Wave simulation in non-smooth media by PINN with quadratic neural network and PML condition

Yanqi Wu∗†, Hossein S. Aghamiry‡, Stephane Operto‡, Jianwei Ma∗†

18b912018@stu.hit.edu.cn, aghamiry@geoazur.unice.fr,
operto@geoazur.unice.fr, jwm@pku.edu.cn

∗Department of Mathematics and Center of Geophysics, Harbin Institute of Technology, Harbin 150001, China
†School of Earth and Space Science, Peking University, Beijing 100871, China
‡Geoazur, Université Côte d’Azur, CNRS, IRD, OCA, Valbonne, France.

(September 29, 2022)

Running head: PINN for scattered wave equation

ABSTRACT

Frequency-domain simulation of seismic waves plays an important role in seismic inversion, but it remains challenging in large models. The recently proposed physics-informed neural network (PINN), as an effective deep learning method, has achieved successful applications in solving a wide range of partial differential equations (PDEs), and there is still room for improvement on this front. For example, PINN can lead to inaccurate solutions when PDE coefficients are non-smooth and describe structurally-complex media. In this paper, we solve the acoustic and visco-acoustic scattered-field
wave equation in the frequency domain with PINN instead of the wave equation to remove source singularity. We first illustrate that non-smooth velocity models lead to inaccurate wavefields when no boundary conditions are implemented in the loss function. Then, we add the perfectly matched layer (PML) conditions in the loss function of PINN and design a quadratic neural network to overcome the detrimental effects of non-smooth models in PINN. We show that PML and quadratic neurons improve the results as well as attenuation and discuss the reason for this improvement. We also illustrate that a network trained during a wavefield simulation can be used to pre-train the neural network of another wavefield simulation after PDE-coefficient alteration and improve the convergence speed accordingly. This pre-training strategy should find application in iterative full waveform inversion (FWI) and time-lag target-oriented imaging when the model perturbation between two consecutive iterations or two consecutive experiments can be small.
INTRODUCTION

The numerical simulation of seismic waves is an important topic in seismic exploration since its accuracy can impact wave-equation based imaging results (Virieux et al., 2009, 2011). Moreover, it represents the most computationally-expensive ingredient of Full waveform inversion (FWI), which is a staple for high-resolution imaging of the Earth’s interior in exploration geophysics and earthquake seismology (Virieux et al., 2017; Tromp, 2019). The wave equation can be solved either in time or frequency domains (Plessix, 2017; Kostin et al., 2019). Although explicit time-domain methods have low memory requirements and can be efficiently parallelized over sources, frequency-domain modeling is still attractive because efficient multiscale frequency-domain FWI on long-offset data can be implemented with a few discrete frequencies and attenuation effects of arbitrary complexity (Virieux et al., 2009). However, frequency-domain modeling requires solving a large and sparse linear system per frequency, which is difficult to implement in large models, despite several methods based on direct and iterative solvers have been proposed to address this challenge (Plessix, 2007; Erlangga and Herrmann, 2008; Amestoy et al., 2016).

Recently, deep learning (DL), combining machine learning and deep neural networks (NN), has made significant progress in solving problems with high-dimensional data, and it has proved to be effective at discovering intricate structures from huge, complex data sets (Le Cun et al., 2015). Putting aside the idea of artificial intelligence (AI) and the successful applications in data science, DL has shown a bright prospect for solving partial differential equations (PDEs), especially in high-dimensional settings.
Generally, DL-based PDE solving methods can be divided into two broad categories (Karniadakis et al., 2021):

- Minimizing the difference between network prediction and real data when there are a lot of data and ambiguous physical rules (Han et al., 2018; Long et al., 2018; Long et al., 2019),

- Minimizing the governing-equation errors when there are few data (boundary and initial conditions) and precisely known physical rules (Zhu et al., 2019; Sun et al., 2020).

Physics-informed neural networks (PINNs), seamlessly integrating both the data-based and mathematical model-based terms, has flexibility in informing physical laws described in differential equations and thus achieving successful applications in solving a wide range of PDEs (Cai et al., 2022; Pang et al., 2019; Fang and Zhan, 2019; Zhang et al., 2020). PINNs output the PDE solutions by a NN that is trained to minimize the PDE residuals on the part of points in the certain domain (Raissi et al., 2019). They use automatic differentiation to calculate the partial derivatives in PDEs without the explicit need for mesh generation. Both the forward and inversion problems could be solved in this framework. Since it is discretization-free, PINNs can deal with PDEs in high-dimensional domains, which is difficult in conventional numerical methods.

However, PINN methods can not defeat the traditional numerical methods yet be-
cause they are not accurate or efficient enough for solving standard forward problems. PINN suffers from a slow convergence rate and accuracy degradation for a certain class of PDEs since it recasts the forward problem as the minimization of a loss function for estimating the weights of the NN. Then, the efficiency is limited by the convergence rate during network training because of the high-dimensional non-convex loss function and the limited accuracy induced by the size of the network. Based on the universal approximation theorem, the objective solution should be smooth when a NN is limited in size (Barron, 1993).

Even with those limitations, attracted by the potential of PINN, many researchers have been implementing the method for improvement. A first approach aims at optimizing the NN architecture: Using adaptive activation functions, Jagtap et al. (2020) significantly improved the convergence rate, especially at early training, as well as the solution accuracy of PINN. Gao et al. (2021) proposed a novel physics-constrained CNN learning architecture to learn solutions of parametric PDEs on irregular domains without any labeled data. To deal with PDEs with a non-smooth solution, Kharazmi et al. (2021) proposed a variational formulation of PINN based on the Galerkin method (hp-VPINN) and Mao et al. (2020) studied how the chosen positions of the training samples impact the performances for smooth solutions and Riemann problems.

For wave propagation problems, most of the research focus on time domain wave equation. Both of the forward and inversion problems are studied with PINN. Xu et al. (2019) performed the velocity inversion with PINN in simple inhomogenous media and obtained a better result than FWI. Voytan and Sen (2020) extended PINNs
to inhomogeneous velocity model and implementing absorbing boundary conditions. Karimpouli and Tahmasebi (2020) solved a 1D form seismic wave equation and simultaneously invert the velocity model. Moseley et al. (2020) proved that PINN can model and predict a wide range of physical phenomena by only given the first few timesteps of the solution. Huang et al. (2021) eliminated the source singularity and used a multi-scale deep neural network with periodic activation function to improve the accuracy and convergence speed of the PINNs method. Rasht-Behesht et al. (2022) applied PINNs to seismic inversions within heterogeneous media and demonstrated that PINNs yield excellent results for inversions on all cases considered and with limited computational complexity. For the wave propagation in frequency domain, Song et al. (2021) and Song and Alkhalifah (2022) used PINN to solve the scattered-field (Lippmann-Schwinger) wave equation instead of the total-field wave equation to remove the singularity and complexity of the wavefield near the source. However, sharp contrasts in the PDE coefficients (in this case, velocity contrasts in the underground medium) make the scattered wavefield more complex and non-smooth and hence affect the accuracy of PINN. In this paper, we address the problem of non-smooth velocity models with two recipes. First, we add absorbing perfectly matched layer (PML) conditions (Bérenger, 1994) into PINN to explicitly strengthen the coupling of the two outputs of PINN (in this case, the real and imaginary part of the wavefield) on the boundaries. Second, we use quadratic neurons instead of the traditional linear neurons in the first layer of the NN to increase its nonlinearity and improve the estimation of the complex scattered wavefield. We show the improved wavefield reconstruction induced by these two ingredients with the Marmousi
model. We also solve the scattered-field wave equation in a complex-valued velocity model, such that the two outputs of PINN could be explicitly coupled everywhere in the medium. We first illustrate how non-smooth velocity models deteriorate the accuracy of the wavefields computed with PINN. Then, we show and discuss why PMLs mitigate this issue and contribute to reconstructing more accurate wavefields and why PINN behaves better in a complex-valued velocity model. We also propose a pre-training workflow that mitigates the computational burden of wavefield simulation after model alteration. This pre-training strategy can find interesting applications in iterative FWI and time-lapse imaging to mitigate the computational burden of the forward problem in iterations or during the monitoring stage.

