A modified physics-informed neural network with positional encoding

Xinquan Huang, Tariq Alkhalifah and Chao Song, King Abdullah University of Science and Technology*

SUMMARY

Recently developed physics-informed neural network (PINN) for solving for the scattered wavefield in the Helmholtz equation showed large potential in seismic modeling because of its flexibility, low memory requirement, and no limitations on the shape of the solution space. However, the predicted solutions were somewhat smooth and the convergence of the training was slow. Thus, we propose a modified PINN using sinusoidal activation functions and positional encoding, aiming to accelerate the convergence and fit better. We transform the scalar input coordinate parameters using positional encoding into high-dimensional embedded vectors and train a fully-connected neural network to predict the real and imaginary parts of the scattered waveifeld. Numerical results show that, compared to the commonly used PINN, the proposed modified PINN using positional encoding exhibits notable superiority in terms of convergence and accuracy.

INTRODUCTION

Frequency-domain wave equation modeling is more computationally efficient compared to time-domain wave equation modeling (Marfurt, 1984), especially for applications like multiscale full waveform inversion (FWI). However, when the size of the subsurface model increases and the frequency is high, the computational cost of frequency-domain seismic modeling in general is high (increases exponentially with frequency). This limits our ability to perform FWI on high frequency data.

The universal approximation theorem (Hornik et al., 1989) states that any continuous function may be approximated by a sufficiently large and deep neural network (DNN). Considering that the seismic wavefield in the frequency domain is a function of sparse and it is inherently smooth, we can use a neural network to approximate this function. The typical training of a deep neural network requires massive labeled data, which is inefficient for seismic modeling and sometimes unavailable. Thanks to the known governing equation for seismic wave propagation in the frequency domain, which is Helmholtz equation, physics-informed loss function can be used to train the network in a self-supervised fashion. This is referred to as a physics-informed neural network (PINN). Actually, the idea of using neural networks (NNs) as surrogate modeling for partial differential equations (PDEs) was suggested decades ago (Lee and Kang, 1990; Lagaris et al., 1998, 2000). With the increasing recent sing popularity of machine learning and the recent developments in computing power, such applications are becoming more feasible. Raissi et al. (2019) introduced a mesh-free approach. Alkhalifah et al. (2020) applied PINN to predict the scattered Green's function in the frequency domain. Considering the complexity of wavefields, the predicted wavefields are somewhat smoother than the true ones. Thus, in this abstract, we explore the ways to improve the performance of PINN for wavefield solutions.

In PINN, we seek to find the NN parameters capable of admitting the wavefield and its second-order derivatives required by the wave equation. However, the commonly used activation function ReLU in convolutional NN can not be used here because its second-order derivate is equal to zero. Even though other alternative activation functions, like tanh, atan, and softplus, can be used, their performances tend to be less than optimal. On the other hand, periodic activation functions has shown their ability in representing details on implicit neural scene representation and full wavefield representations (Sitzmann et al., 2020; Song et al., 2021). However, its direct application to seismic scattered wavefield has showed improvement only with certain hyperparameter choices. To address this limitation and further improve the accuracy, we propose a modified PINN using the sine activation function, but now with positional encoding. In the following sections, we compare their performance on a simple layered model extracted from the Marmousi model and demonstrate the ability of our proposed method both to accelerate the convergence and to further improve the details.

SCATTERED WAVEFIELD USING PINN

We first introduce the concept of PINN applied to the scattered wavefield. Because of the point source singularity, the success of numerical methods are highly dependent on the sampling of the source region. This is even a bigger issue with PINN predicted wavefields (Alkhalifah et al., 2020). To avoid this limitation, we solve for the scattered wavefield, $\delta\mathbf{U} = \mathbf{U} - \mathbf{U}_0$, satisfying the following equation:

$$\omega^2\mathbf{m}\delta\mathbf{U} + \nabla^2\delta\mathbf{U} + \omega^2\delta\mathbf{m}\mathbf{U}_0 = 0, \qquad (1)$$

where $\mathbf{U}_0$ is the background wavefield, $\mathbf{m}$ is the squared slowness, ω is the angular frequency, and $\delta\mathbf{m}$ is the squared slowness perturbation. In PINN, we train a neural network F to map the coordinates of wavefield and source location, $\mathbf{x} = x, z, x_s$, to the wavefield values $\mathbf{U}$ within this physical constraint.

