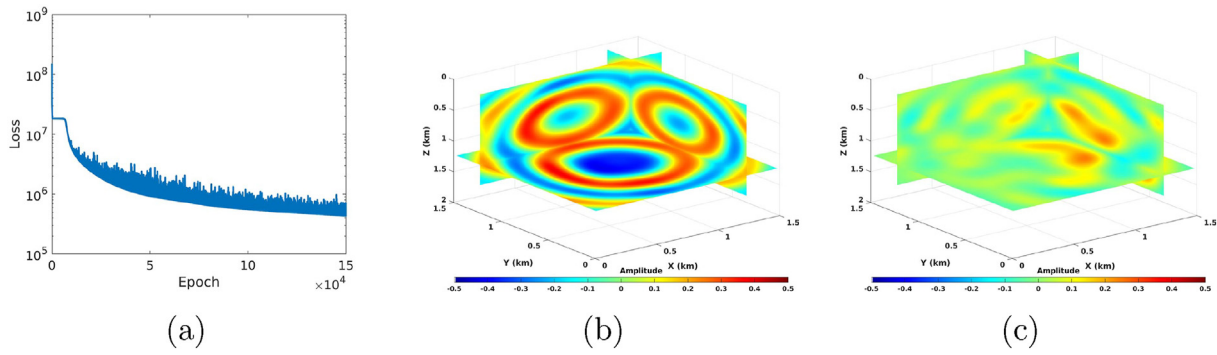


Fig. 12. a) The loss function for the training of the NN using an Adam optimizer. b) The real part of the predicted scattered wavefield from the NN. c) The difference between the predicted wavefield and the true one in Fig. 11(b).

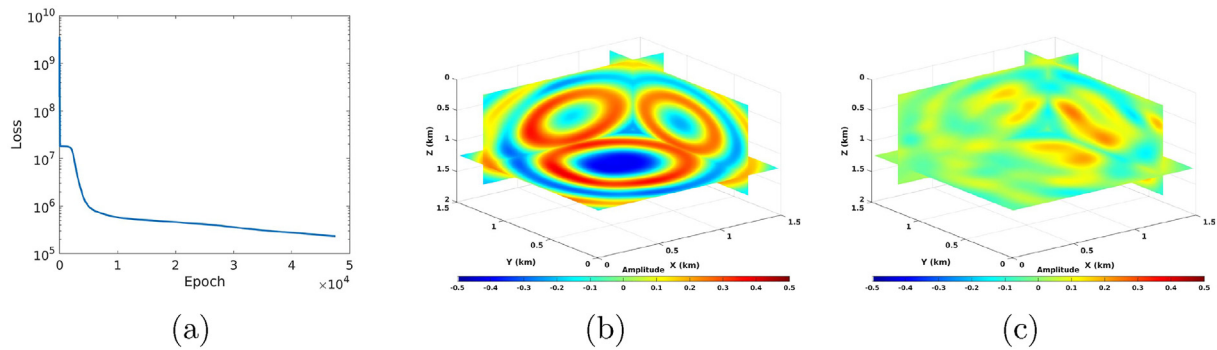


Fig. 13. a) The loss function for the training of the NN using an LBFGS optimizer. b) The real part of the predicted scattered wavefield from the NN. c) The difference between the predicted scattered wavefield and the true one in Fig. 11(b).

solved analytically (and instantly), the perturbations (difference between true and background models) will often extend the model domain, and random samples of the model space can be used in the training. To remove the signature of the source singularity from the scattered wavefield, the background velocity should be chosen equal to the velocity at the source. For marine data, this is given by a velocity of approximately 1.5 km/s. In general, the accuracy of predicting the wavefield depends mainly on the complexity of the wavefield. Near the source, the wavefield is complex, but it could also be complex in many other areas depending on the velocity model. In this case, we will need a larger neural network model, as well as better sampling of these complex regions. The random training points will often sample the domain reasonably well, but it does not take into account the complexity of the wavefield. From our observation, and especially with the Marmousi model, for a fixed neural network model size and random sampling of the training, the NN provides a uniformly smooth wavefield compared the true one. We can also utilize the concept of collocation points, as well, as adaptively adding points in regions requiring more emphasis (i.e., with high residuals) (McFall and Mahan, 2009). These options have their own cost. The number of training samples used to train the NN to predict the scattered wavefield is a delicate matter (Zhou and Wu, 2011). It directly affects the cost of the training and yet it is necessary that we have enough samples to accurately train the network to predict the wavefield. Training examples and their influence on the training is an ongoing research topic in the machine learning community.

Another feature of using a cost function for NN training, like in Raissi et al. (2019), we can fit the boundary condition or even the data as part of the objective and, thus, include two or more terms in the cost function. For the data fitting case, this amounts to something like the wavefield reconstruction method (Van Leeuwen and Herrmann, 2013), which is also solved in the frequency domain and faces similar challenges with regard to data and model sizes (Song et al., 2021). Thus, an important

feature of such neural network wavefield solutions is the fixed memory requirements, mainly controlled by the architecture of the network. It is, thus, independent of the size of the gridded velocity model. As we saw, the errors associated with reducing the size of the network are not of the dispersion kind, like for conventional numerical solvers considering the velocity model discretization, but they manifest themselves in smoothing the wavefield.