THEORY

The non-smooth behavior of monochromatic wavefields near the point source makes the wave equation challenging to solve with PINN, especially when the network is not excessively large. To bypass this issue, Alkhalifah et al. (2021) proposed to solve the scattered-field wave equation (Lippmann-Schwinger equation) to avoid the singularity at source. In the frequency-domain, this equation is given by

\[
(\omega^2 m(x) + \nabla^2) \delta u(x) = -\omega^2 (m(x) - m_0(x)) u_0(x),
\]

where \(\omega\) is the angular frequency, \(m\) is the squared-slowness model as a function of space coordinates \(x = (x, z)\), \(\nabla^2\) is the Laplace operator, \(\delta u\) is the complex-valued scattered wavefield, \(m_0\) is the background squared-slowness model and \(u_0\) is the background wavefield in \(m_0\). Moreover, \(\delta u = \delta u_r + i\delta u_i\), where \(\delta u_r\) and \(\delta u_i\) are
the real and imaginary parts of the scattered wavefield and $\hat{i} = \sqrt{-1}$. In this study, we use a homogeneous background model, leading to

$$u_0[x] = \frac{i}{4} H_0^{(2)} \left( \sqrt{m_0 \omega} |x - x_s| \right),$$  \hspace{1cm} (2)

where $H_0^{(2)}$ is the zero-order Hankel function of the second kind, $m_0$ is the constant squared slowness and $x_s$ denotes the position of the source (Carcione, 2015). Absorbing boundary conditions are classically used to simulate wave propagation in infinite media with classical numerical methods. The scattered-field wave equation with perfectly matched layer (PML) absorbing boundary conditions (Bérenger, 1994) reads

$$\frac{\partial}{\partial x} \left( e_x[z] \frac{\partial \delta u[x]}{\partial x} \right) + \frac{\partial}{\partial z} \left( e_x[z] \frac{\partial \delta u[x]}{\partial z} \right) + \omega^2 e_x[x] e_z[z] m[x] \delta u[x] =$$

$$- \omega^2 e_x[x] e_z[z] (m[x] - m_0[x]) u_0[x],$$

where $e_x[x] = 1 - \frac{\hat{i} \sigma_x[x]}{\omega}$ and $e_z[z] = 1 - \frac{\hat{i} \sigma_z[z]}{\omega}$ (Chen et al., 2013). The damping function $\sigma_x[x]$ (and similarly for $\sigma_z[z]$) defined as

$$\sigma_x[x] = \begin{cases} 
2\pi a_0 f_0 \left( \frac{l_x[x]}{L_{PML}} \right)^2, & \text{inside PML} \\
0, & \text{outside PML}
\end{cases},$$

where $f_0$ is the peak frequency of the source, $L_{PML}$ is PML thickness, and $l_x[x]$ is the horizontal distance from the point $(x, z)$ in the PML to the PML-medium interface (Chen et al., 2013). Moreover, $a_0$ is a constant set to 1.79 (Zeng et al., 2001).
PINN for the scattered wave equation with PML

Frequency-domain seismic wave modeling (Helmholtz problem) is a boundary-value problem. It is classically tackled by solving a large and sparse indefinite linear system per frequency with direct or iterative methods, which is a challenging task for large-scale problems (Ernst and Gander, 2011; Dolean et al., 2015). The PINN method, which is mesh-free, bypasses this linear-algebra problem by minimizing the following loss function estimated at \( N \) random spatial positions (Alkhalifah et al., 2021):

\[
\begin{align*}
    f[\theta] &= \frac{1}{N} \sum_{j=1}^{N} \left| \omega^2 m^{(j)} \delta u_r^{(j)}[\theta] + \nabla^2 \delta u_r^{(j)}[\theta] + \omega^2 (m - m_0)^{(j)} u_{0r}^{(j)} \right|^2 \\
    &+ \frac{1}{N} \sum_{j=1}^{N} \left| \omega^2 m^{(j)} \delta u_i^{(j)}[\theta] + \nabla^2 \delta u_i^{(j)}[\theta] + \omega^2 (m - m_0)^{(j)} u_{0i}^{(j)} \right|^2,
\end{align*}
\]  (5)