$$F : \mathbf{x} \rightarrow \mathbf{U}, \quad s.t. \quad Eq.(1). \qquad (2)$$

Then the loss function can be written as:

$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}\left|\omega^2 m^i \delta U^i + \nabla^2\delta U^i + \omega^2\delta m^i U_0^i\right|_2^2, \qquad (3)$$

where i and N represent the sample index and the number of samples in the space domain, respectively. Here we just need the background wavefield, which can be calculated with a constant background velocity using analytical solution, in this case, the training does not need any labeled data.

In this study, the mapping is performed between the coordinate values and the seismic wavefield. Instead, we use an embedding vector from positional encoding as input to the network.

POSITIONAL ENCODING

Although DNN theoretically can approximate scattered wavefield functions, the training of such NN resulted in smooth representations of the scattered wavefield, as well as slow convergence. Actually, this phenomenon can be contributed to the low-frequency bias of DNN (Rahaman et al., 2018). Representing the input with higher frequency features may help alleviate this limitation (Rahaman et al., 2018; Mildenhall et al., 2020). Thus, we utilize this feature to improve the neural representation of seismic wavefields.

The key idea here is to transform the inputs, given by the slowly varying values of the dimensions in space, into higher frequency representations. This can be achieved by transforming the input into higher-dimensional space, and the easiest way to do so is to use one-hot encoding. However, considering the large training samples, which are drawn randomly over the space range, the range of coordinate values is large due to random sampling in the continuous space. The one-hot encoding of space coordinates will result in very large sparse embedding vectors, and this will add to the computational burden for our training. Thus, we use positional encoding as an alternative encoding procedure. In positional encoding, we use a series of sinusoidal functions (Vaswani et al., 2017) to represent the inputs:

$$pe(\mathbf{x}) = [sin(2^0\pi\mathbf{x}), cos(2^0\pi\mathbf{x}), ..., sin(2^{d-1}\pi\mathbf{x}), cos(2^{d-1}\pi\mathbf{x})], \quad (4)$$

where d is the dimension of positional encoding and is set to 4 in the following experiments. We normalize the input $\mathbf{x}$ to lie between $[-1.0, 1.0]$ and apply pe to each coordinate. The resulting 8-dimensional embedded vector for each input parameter and original inputs are grouped to form a new 27-dimensional input to the NN.

NUMERICAL EXPERIMENTS

We will test three versions of PINN to evaluate the ability of positional encoding (PE) to accelerate the training process and improve the accuracy within the same number of training epochs. The tests are based on a simple layered model extracted from the Marmousi model (Figure 1) covering an area of 2.5×2.5 km^2. We generated 40000 random samples from the areas defined by x, z for wavefield, x_s for the horizontal source location, δm for squared slowness perturbation, m_0 for background squared slowness. The depth of source locations z_s is set to 0.025 km. The background wavefield is calculated analytically for a background velocity of 1.5 km/s. We train an 8-layer fully-connected NN (Figure 2) considering a frequency of 5 Hz for 100000 epochs using Adam optimizer. To evaluate the results, we solve the Helmholtz equation for a frequency of 5 Hz along with the velocity model in Figure 1. The solution is discretized in the 2.5×2.5 km^2 area using 100 samples in both the x and z directions. The sources are located at a depth of 0.025 km and lateral distances of 1.0 km and 1.5 km. In the following experiments, we show the results of three networks with different settings but using the same hyper-parameters (training samples, learning rate, and optimizer). The first one is a PINN using the tanh activation function without PE; the second one is a PINN using the sine activation function without PE, and the third one is the PINN using the sine activation function with PE.