The cost of training the neural network depends on the number of random samples used in the training (the training set) to optimize the network parameters, as well as, the size of the network. As the trained NN tries to fit the loss function given by the wave equation or its Born form and any boundary conditions, the size of the network, including the number of layers and neurons, defines the details of the predicted wavefield. As we saw, smaller networks, cheaper to train, admit smoother wavefields. Thus, the size of the neural network will depend on the application and the objective involved in the application. For more accurate wavefields, the cost can be much higher than conventional numerical methods, as we are solving an optimization problem starting from random initialization. Transfer learning, in which we use the previously trained network as an initial NN model for a slightly updated wavefield due to a small shift in the source location or a change in the velocity model, can reduce some of this cost (Song and Alkhalifah, 2021). On the other hand, an application like waveform inversion may require smoother wavefields in the early iterations as we build up the background low wavenumber model and, thus, a neural network wavefield can help us obtain smoother velocities and/or gradients without the need for smoothing or spatial filtering. Intuitively, the smoother the wavefield, the less NN parameters we will need, which bodes well for efficient low frequencies. Interestingly, the cost increase of enlarging the network to predict wavefields for higher frequencies, is less severe than that needed for finite difference methods. However, we noticed that for high frequencies the training is harder as we use inverse tangent activation

functions to develop the sinusoidal wavefield. An alternative is to use sine activation functions, which we plan to investigate in the near future.

Though, at this early stage of using this functional ML solution, the cost may not justify replacing the regular Helmholtz solvers for 2D, and maybe even 3D, isotropic examples, the potential and flexibility of the approach will induce more interesting applications. This includes applications on more complex physics, like anisotropy (i.e. (Song et al., 2021)) and elasticity, where the regular Helmholtz solver becomes impractical. This includes applications involving complex wavefields in 3D, like those for orthorhombic anisotropy, even if the perturbations are small. Conventional frequency-domain solutions for such complex physics are hard and somewhat beyond our capability, especially if the model size is large. Machine learning is an optimal platform for large problems and large data as it adapts to this objective and learns the appropriate solution by recognizing patterns. Thus, for complex physics, our cost function will change, and possibly include more terms, but the training machinery is the same and does not involve solving for the inverse of a large matrix.

6. Conclusions

We trained a neural network to provide functional solutions to the Helmholtz equation. To avoid the point source singularity, we use a fully connected network that takes in space coordinates within the domain of interest and outputs the real and imaginary parts of the scattered (instead of the full) wavefield in the frequency domain. The background velocity is homogeneous, which admits analytical solutions of the background wavefield. With automatic differentiation, the network is capable, as well, of evaluating the partial derivatives of the scattered wavefield necessary to evaluate the loss function given by the Lippmann Schwinger form of the wave equation. This loss function is used to update the network parameters. With a scattered wavefield corresponding to perturbations spanning the space domain, the training of the network can be performed with less random samples. However, the network and the number of samples should increase with an increase in frequency. This increase in cost is far less than the exponential increase we experience in the case of increasing frequency for finite difference methods. Overall, the output wavefields are somewhat smoother than the exact ones, and this can be attributed to the compromise feature of our relatively small network and this feature might be useful for applications like waveform inversion. In fact, the smaller the network, the smoother the output scattered wavefield.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank KAUST and KFUPM for their support, and the seismic wave analysis group (SWAG) for constructive discussions.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.aiig.2021.08.002>.

References

- Alterman, Z., Karal Jr., F., 1968. Propagation of elastic waves in layered media by finite difference methods. *Bull. Seismol. Soc. Am.* 58, 367–398.
- Aminzadeh, F., Burkhard, N., Nicoletis, L., Rocca, F., Wyatt, K., 1994. SEG/EAGE 3-D modeling project: 2nd update. *Lead. Edge* 13, 949–952.