where \( (j) \) denotes the training sample index. The inputs to the network, like a function, are \( N \) spatial locations \( x \), where \( N \) refers to the number of training samples. The output of the network are the real and imaginary values of the scattered wavefield \( (\delta u_r \text{ and } \delta u_i, \text{ respectively}) \), which are functions of the optimization variables \( \theta \), namely the weights of the NN. We decompose the loss function into two terms related to the real and imaginary parts of the governing-equation errors because NN cannot manage complex numbers for now. It is worth noting that the first and the second terms involve only \( \delta u_r \) and \( \delta u_i \), respectively, when \( m \) is real valued and no boundary conditions are explicitly implemented in PINN. This suggests that the loss function of equation 5 may lack some physical constraints considering that, the wavefields being complex valued, \( \delta u_r \) and \( \delta u_i \) are closely related. Indeed, Alkhalifah et al. (2021) and Rasht-Behesht et al. (2021) showed that PINN can theoretically simulate waves in
infinite media from the Helmholtz equation without boundary conditions. The reason we can respectively deal with $\delta u_r$ and $\delta u_i$ in equation 5 without boundary conditions is that their relationship is implicitly informed by $u_0$ and subsurface properties. Here, we propose however to enforce this relationship more explicitly in PINN by implementing the scattered wave equation with PML, equation 3, in the loss function. The loss function $f[\theta]$ with PML reads
\[
f_{PML}[\theta] = \frac{1}{N} \sum_{j=1}^{N} \left( F_r \left[ \delta u_r^{(j)}[\theta], \delta u_r^{(j)}[\theta] \right]^2 + F_i \left[ \delta u_r^{(j)}[\theta], \delta u_i^{(j)}[\theta] \right]^2 \right),
\]
where the real and imaginary parts of the equation errors are
\[
F_r[\delta u_r, \delta u_i] = \frac{\partial}{\partial x} \left( \frac{1 + c^2 l_2^2 l_x^2}{1 + c^2 l_2^4} \frac{\partial \delta u_r}{\partial x} + \frac{c (l_z^2 - l_x^2)}{1 + c^2 l_2^4} \frac{\partial \delta u_i}{\partial x} \right)
+ \frac{\partial}{\partial z} \left( \frac{1 + c^2 l_2^2 l_z^2}{1 + c^2 l_2^4} \frac{\partial \delta u_r}{\partial z} + \frac{c (l_x^2 - l_z^2)}{1 + c^2 l_2^4} \frac{\partial \delta u_i}{\partial z} \right)
+ (1 - c^2 l_2^2 l_x^2) \frac{\omega^2}{L_{PML}} \left( m \delta u_r + (m - m_0) u_{0r} \right)
+ c (l_x^2 + l_z^2) \frac{\omega^2}{L_{PML}} \left( m \delta u_i + (m - m_0) u_{0i} \right),
\]
and
\[
F_i[\delta u_r, \delta u_i] = \frac{\partial}{\partial x} \left( \frac{c (l_z^2 - l_x^2)}{1 + c^2 l_2^4} \frac{\partial \delta u_r}{\partial x} + \frac{1 + c^2 l_2^2 l_x^2}{1 + c^2 l_2^4} \frac{\partial \delta u_i}{\partial x} \right)
+ \frac{\partial}{\partial z} \left( \frac{c (l_x^2 - l_z^2)}{1 + c^2 l_2^4} \frac{\partial \delta u_r}{\partial z} + \frac{1 + c^2 l_2^2 l_z^2}{1 + c^2 l_2^4} \frac{\partial \delta u_i}{\partial z} \right)
- c (l_x^2 + l_z^2) \frac{\omega^2}{L_{PML}} \left( m \delta u_r + (m - m_0) u_{0r} \right)
+ (1 - c^2 l_2^2 l_x^2) \frac{\omega^2}{L_{PML}} \left( m \delta u_i + (m - m_0) u_{0i} \right),
\]
c = \frac{2 \pi a_0 f_0}{\omega L_{PML}} \text{ and } u_{0r}, u_{0i} \text{ are computed with the closed-form expression given in equation 2. One can readily check that the two terms of the loss function } f_{PML} \text{ are now functions of both } \delta u_r \text{ and } \delta u_i \text{ unlike those of equation 5.}

Aside from all the constant coefficients in equation 4, we also need to compute dur-
ing the training the derivative of \( l_x[x] \) and \( l_z[z] \) without relying on mesh. This is performed in \textit{tensorflow} by \textit{Relu} activation function:

\[
  l_x[x] = \max(0, x_{b1} - x) + \max(0, x - x_{b2}) \\
  = \text{Relu}(x_{b1} - x) + \text{Relu}(x - x_{b2}),
\]

where \( x_{b1} \) and \( x_{b2} \) denote the position of the boundaries. The functions \( \sigma_z \) and \( l_z \) can be chosen similarly.

Although PMLs explicitly provide some physical constraints, like the relationship between the real and imaginary parts of the wavefield, it only works on the boundary. In attenuating media, the velocity model is complex-valued, which can couple the real and imaginary parts of wavefield everywhere in space. Moreover, attenuating effects are also necessary to represent the real physics and can be part of optimization variables during the imaging inverse problem (e.g., Operto and Miniussi, 2018). Here, we implement attenuation with the Kolsky-Futterman (KF) model. In this case, the squared slowness reads

\[
  m[x] = \frac{1}{v^2[x]} \left( 1 - \frac{\alpha[x]}{\pi} \ln \frac{\omega}{\omega_r} + \frac{\alpha[x]}{2} \right)^2,
\]

where \( \omega_r \) is a reference angular frequency, \( v \) and \( \alpha \) are the real-valued phase velocity and attenuation factor (inverse of quality factor \( Q \)) at the reference frequency. The complex-squared slowness can be decomposed to the real \( (m_r) \) and imaginary \( (m_i) \) parts as

\[
  m[x] = m_r[x] + \hat{i}m_i[x],
\]
where
\[
m_r[x] = \frac{1}{v^2[x]} \left( 1 - \frac{\alpha[x]}{\pi} \ln \left| \frac{\omega}{\omega_r} \right| \right)^2 - \frac{\alpha^2[x]}{4v^2[x]},
\]
(12)
\[
m_i[x] = \frac{\alpha[x]}{v^2[x]} \left( 1 - \frac{\alpha[x]}{\pi} \ln \left| \frac{\omega}{\omega_r} \right| \right).
\]

Then, the real and imaginary parts of the equation errors with complex-valued slowness are:
\[
F_r[\delta u_r, \delta u_i] = \frac{\partial}{\partial x} \left( \frac{1 + c^2 l_x^2 l_z^2}{1 + c^2 l_x^2} \frac{\partial \delta u_r}{\partial x} + \frac{c (l_z^2 - l_x^2)}{1 + c^2 l_x^2} \frac{\partial \delta u_i}{\partial x} \right)
+ \frac{\partial}{\partial z} \left( \frac{1 + c^2 l_x^2 l_z^2}{1 + c^2 l_z^2} \frac{\partial \delta u_r}{\partial z} + \frac{c (l_z^2 - l_x^2)}{1 + c^2 l_z^2} \frac{\partial \delta u_i}{\partial z} \right)
+ (1 - c^2 l_x^2 l_z^2) \omega^2 (m_r \delta u_r - m_i \delta u_i + (m_r - m_{0r}) u_{0r} - (m_i - m_{0i}) u_{0i})
+ c (l_x^2 + l_z^2) \omega^2 (m_r \delta u_i + m_i \delta u_r + (m_r - m_{0r}) u_{0i} + (m_i - m_{0i}) u_{0r}),
\]
(13)

and
\[
F_i[\delta u_r, \delta u_i] = \frac{\partial}{\partial x} \left( \frac{c (l_z^2 - l_x^2)}{1 + c^2 l_x^2} \frac{\partial \delta u_r}{\partial x} + \frac{1 + c^2 l_x^2 l_z^2}{1 + c^2 l_x^2} \frac{\partial \delta u_i}{\partial x} \right)
+ \frac{\partial}{\partial z} \left( \frac{c (l_z^2 - l_x^2)}{1 + c^2 l_z^2} \frac{\partial \delta u_r}{\partial z} + \frac{1 + c^2 l_x^2 l_z^2}{1 + c^2 l_z^2} \frac{\partial \delta u_i}{\partial z} \right)
- c (l_z^2 + l_x^2) \omega^2 (m_r \delta u_r - m_i \delta u_i + (m_r - m_{0r}) u_{0r} - (m_i - m_{0i}) u_{0i})
+ (1 - c^2 l_x^2 l_z^2) \omega^2 (m_r \delta u_i + m_i \delta u_r + (m_r - m_{0r}) u_{0i} + (m_i - m_{0i}) u_{0r}),
\]
(14)

where \(m_{0r}\) is the real part and \(m_{0i}\) is the imaginary part of the background squared-slowness model.