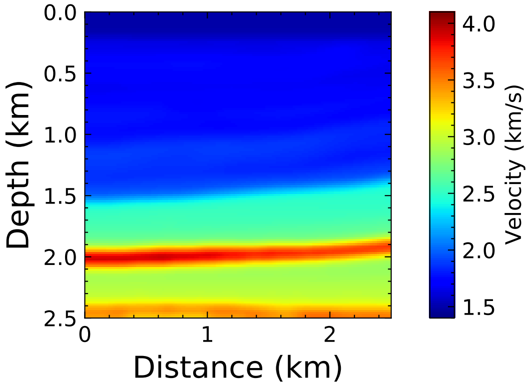


Figure 1: The true velocity model extracted from the Marmousi model.

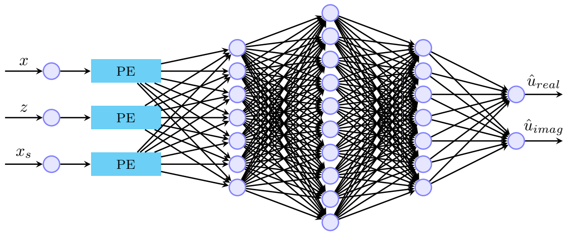


Figure 2: A fully-connected NN comprises of 8 hidden layers with 64, 64, 32, 32, 16, 16, 8, 8 neurons in each layer. The outputs are the real and imaginary parts of the scattered wavefield.

Figure 3 shows that by using tanh activation function, the loss stagnates for thousands of epochs and decrease slowly later. On the other hand, the sine activation function (a periodic function) can accelerate the convergence of the training process to some extent. The predictions of the wavefield at 1.0 km (Figure 4) and 1.5 km (Figure 6) are compared with the true ones. However, as shown in Figures 5a, 7a, 5b, 7b, the predictions do not show much improvement. As for the loss and prediction using the sine activation function with PE, it improves both convergence and accuracy. The commonly observed stagnation of the loss curves is considerably reduced and as a result, the convergence is much faster. Figures 5c and 7c also show that the prediction with PE is much better than the ones without PE, as the details of wavefield are better represented. We also calculated the velocity models using the PINN predictions according to Eq. 1 (Song and Alkhalifah, 2020), as shown in Figure 8. We see that the PINN with PE reconstructs higher resolution details of the velocity model, compared to the other predictions.

To further demonstrate the merits of the PE with exactly the same number of training parameters, we add a layer to the conventional NN. To do so, we consider the network with the PE

10.1190/segam2021-3584127.1

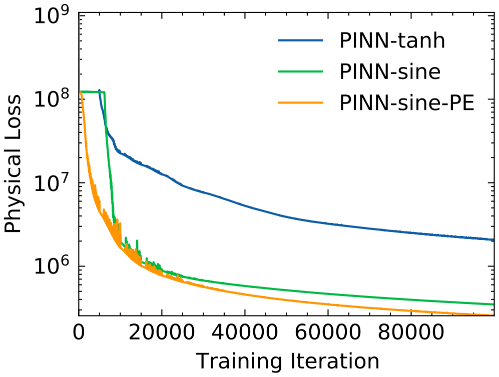
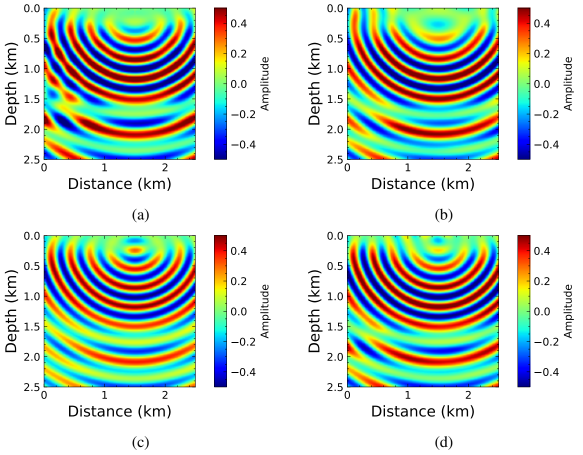
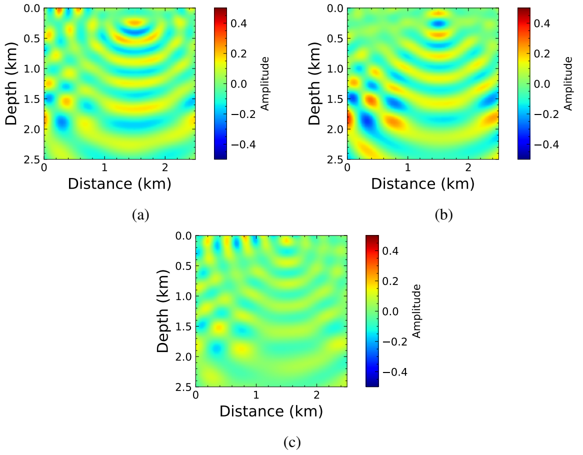