- Araya-Polo, M., Farris, S., Florez, M., 2019. Deep learning-driven velocity model building workflow. *Lead. Edge* 38, 872a1–872a9.
- Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2018. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* 18.
- Claerbout, J.F., 1985. *Imaging the Earth's Interior*, vol. 1. Blackwell scientific publications Oxford.
- Clément, F., Kern, M., Rubin, C., 1990. Conjugate gradient type methods for the solution of the 3D Helmholtz equation. In: *Proceedings of the First Copper Mountain Conference on Iterative Methods*.
- Courant, R., 1928. On the partial difference equations of mathematical physics. *Math. Ann.* 100, 32–74.
- Dwivedi, V., Parashar, N., Srinivasan, B., 2021. Distributed learning machines for solving forward and inverse problems in partial differential equations. *Neurocomputing* 420, 299–316.
- Holm-Jensen, T., Hansen, T.M., 2020. Linear waveform tomography inversion using machine learning algorithms. *Math. Geosci.* 52, 31–51.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Network*. 2, 359–366.
- Hughes, T.W., Williamson, I.A., Minkov, M., Fan, S., 2019. Wave physics as an analog recurrent neural network. *Science advances* 5, eaay6946.
- Kissas, G., Yang, Y., Hwuang, E., Witschey, W.R., Detre, J.A., Perdikaris, P., 2020. Machine learning in cardiovascular flows modeling: predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.* 358, 112623.
- Lippmann, B.A., Schwinger, J., 1950. Variational principles for scattering processes. I. *Phys. Rev.* 79, 469.
- Liu, D.C., Nocedal, J., 1989. On the limited memory BFGS method for large scale optimization. *Math. Program.* 45, 503–528.
- McFall, K.S., Mahan, J.R., 2009. Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions. *IEEE Trans. Neural Network*. 20, 1221–1233.
- Mosser, L., Dubrule, O., Blunt, M.J., 2018. Stochastic reconstruction of an oolitic limestone by generative adversarial networks. *Transport Porous Media* 125, 81–103.
- Neal, B., Mittal, S., Baratin, A., Tantia, V., Scicluna, M., Lacoste-Julien, S., Mitliagkas, I., 2018. A Modern Take on the Bias-Variance Tradeoff in Neural Networks, *arXiv Preprint arXiv:1810.08591*.
- Ovcharenko, O., Kazei, V., Kalita, M., Peter, D., Alkhalifah, T., 2019. Deep learning for low-frequency extrapolation from multioffset seismic data. *Geophysics* 84, R989–R1001.
- Pinkus, A., 1999. Approximation theory of the MLP model. *Acta Numerica* 1999 88, 143–195.
- Pratt, R.G., 1999. Seismic waveform inversion in the frequency domain, Part 1: theory and verification in a physical scale model. *Geophysics* 64, 888–901.
- Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707.
- Richards, P.G., Aki, K., 1980. *Quantitative Seismology: Theory and Methods*, vol. 859. Freeman, New York.
- Röth, G., Tarantola, A., 1994. Neural networks and inversion of seismic data. *J. Geophys. Res.: Solid Earth* 99, 6753–6768.
- Sahli Costabal, F., Yang, Y., Perdikaris, P., Hurtado, D.E., Kuhl, E., 2020. Physics-informed neural networks for cardiac activation mapping. *Frontiers in Physics* 8, 42.
- Sirgue, L., Pratt, R.G., 2004. Efficient waveform inversion and imaging: a strategy for selecting temporal frequencies. *Geophysics* 69, 231–248.
- Sirgue, L., Etgen, J., Albertin, U., 2008. 3D frequency domain waveform inversion using time domain finite difference methods. In: *70th EAGE Conference and Exhibition Incorporating SPE EUROPEC 2008*. European Association of Geoscientists & Engineers.
- Song, C., Alkhalifah, T.A., 2020. Efficient wavefield inversion with outer iterations and total variation constraint. *IEEE Trans. Geosci. Rem. Sens.* 58, 5836–5846. <https://doi.org/10.1109/TGRS.2020.2971697>.
- Song, C., Alkhalifah, T., 2021. Wavefield Reconstruction Inversion via Physics-Informed Neural Networks, *arXiv*.
- Song, C., Alkhalifah, T., Waheed, U.B., 2021. Solving the frequency-domain acoustic VTI wave equation using physics-informed neural networks. *Geophys. J. Int.* 225, 846–859.
- Sorteberg, W.E., Garasto, S., Pouplin, A.S., Cantwell, C.D., Bharath, A.A., 2018. Approximating the Solution to Wave Propagation Using Deep Neural Networks, *arXiv Preprint arXiv:1812.01609*.
- Van Leeuwen, T., Herrmann, F.J., 2013. Mitigating local minima in full-waveform inversion by expanding the search space. *Geophys. J. Int.* 195, 661–667.
- Wrona, T., Pan, I., Gawthorpe, R.L., Fossen, H., 2018. Seismic facies analysis using machine learning. *Geophysics* 83, O83–O95.
- Wu, Z., Alkhalifah, T., 2018a. An efficient Helmholtz solver for acoustic transversely isotropic media. *Geophysics* 83, C75–C83.
- Wu, Z., Alkhalifah, T., 2018b. A highly accurate finite-difference method with minimum dispersion error for solving the helmholtz equation. *J. Comput. Phys.* 365, 350–361.
- Zhang, Z.-D., Alkhalifah, T., 2019. Regularized elastic full-waveform inversion using deep learning. *Geophysics* 84, R741–R751.
- Zhou, Y., Wu, Y., 2011. Analyses on influence of training data set to neural network supervised learning performance. In: *Advances in Computer Science, Intelligent System and Environment*. Springer, pp. 19–25.