**Quadratic neural network**

Although replacing the full wavefield with the scattered wavefield can address the source singularity problem, the scattered wavefield still contains the rough compo-
nents of the total field. This may be one of the reasons why PINN reconstructed a smoother solution than the exact one (Alkhalifah et al., 2021). We designed the NN with eight hidden layers with 80 neurons in every layer. The activation function between layers, other than the last layer, is a hyperbolic tangent in all the examples. The last hidden layer connected to the output layer is linear. Adam optimizer is chosen to train the NN. Increasing the size of the NN should lead to a more accurate reconstruction, albeit with an increased computational cost. Here, we propose using quadratic neurons instead of the traditional linear neuron at the first layer of NN to increase the nonlinearity of the NN and improve the estimation of the scattered wavefield accordingly. Let us take a look at one neuron of the deep NN. Usually, the output $x^{(1)}$ of a neuron in the first layer is given by:

$$x^{(1)} = \tanh \left( xw_1 + z w_2 + b \right), \quad (15)$$

where $w_1$, $w_2$ and $b$ are parameters to be optimized, the activation function is hyperbolic tangent and $x$ and $z$ are the inputs of the network, in this case, the space coordinates of the wavefield. As shown in Figure 1, the improved formulation is given by:

$$x^{(1)} = \tanh \left( xw_1 + zw_2 + xzw_3 + x^2 w_4 + z^2 w_5 + b \right). \quad (16)$$

From the NN perspective, it increases the non-linearity and flexibility by only adding a few parameters leading to a negligible computational overhead. From a physical standpoint, wavefield is closely related to the distance from the current point to the source, which is easier to describe with the quadratic of space coordinate components.
NUMERICAL MODELING

We first solve the frequency-domain scattered-field wave equation in a smoothed Marmousi model for $f = 5Hz$ frequency (Figure 2a). We assess the accuracy of the wavefield reconstructed by PINN against the wavefield computed with a finite-difference frequency-domain (FDFD) method with adaptive FD coefficients (Chen et al., 2013). Five thousand training points are randomly chosen from a $100 \times 100$ Cartesian grid in every epoch. The grid intervals in $x$ and $z$ are $40m$ and $20m$, respectively. The source is at the middle of the surface. It is reminded that the NN is formed by eight hidden layers with 80 neurons in every layer. The activation function between layers, other than the last layer, is an hyperbolic tangent in all the examples. The last hidden layer connected to the output layer is linear. We train the network with Adam optimizer and 150000 epochs on NVIDIA Tesla V100 32 GB GPU. The scattered wavefield computed with PINN when the loss function doesn’t contain PML condition are shown in Figures 2b-2d. These results suggest that PINN manages to solve the scattered-field wave equation in the smoothed velocity model with acceptable accuracy, although some mild differences with the FDFD wavefield are shown. However, if we replace the smoothed Marmousi model by the original Marmousi model (Figure 3a), the wavefield reconstructed by PINN is highly inaccurate (Figures 3c and 3d). Moreover, this inaccurate reconstruction is barely improved when increasing the size of the NN. In fact, we have tested several NNs, e.g., 6 and 10 hidden layers (depth) with 80 neurons in every layer, eight hidden layers with 80 and 60 neurons in every layer, or a network given by $128, 128, 64, 64, 32, 32, 16, 16, 8, 8$ neurons in
every hidden layer from input to output without significant improvements relative to the results of Figures 2c and 2d.

**PINN with PML**

Since the network design barely changed the solution, a reasonable statement is that inaccurate wavefield reconstruction not only results from the network but also from the loss function. Considering equation 5, we notice that the real and imaginary parts of the scattered wavefield are calculated to some extent separately by minimization of the two terms of the loss function while they are closely physically entwined. In the FDFD method, the wavefield is obtained by computing the inverse of a large and sparse matrix $A$ with PML absorbing conditions. Thus, the real and imaginary parts of the wavefield are both closely dependent on the structure of $A$ and the boundary conditions. However, in equation 5, $\delta u_r[\theta]$ and $\delta u_i[\theta]$ depend on the parameters of the NN, and the relationship between the real and imaginary part is only implicitly informed to the NN through the background wavefield $u_{0r}$ and $u_{0i}$ in the right-hand side of the scattered field equation. Moreover, the non-smoothness of the velocity model will also increase the difficulty of learning the coupled relationship between the real and imaginary parts of the scattered field since the sharp contrasts of the medium are transferred in the scattering (or contrast) source. Accordingly, we repeat the simulation in the non-smooth Marmousi model using the loss function with PML, equation 6, to explicitly enforce physical constraints on the boundaries and couple the real and imaginary parts of the scattered wavefield in the two terms of the loss
function. Although PINN may automatically satisfy the wave equation in infinite media without implementing boundary conditions, a PML condition should not harm it. More insights about the effects of the PML during the training performed by PINN are illustrated in the final Discussion section.

The reconstructed wavefield (Figures 4a and 4b) shows how adding PML condition in the loss function significantly improves the wavefield reconstruction compared to Figures 3c and 3d. However, compared to the ground truth shown in 3b, the wavefield in Figure 4a remains not accurate enough.

**PINN with PML and quadratic neural network**

According to the wave equation, a non-smooth velocity model will cause a non-smooth (or discontinuous) Laplacian of the wavefield. However, the conventional NN with hyperbolic tangent as activation function is differentiable in any order. We may improve the NN by increasing its size, but this will generate significant computational overhead. Instead, we propose to use quadratic neurons instead of linear neurons on the first layer. Figures 4c and 4d show the wavefield reconstructed by PINN + PML + quadratic NN. Compared to PINN + PML shown in Figures 4a and 4b, the wavefield reconstruction is improved with quadratic NN, especially near the surface. These more accurate wavefield reconstructions are indeed critical for future FWI applications.
PINN in attenuating medium

Implementing intrinsic attenuation in the underground medium makes the squared-slowness model complex-valued. In addition to better representing the physics of wave propagation into the Earth, attenuation couples the real and imaginary part of the wavefield at every spatial position in each term of the loss function, equations 13 and 14, hence strengthening physical constraints in PINN. We apply attenuation with the homogeneous quality factor of $Q = 15$ and repeat the wavefield reconstruction using PINN (Figures 5c), PINN + PML (Figures 6a) and PINN + PML + quadratic NN (Figures 6c). After considering the attenuation, the wavefield reconstructed by PINN (Figure 5c) is still inaccurate, although it is slightly improved compared to the wavefields (Figure 3c) computed in the non-attenuating medium. Moreover, the visco-acoustic wavefields computed with PINN + PML (Figure 6a) and PINN + PML + quadratic NN (Figure 6c) contain more details than the corresponding wavefields computed without attenuation (Figure 4a and 4c) although the diffusive effects of attenuation, hence suggesting a more accurate wavefield reconstruction, especially in the shallow area.