Figure 3: The comparion for loss curves of these three models.



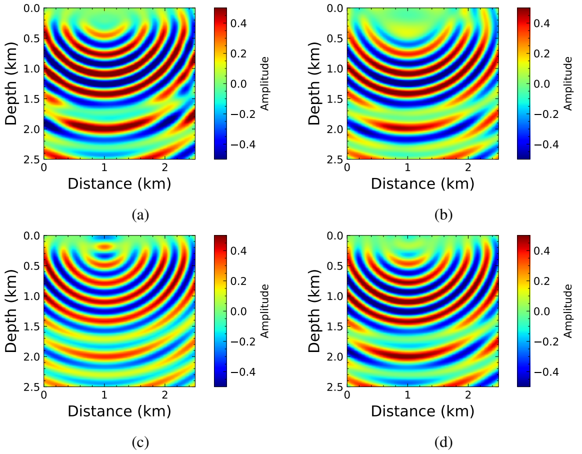
(a) (b) (c) (d)

Figure 4: The real parts of scattered wavefield solution for a source located at 1.5 km via a) numerical methods, b) PINN using tanh without PE, c) PINN using sine without PE, d) PINN using sine with PE.



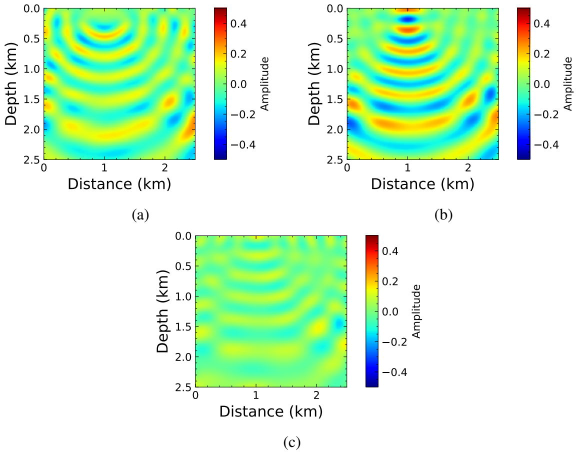
(a) (b) (c)

Figure 5: The difference between true and NN solutions at 1.5 km via a) PINN using tanh without PE, b) PINN using sine without PE, c) PINN using sine with PE.



(a) (b) (c) (d)

Figure 6: The imaginary parts of scattered wavefield solution for a source located at 1.0 km via a) numerical methods, b) PINN using tanh without PE, c) PINN using sine without PE, d) PINN using sine with PE.

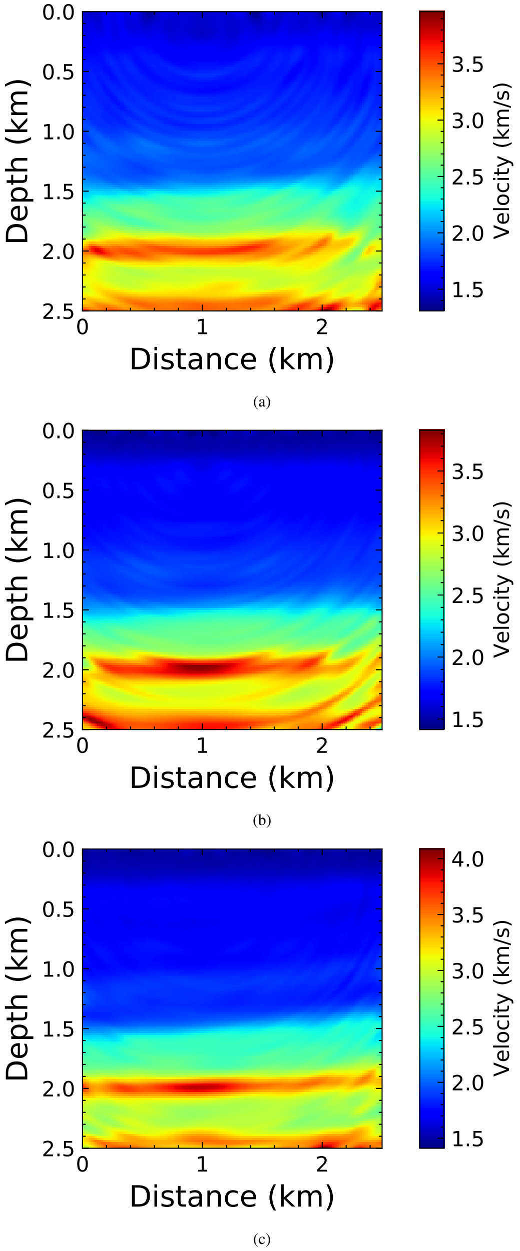


(a) (b) (c)

Figure 7: The difference between true and NN solutions at 1.0 km via a) PINN using tanh without PE, b) PINN using sine without PE, c) PINN using sine with PE.

as a 9-hidden-layer DNN with the first hidden layer fixed. So to be fair, we train one more fully-connected NN, whose 9 hidden layers are 27, 64, 64, 32, 32, 16, 16, 8, 8. The number of Adam optimizer epoch here is set to 50000. The convergence curve are shown in Figure 9. The stagnation feature is even worse than before as the network has more parameters. The loss for the commonly used tanh is still high even with a lot of iterations. Although the PINN using sine activation function without PE can arrive at a lower loss than the PINN using the sine activation function with PE, it shows overfitting reflected in the mean square errors of the predictions (Table 1). Thus, the experiments show that NN with PE can accelerate the convergence and avoid overfitting.

10.1190/segam2021-3584127.1



(a)

(b)

(c)

Figure 8: The velocity models estimated using predicted wavefield via a) PINN using tanh without PE, b) PINN using sine without PE, c) PINN using sine with PE.

Table 1: The normalized mean square errors of prediction using different NN setups.

Source Location	tanh	sine	sine + PE
1.0 km (real parts)	0.565	0.501	0.195
1.0 km (imaginary parts)	0.539	0.557	0.177
1.5 km (real parts)	0.610	0.543	0.203
1.5 km (imaginary parts)	0.525	0.539	0.240

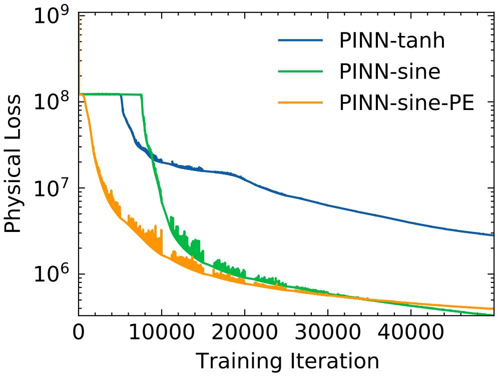


Figure 9: The comparison for loss curves of models for the 9-layered NN and NN with PE.