To relate more quantitatively the impact of modeling errors on velocity reconstruction, we take advantage of the bi-linearity of the wave equation \cite{Aghamiry2019} to reconstruct the velocity model, as shown in Figures 7, from the simulated monochromatic wavefields. The bilinearity of the wave equation implies that the squared slowness model can be inferred exactly from a monochromatic wavefield by the following pointwise division:

$$m[x] = \frac{b[x, \omega] - \nabla^2 u[x, \omega]}{\omega^2 u[x, \omega]}, \quad (17)$$
where $b[x, \omega]$ is the physical source of the Helmholtz (total-field) equation.

Without PML conditions and quadratic NN, the PINN wavefield can hardly recover the correct velocity model no matter in acoustic (Figures 7a) or visco-acoustic (Figures 7b) medium. Adding PML significantly improves the velocity reconstruction, but the velocity contrasts are blurred in acoustic (Figures 7c) and visco-acoustic (Figures 7d) medium. Using jointly quadratic NN and PML further improves the velocity reconstruction, in particular in the deep part at the reservoir level in acoustic (Figures 7e) and visco-acoustic (Figures 7f) medium, despite some artifacts in the shallow region remaining in Figure 7e. More insights about how quadratic NN improves the results of PINN are illustrated in the final discussion section. Compared to the velocity models of Figures 7c and 7e (simulation without attenuation), those of Figures 7d and 7f (simulation with attenuation) show an improvement in the most contrasted area of the model and near the surface. Compared to Figure 7e, the artifacts in the shallow region are also mitigated in Figure 7f. This further illustrates how the reconstructed wavefields are improved by PML, quadratic NN and attenuation effects.

Figure 8 shows the evolution of the loss functions with iterations for the above six cases. Although the PML condition can improve the performance of PINN, it somehow causes a slow convergence rate in both the attenuation and non-attenuation cases, especially during the early iterations. Combining PML with quadratic NN can significantly accelerate the convergence rate.
DISCUSSION

How do non-smooth velocities affect the results?

To suggest an answer to the above question, let’s come back to equation 1. In the PINN method, $\delta u$ is the output of the NN and the solution we are looking for. All the other parameters are known. Therefore, the left-hand side term of equation 1 contains network parameters, unlike the right-hand side term. Let us say that the left-hand side term represents some operations on NN (multiplication and derivation) and the right-hand side term represents the label of the NN after operations by some abuse of language. It should be noted that both the right- and left-hand side terms contain the model $m$. Considering the poor performance of PINN in non-smooth $m$, we should wonder which of the two terms (operations versus labels) drives PINN to a wrong solution? To answer this question, we consider the following two equations:

\[
(\omega^2 m[x] + \nabla^2) \delta u[x] = -\omega^2 (m_s[x] - m_0[x]) u_0[x], \tag{18}
\]

\[
(\omega^2 m_s[x] + \nabla^2) \delta u[x] = -\omega^2 (m[x] - m_0[x]) u_0[x], \tag{19}
\]

where $m$ is non-smooth Marmousi model as shown in Figure 3a and $m_s$ is smooth Marmousi model as shown in Figure 2a. Figures 9c and 9d show the estimated solutions of equations 18 and 19 by PINN without using PML conditions or quadratic neurons. Apart from the velocity model, all the parameters are the same as those used in the example of Figure 2. The left-hand side term in equations 18 and 19 contains the direct operation on the output of NN, and the right-hand side terms do not change during training, like a label. Based on the universal approximation theorem, if the
label or the objective solution is not smooth enough along the input space, we can improve the performance of NN by increasing its size. However, Figure 9 shows that it is the non-smooth velocity in the operation term instead of the label term that causes the poor performance of PINN. This may explain why we can not improve the PINN results by simply increasing the size of the NN. Further research about the operation term in PINN will be necessary during future work.

**How do PMLs improve the accuracy of PINN?**

Let’s consider the total-field wave equation:

$$
(\omega^2 m[x] + \nabla^2) u[\omega, x] = b[\omega, x].
$$

(20)

where $u$ is the full wavefield and $b$ is the physical source. To mitigate the singularity of the point source $b$, we implement the real part of $b$ with a two-dimensional Gaussian function and the imaginary part is set to zero. The velocity model is homogeneous and $m = 1/(1.5)^2 s^2/km^2$ and the frequency is set to $3Hz$. The training samples are the points on 100 by 100 mesh with a grid interval of $h = 20m$. The NN is trained with 50000 epochs. Figures 10 show the estimated solutions by PINN with and without PML conditions. Without PML conditions, the real part of the wavefield is inaccurately reconstructed while the imaginary part remains close to zero. As it is mentioned before, when solving the frequency-domain wave equation or the scattered-field equation with PINN, the real and imaginary parts of the wavefield are calculated separately when PML conditions are not used. Thus, the connection between the real and imaginary part of the wavefield is not informed in the loss.
function of PINN. In other words, the physical constraints are not enough to find the correct solutions. After adding PMLs, the PINN method can successfully reconstruct the solution of the wave equation. Figure 11 shows the convergence procedure during training and the role of the PMLs. The NN first reconstructs the real part of the wavefield approximately, proceeding from the source to the boundaries while the imaginary part remains close to zero. Once the real part of the wavefield reaches the boundary, the NN starts updating the imaginary part of the wavefield from the boundary to the source through the constraint provided by the PML conditions while refining the real part of the wavefield accordingly.

**How does quadratic neuron improve the neural network?**

As mentioned before, the non-smooth velocity model causes a non-smooth (or discontinuous) Laplacian of wavefield (i.e., $\nabla^2 \delta u$), which is difficult to approximate with PINN. There are two possible reasons: the network size is not large enough, or the network is not sensitive to the change of Laplacian during training. In our opinion, the network in this study has sufficient size. Instead, we believe that the quadratic NN improves the sensitivity of PINN to the change of Laplacian. To support this statement, we check in more detail the accuracy with which the Laplacian term is reconstructed by PINN. Figures 12a and 12b show the real part of the scattered wavefield and the Laplacian, respectively, in the non-attenuating Marmousi model along a vertical profile located at $X=1200m$. A direct comparison between the solutions computed with FDFD, PINN, PINN + PML and PINN + PML + quadratic NN is
shown. Figures 12c and 12d show the same information for the attenuating Mar-
mousi model. The wavefields from quadratic NN in both Figures 12a and 12c have
better results than those without using quadratic NN. Compared to the real part of
the scattered wavefields, non-smooth patterns are more obvious in the Laplacian of
wavefields. This non-smooth behavior, as shown in Figures 12b and 12d, is clearly
better matched when quadratic NN is used. Moreover, the computational overhead
generated by the quadratic NN is negligible since we add only three weights in the
neurons of the first layer.