DISCUSSION

As mentioned above, DNN has a low-frequency-bias property, which means it tends to converge to a low frequency representation of a function. Thus, the slow variation in the input values given by the space dimension will not map well to a fast varying output wavefield representing the input values. In other words, for the dimension x as an example, two nearby points will vary in value by a small amount as inputs relative to the complexity of the network up to a point that their effective influence on the output is lowered resulting in smoother outputs. This feature is described in detail in Rahaman et al. (2018). The positional encoding will map these slow varying variables to higher dimensional vectors that can influence the network more and result in higher accuracy. Though the example shared in this abstract is simple, we will show more realistic examples as part of the presentation of the work in the conference.

CONCLUSION

We proposed the use of positional encoding to improve the performance of PINN in predicting fast varying wavefield solutions. This resulted in faster training convergence and higher prediction accuracy. We specifically transform the inputs using positional encoding given by sinusoidal functions to a higher dimensional space, and then feed them into a PINN with sine activation functions. Applications on a simple layered model showed that the proposed NN with positional encoding performed much better, which may provide a stepping stone for high-resolution wavefield representation using PINN in the future.

ACKNOWLEDGMENTS

We thank KAUST for its support and the SWAG group for the collaborative environment. This work utilized the resources of the Supercomputing Laboratory at King Abdullah University of Science and Technology (KAUST) in Thuwal, Saudi Arabia, and we are grateful for that.

10.1190/segam2021-3584127.1

REFERENCES

Alkhalifah, T., C. Song, and U. bin Waheed, 2020, Machine learned Green's functions that approximately satisfy the wave equation: 90th Annual International Meeting, SEG, Expanded Abstracts, 2638–2642, doi: https://doi.org/10.1190/segam2020-3421468.1.

Hornik, K., M. Stinchcombe, and H. White, 1989, Multilayer feedforward networks are universal approximators: Neural Networks, **2**, 359–366, doi: https://doi.org/10.1016/0893-6080(89)90020-8.

Lagaris, I., A. Likas, and D. Fotiadis, 1998, Artificial neural networks for solving ordinary and partial differential equations: IEEE Transactions on Neural Networks, **9**, 987–1000, doi: https://doi.org/10.1109/72.712178.

Lagaris, I., A. Likas, and D. Papageorgiou, 2000, Neural-network methods for boundary value problems with irregular boundaries: IEEE Transactions on Neural Networks, **11**, 1041–1049, doi: https://doi.org/10.1109/72.870037.

Lee, H., and I. S. Kang, 1990, Neural algorithm for solving differential equations: Journal of Computational Physics, **91**, 110–131, doi: https://doi.org/10.1016/0021-9991(90)90007-N.

Marfurt, K. J., 1984, Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations: Geophysics, **49**, 533–549, doi: https://doi.org/10.1190/1.1441689.

Mildenhall, B., P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, 2020, NERF: representing scenes as neural radiance fields for view synthesis: Computer Vision – ECCV 2020, Springer International Publishing, 405–421, doi: https://doi.org/10.1007/978-3-030-58452-8_24.

Rahaman, N., A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville, 2018, On the spectral bias of neural networks: arXiv.

Raissi, M., P. Perdikaris, and G. 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, doi: https://doi.org/10.1016/j.jcp.2018.10.045.

Sitzmann, V., J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, 2020, Implicit neural representations with periodic activation functions: arXiv.

Song, C., and T. Alkhalifah, 2020, Wavefield reconstruction inversion via machine learned functions: 90th Annual International Meeting, SEG, Expanded Abstracts, 1710–1714, doi: https://doi.org/10.1190/segam2020- 3427351.1.

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, doi: https://doi.org/10.1093/gji/ggab010.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, 2017, Attention is all you need: arXiv.

10.1190/segam2021-3584127.1