Using pretraining for efficient wavefield simulation after model
alteration

Although PINN has shown its potential to solve the wave equation, there are still
some remaining issues to solve. First, PINN recasts the forward problem as the min-
imization of a loss function for estimating the weights of the NN, which makes the
computational cost to be a crucial issue. This issue can be greatly mitigated by pre-
training. For example, we can firstly train a NN on a simple and smooth attenuating
velocity model, e.g., a model where velocity linearly increases with depth as illus-
trated in Figure 13a. Due to the smoothness of the velocity model, the reconstructed
wavefield by PINN + PML + quadratic NN (Figure 13d) matches reasonably well
with the FDFD wavefield (Figure 13b) after 150000 epochs. Moreover, the differ-
ences between the two wavefields suggest that they mostly result from the different
behavior of the PMLs in the FDFD and NN approaches, as shown in Figure 13e. The
It is worth mentioning that this process could be performed once and for all before performing a specific simulation. Then, we switch from the linear velocity model to a smooth version of the Marmousi model, as shown in Figure 14a, and train the pre-trained NN of the previous simulation (based on the linear velocity model) with the smooth Marmousi model. Figures 14d and 14g show the reconstructed wavefield by PINN + PML + quadratic NN after 5000 and 20000 epochs, respectively. It takes 7 minutes for every 5000 epochs. A good agreement between the FDFD and the PINN wavefields is shown, although the number of epochs was reduced by around one order of magnitude compared to the first simulation. We continue this pre-training workflow by changing one more time velocity model by considering a sharper version of the Marmousi model as shown in Figure 15a. We train the previous pre-trained NN of Figure 14d with 5000 epochs to reconstruct the wavefields of Figure 15d and train the NN of Figure 14g with 20000 epochs to reconstruct wavefields of Figure 15g. Again, they (Figure 15d and 15g) match fairly well with the FDFD wavefield (Figure 15b), although the limited number of epochs involved in the PINN simulation.

The right columns of Figure 13, 14 and 15 show the recovered velocity model from the lift wavefield of FDFD and PINNs. We conclude that we do not need to train a new NN when we perform a wavefield simulation after model alteration. Instead, we can train the NN that has been pre-trained during the simulation in the unaltered medium. In our case, this pre-training decreases the iteration count by around one order of magnitude and the more similar the models are, the less iterations the simulation takes. This pre-training strategy should find very interesting applications in FWI when the model perturbations become increasingly small as the FWI approaches
the convergence point. Another application is time-lapse and target-oriented imaging, where the model alteration in the targeted area is generally very small.

Although PINN may show little advantage in solving simple forward problems (e.g., 2D, single source and single frequency), PINN may be beneficial for solving large-scale 3D forward problems involving multiple sources and frequencies. In this framework, the pre-training strategy introduced above may find applications to efficiently process multiple frequencies and sources in FWI applications. This will be investigated in future studies.

We have shown that the nonlinearity of Laplacian of the NN outputs with respect to the inputs has a significant effect on the performance of PINN. Since the quadratic neurons can improve the accuracy and nonlinearity of the Laplacian, a further study for designing a better NN architecture based on the characteristic of the Laplacian or derivative should be another interesting perspective.

**CONCLUSION**

We solve the frequency-domain scattered-field wave equation for non-smooth velocity models with PINN. We show numerically that the poor performance in contrasted media mainly results from the lack of physical constraint tying together the real and imaginary parts of the wavefields rather than from the size of the NN. This statement was further validated by tracking the history of reconstructing the real and imaginary parts of the wavefield over epochs. We mitigate this issue by explicitly implementing PML conditions in the loss function of PINN. We also use quadratic
neurons in the first layer of the NN rather than linear neurons to simulate more efficiently the discontinuities in the Laplacian of the scattered wavefields. Adding physical attenuation is another leverage to couple the real and imaginary parts of the wavefields everywhere in the medium through complex wave speeds. We show that pre-training can significantly mitigate the computational cost of PINN after model alteration. This opens the door to FWI and target-oriented imaging applications where the cost of the forward problem can be decreased as the inversion approaches the convergence point. Although the accuracy of the wavefields computed with PINN does not reach yet that of classical numerical methods, it may be sufficient to start investigating FWI applications with PINN.

ACKNOWLEDGMENTS

We thank Arash Rezaei who helps to transferring the Matlab codes of FDFD to Python. This work was (partially) supported by Beijing Natural Science Foundation No. Z210001, the WIND consortium (www.geoazur.fr/WIND), and the China Scholarship Council. This work was supported by the French government, through the UCAJEDI Investments in the Future project managed by the National Research Agency (ANR) under reference number ANR-15-IDEX-01. The authors are grateful to the OPAL infrastructure from Université Côte d’Azur and the Université Côte d’Azur’s Center for High-Performance Computing for providing resources and support.
REFERENCES

Aghamiry, H., A. Gholami, and S. Operto, 2019, Improving full-waveform inversion by wavefield reconstruction with alternating direction method of multipliers: Geophysics, 84(1), R139–R162.

Alkhalifah, T., C. Song, U. bin Waheed, and Q. Hao, 2021, Wavefield solutions from machine learned functions constrained by the helmholtz equation: Artificial Intelligence in Geosciences, 2, 11–19.

Amestoy, P., R. Brossier, A. Buttari, J.-Y. L’Excellent, T. Mary, L. Métivier, A. Miniussi, and S. Operto, 2016, Fast 3D frequency-domain FWI with a parallel Block Low-Rank multifrontal direct solver: application to OBC data from the North Sea: Geophysics, 81, R363 – R383.

Barron, A. R., 1993, Universal approximation bounds for superpositions of a sigmoidal function: IEEE Transactions on Information Theory, 39, 930–945.

Bérenger, J.-P., 1994, A perfectly matched layer for absorption of electromagnetic waves: Journal of Computational Physics, 114, 185–200.

Berg, J., and K. Nyström, 2019, Data-driven discovery of pdes in complex datasets: Journal of Computational Physics, 384, 239–252.

Burger, M., L. Ruthotto, S. Osher, et al., 2021, Connections between deep learning and partial differential equations: European Journal of Applied Mathematics, 32, 395–396.

Cai, S., Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, 2022, Physics-informed neural networks (pinns) for fluid mechanics: A review: Acta Mechanica Sinica,
Carcione, J. M., 2015, Wave fields in real media, wave propagation in anisotropic, anelastic, porous and electromagnetic media, third edition ed.: Elsevier.

Chen, Z., D. Cheng, W. Feng, and T. Wu, 2013, An optimal 9-point finite difference scheme for the Helmholtz equation with PML: International Journal of Numerical Analysis & Modeling, 10.

Dolean, V., P. Jolivet, and F. Nataf, 2015, An introduction to domain decomposition methods - algorithms, theory, and parallel implementation: SIAM.

Erlangga, Y. A., and F. J. Herrmann, 2008, An iterative multilevel method for computing wavefields in frequency-domain seismic inversion: SEG Technical Program Expanded Abstracts, 27, 1956–1960.

Ernst, O. G., and M. J. Gander, 2011, Why it is difficult to solve helmholtz problems with classical iterative methods: volume 83 of Numerical Analysis of Multiscale Problems.

Fang, Z., and J. Zhan, 2019, A physics-informed neural network framework for pdes on 3d surfaces: time independent problems: IEEE Access, 8, 26328–26335.

Gao, H., L. Sun, and J.-X. Wang, 2021, Phygeonet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain: Journal of Computational Physics, 428, 110079.

Han, J., A. Jentzen, and W. E, 2018, Solving high-dimensional partial differential equations using deep learning: Proceedings of the National Academy of Sciences, 115, 8505–8510.

Huang, X., H. Liu, B. Shi, Z. Wang, K. Yang, Y. Li, B. Weng, M. Wang, H. Chu, J.
Zhou, et al., 2021, Solving partial differential equations with point source based on physics-informed neural networks: arXiv preprint arXiv:2111.01394.

Jagtap, A. D., K. Kawaguchi, and G. E. Karniadakis, 2020, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks: Journal of Computational Physics, 404, 109136.

Karimpouli, S., and P. Tahmasebi, 2020, Physics informed machine learning: Seismic wave equation: Geoscience Frontiers, 11, 1993–2001.

Karniadakis, G. E., I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, 2021, Physics-informed machine learning: Nature Reviews Physics, 3, 422–440.

Kharazmi, E., Z. Zhang, and G. E. Karniadakis, 2021, hp-vpinns: Variational physics-informed neural networks with domain decomposition: Computer Methods in Applied Mechanics and Engineering, 374, 113547.

Kostin, V., S. Solovyev, A. Bakulin, and M. Dmitriev, 2019, Direct frequency-domain 3D acoustic solver with intermediate data compression benchmarked against time-domain modeling for full-waveform inversion applications: Geophysics, 84(4), T207–T219.

Le Cun, Y., Y. Bengio, and G. Hinton, 2015, Deep learning: Nature, 521, 436–444.

Long, Z., Y. Lu, and B. Dong, 2019, Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network: Journal of Computational Physics, 399, 108925.

Long, Z., Y. Lu, X. Ma, and B. Dong, 2018, Pde-net: Learning pdes from data: International Conference on Machine Learning, PMLR, 3208–3216.

Mao, Z., A. D. Jagtap, and G. E. Karniadakis, 2020, Physics-informed neural net-
works for high-speed flows: Computer Methods in Applied Mechanics and Engineering, 360, 112789.

Moseley, B., A. Markham, and T. Nissen-Meyer, 2020, Solving the wave equation with physics-informed deep learning: arXiv preprint arXiv:2006.11894.

Operto, S., and A. Miniussi, 2018, On the role of density and attenuation in 3D multi-parameter visco-acoustic VTI frequency-domain FWI: an OBC case study from the North Sea: Geophysical Journal International, 213, 2037–2059.

Pang, G., L. Lu, and G. E. Karniadakis, 2019, fpinns: Fractional physics-informed neural networks: SIAM Journal on Scientific Computing, 41, A2603–A2626.

Plessix, R. E., 2007, A Helmholtz iterative solver for 3D seismic-imaging problems: Geophysics, 72, SM185–SM194.

——–, 2017, Some computational aspects of the time and frequency domain formulations of seismic waveform inversion, in Modern solvers for Helmholtz problems, Geosystems Mathematics: Springer, 159–187.

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

Rasht-Behesht, M., C. Huber, K. Shukla, and G. E. Karniadakis, 2021, Physics-informed neural networks (pinns) for wave propagation and full waveform inversions: arXiv preprint arXiv:2108.12035.

——–, 2022, Physics-informed neural networks (pinns) for wave propagation and full waveform inversions: Journal of Geophysical Research: Solid Earth, 127,
Sirignano, J., and K. Spiliopoulos, 2018, Dgm: A deep learning algorithm for solving partial differential equations: Journal of Computational Physics, 375, 1339–1364.

Song, C., T. Alkhalifah, and U. B. Waheed, 2021, Solving the frequency-domain acoustic vti wave equation using physics-informed neural networks: Geophysical Journal International, 225, 846–859.

Song, C., and T. A. Alkhalifah, 2022, Wavefield reconstruction inversion via physics-informed neural networks: IEEE Transactions on Geoscience and Remote Sensing, 60.

Sun, L., H. Gao, S. Pan, and J.-X. Wang, 2020, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data: Computer Methods in Applied Mechanics and Engineering, 361, 112732.

Tromp, J., 2019, Seismic wavefield imaging of Earth’s interior across scales: Nature Reviews Earth & Environment, 1, 40–53.

Virieux, J., A. Asnaashari, R. Brossier, L. Métivier, A. Ribodetti, and W. Zhou, 2017, An introduction to Full Waveform Inversion, in Encyclopedia of Exploration Geophysics: Society of Exploration Geophysics, R1–1–R1–40.

Virieux, J., H. Calandra, and R. E. Plessix, 2011, A review of the spectral, pseudospectral, finite-difference and finite-element modelling techniques for geophysical imaging: Geophysical Prospecting, 59, 794–813.

Virieux, J., S. Operto, H. Ben Hadj Ali, R. Brossier, V. Etienne, F. Sourbier, L. Giraud, and A. Haidar, 2009, Seismic wave modeling for seismic imaging: The Leading Edge, 28, 538–544.
Voytan, D., and M. K. Sen, 2020, Wave propagation with physics informed neural networks: Presented at the SEG International Exposition and Annual Meeting, OnePetro.

Xu, Y., J. Li, and X. Chen, 2019, Physics informed neural networks for velocity inversion: Presented at the SEG International Exposition and Annual Meeting, OnePetro.

Zeng, Y., J. He, and Q. Liu, 2001, The application of the perfectly matched layer in numerical modeling of wave propagation in poroelastic media: Geophysics, 66, 1258–1266.

Zhang, D., L. Guo, and G. E. Karniadakis, 2020, Learning in modal space: Solving time-dependent stochastic pdes using physics-informed neural networks: SIAM Journal on Scientific Computing, 42, A639–A665.

Zhu, Y., N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris, 2019, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data: Journal of Computational Physics, 394, 56–81.
LIST OF FIGURES

1 Illustration of quadratic neural network.

2 The simulation results in smoothed Marmousi model. (a) Smoothed Marmousi model. (b) FDFD scattered fields (real part). (c) PINN scattered fields (real part). (d) Difference between Figure 2b and Figure 2c.

3 The simulation results in non-smoothed Marmousi model. (a) Non-smoothed Marmousi model. (b) FDFD scattered fields (real part). (c) PINN scattered fields (real part). (d) Difference between Figure 3b and Figure 3c.

4 The simulation results in non-smoothed Marmousi model with PML and quadratic NN. (a) PINN scattered field with PML (real part). (b) Difference between Figure 3b and Figure 4a. (c) PINN scattered field with PML and quadratic NN (real part). (d) Difference between Figure 3b and Figure 4c.

5 The simulation results in attenuating Marmousi model with homogeneous quality factor Q=15. (a) Non-smoothed Marmousi model. (b) FDFD scattered fields (real part). (c) PINN scattered fields (real part). (d) Difference between Figure 5b and Figure 5c.

6 The simulation results in attenuating Marmousi model with PML, quadratic NN and Q=15. (a) PINN scattered field with PML (real part). (b) Difference between Figure 5b and Figure 6a. (c) PINN scattered field with PML and quadratic NN (real part). (d) Difference between Figure 5b and Figure 6c.

7 Recovered velocity model computed with the reconstructed wavefields by different approaches. (a) PINN with Q = ∞, (b) PINN with Q = 15, (c) PINN + PML
with Q = ∞, (d) PINN + PML with Q = 15, (e) PINN + PML + quadratic NN with Q = ∞, (f) PINN + PML + quadratic NN with Q = 15.

8 The PINNs losses of different approaches during training.

9 Real parts of solutions from equations 18 and 19. (a) FDFD solution of equations 18, (b) FDFD solution of equations 19, (c) PINN solution of equations 18, (d) PINN solution of equations 19, (e) Difference between Figures 9a and 9c, (f) Difference between Figures 9b and 9d.

10 Solutions of full wave equation by PINN and PINN + PML. (a) Real part by PINN, (b) imaginary part by PINN, (c) real part by PINN + PML, (d) imaginary part by PINN + PML.

11 The training process during solving wave equation by PINN with PML. Left column (a,c,e,g) is real parts after 8000, 12000, 14000, and 20000 epochs. Right column (b,d,f,h) is imaginary parts after 8000, 12000, 14000, and 20000 epochs.

12 The vertical profiles of real part and Laplacian of scattered wavefields in Marmousi model at X=1200m. (a) Real part with Q = ∞, (b) Laplacian with Q = ∞, (c) real part with Q = 15, (b) Laplacian with Q = 15,

13 The 1-D gradient velocity model and corresponding scattered wavefield with Q=15. (a) The true velocity model, (b) real part of wavefields by FDFD, (c) recovered velocity models from Figure 13b, (d) real part of wavefields by PINN + PML + quadratic NN, (e) difference between Figures 13b and 13d, (f) recovered velocity models from Figure 13d.

14 The highly smoothed Marmousi model and corresponding scattered wavefield with Q=15. (a) The true velocity model, (b) real part of wavefield by FDFD,
(c) recovered velocity models from Figures 14b, (d) real part of wavefield by PINN + PML + quadratic NN using 5000 iterations after Figure 13d, (e) difference between Figures 14b and 14d, (f) the recovered velocity models from Figure 14d, (g-i) the same as (d-f), but using 20000 iterations after Figure 13d.

15 The slightly smoothed Marmousi model and corresponding scattered wavefield with Q=15. (a) The true velocity model, (b) real part of wavefield by FDFD, (c) recovered velocity models from Figures 15b, (d) real part of wavefield by PINN + PML + quadratic NN using 5000 iterations after Figure 14d, (e) difference between Figures 15b and 15d, (f) the recovered velocity models from Figure 15d, (g-i) the same as (d-f), but using 20000 iterations after Figure 14g.
Figure 1: Illustration of quadratic neural network.
Figure 2: The simulation results in smoothed Marmousi model. (a) Smoothed Marmousi model. (b) FDFD scattered fields (real part). (c) PINN scattered fields (real part). (d) Difference between Figure 2b and Figure 2c.
Figure 3: The simulation results in non-smoothed Marmousi model. (a) Non-smoothed Marmousi model. (b) FDFD scattered fields (real part). (c) PINN scattered fields (real part). (d) Difference between Figure 3b and Figure 3c.
Figure 4: The simulation results in non-smoothed Marmousi model with PML and quadratic NN. (a) PINN scattered field with PML (real part). (b) Difference between Figure 3b and Figure 4a. (c) PINN scattered field with PML and quadratic NN (real part). (d) Difference between Figure 3b and Figure 4c.
Figure 5: The simulation results in attenuating Marmousi model with homogeneous quality factor $Q=15$. (a) Non-smoothed Marmousi model. (b) FDFD scattered fields (real part). (c) PINN scattered fields (real part). (d) Difference between Figure 5b and Figure 5c.
Figure 6: The simulation results in attenuating Marmousi model with PML, quadratic
NN and Q=15. (a) PINN scattered field with PML (real part). (b) Difference between
Figure 5b and Figure 6a. (c) PINN scattered field with PML and quadratic NN (real
part). (d) Difference between Figure 5b and Figure 6c.
Figure 7: Recovered velocity model computed with the reconstructed wavefields by different approaches. (a) PINN with $Q = \infty$, (b) PINN with $Q = 15$, (c) PINN + PML with $Q = \infty$, (d) PINN + PML with $Q = 15$, (e) PINN + PML + quadratic NN with $Q = \infty$, (f) PINN + PML + quadratic NN with $Q = 15$. 

41
Figure 8: The PINNs losses of different approaches during training.
Figure 9: Real parts of solutions from equations 18 and 19. (a) FDFD solution of equations 18, (b) FDFD solution of equations 19, (c) PINN solution of equations 18, (d) PINN solution of equations 19, (e) Difference between Figures 9a and 9c, (f) Difference between Figures 9b and 9d.
Figure 10: Solutions of full wave equation by PINN and PINN + PML. (a) Real part by PINN, (b) imaginary part by PINN, (c) real part by PINN + PML, (d) imaginary part by PINN + PML.
Figure 11: The training process during solving wave equation by PINN with PML. Left column (a,c,e,g) is real parts after 8000, 12000, 14000, and 20000 epochs. Right column (b,d,f,h) is imaginary parts after 8000, 12000, 14000, and 20000 epochs.
Figure 12: The vertical profiles of real part and Laplacian of scattered wavefields in Marmousi model at X=1200m. (a) Real part with Q = ∞, (b) Laplacian with Q = ∞, (c) real part with Q = 15, (b) Laplacian with Q = 15,
Figure 13: The 1-D gradient velocity model and corresponding scattered wavefield with $Q=15$. (a) The true velocity model, (b) real part of wavefields by FDFD, (c) recovered velocity models from Figure 13b, (d) real part of wavefields by PINN + PML + quadratic NN, (e) difference between Figures 13b and 13d, (f) recovered velocity models from Figure 13d.
Figure 14: The highly smoothed Marmousi model and corresponding scattered wavefield with $Q=15$. (a) The true velocity model, (b) real part of wavefield by FDFD, (c) recovered velocity models from Figures 14b, (d) real part of wavefield by PINN + PML + quadratic NN using 5000 iterations after Figure 13d, (e) difference between Figures 14b and 14d, (f) the recovered velocity models from Figure 14d, (g-i) the same as (d-f), but using 20000 iterations after Figure 13d.
Figure 15: The slightly smoothed Marmousi model and corresponding scattered wavefield with $Q=15$. (a) The true velocity model, (b) real part of wavefield by FDFD, (c) recovered velocity models from Figures 15b, (d) real part of wavefield by PINN + PML + quadratic NN using 5000 iterations after Figure 14d, (e) difference between Figures 15b and 15d, (f) the recovered velocity models from Figure 15d, (g-i) the same as (d-f), but using 20000 iterations after Figure 14g.