ECCOMAS

Proceedia

FOURIER NEURAL OPERATOR SURROGATE MODEL TO PREDICT 3D SEISMIC WAVES PROPAGATION

Fanny Lehmann[1,2], Filippo Gatti[2], Michaël Bertin[1], and Didier Clouteau[2]

[1]CEA/DAM/DIF
F-91297, Arpajon, France
e-mail: michael.bertin@cea.fr

[2] Université Paris-Saclay, ENS Paris-Saclay, CentraleSupélec, CNRS, LMPS - Laboratoire de
Mécanique Paris-Saclay
91190 Gif-sur-Yvette, France
e-mail: {fanny.lehmann,filippo.gatti,didier.clouteau}@centralesupelec.fr

Abstract. *With the recent rise of neural operators, scientific machine learning offers new solutions to quantify uncertainties associated with high-fidelity numerical simulations. Traditional neural networks, such as Convolutional Neural Networks (CNN) or Physics-Informed Neural Networks (PINN), are restricted to the prediction of solutions in a predefined configuration. With neural operators, one can learn the general solution of Partial Differential Equations, such as the elastic wave equation, with varying parameters. There have been very few applications of neural operators in seismology. All of them were limited to two-dimensional settings, although the importance of three-dimensional (3D) effects is well known.*

In this work, we apply the Fourier Neural Operator (FNO) to predict ground motion time series from a 3D geological description. We used a high-fidelity simulation code, SEM3D, to build an extensive database of ground motions generated by 30,000 different geologies. With this database, we show that the FNO can produce accurate ground motion even when the underlying geology exhibits large heterogeneities. Intensity measures at moderate and large periods are especially well reproduced.

We present the first seismological application of Fourier Neural Operators in 3D. Thanks to the generalizability of our database, we believe that our model can be used to assess the influence of geological features such as sedimentary basins on ground motion, which is paramount to evaluating site effects.

Keywords: Elastic wave equation, Artificial intelligence, Neural operators, Surrogate model, Seismology, Deep learning

1 Introduction

Thanks to the recent increase in computational resources, high-fidelity simulations have become more affordable for diverse engineering applications. Complex physical systems can be emulated with simulations depending on many parameters. However, depending on the data available to calibrate the simulations, some parameters may be poorly constrained. Therefore, a lot of efforts are now devoted to quantifying the influence of each parameter on the global system response [1, 2, 3].

Physics-based simulations rely on the discretization of the underlying Partial Differential Equations (PDEs). The most common methods to solve PDEs include Finite Elements Methods, Finite Difference Methods, Finite Volume Methods, and Spectral Elements Methods. Despite this diversity, all methods suffer from high computational costs, especially for complex physical systems. Since hundreds to thousands of simulations are necessary for uncertainty quantification analyses in Monte-Carlo Markov Chain frameworks, physics-based models cannot be used directly and faster methods need to be designed.

To reduce the computational time of high-fidelity simulations, surrogate models have been developed. Classical methods, such as Gaussian processes, Kriging, Polynomial chaos, etc., depend on a limited set of parameters [4, 5, 6, 7, 8]. Uncertainty quantification has benefited from the development of neural networks which account for a larger number of parameters and reduce the *a priori* selection of influential variables. Among standard methods, Convolutional Neural Networks (CNNs, [9]) and Physics-Informed Neural Networks (PINNs, [10]) have witnessed successful engineering applications [11, 12, 13]. However, one major drawback of these methods is the difficulty of transferring knowledge between different configurations. For example, when solving the wave equation, CNNs and PINNs are trained with a fixed velocity parameter and cannot predict anything for a different velocity value.

Neural operators have been introduced to solve this issue and improve the generalizability of deep learning methods [14, 15, 16]. The paradigm of neural operators is to consider inputs and outputs as functionals. Therefore, PDEs can be solved for a large class of parameters. This work focuses on the Fourier Neural Operator (FNO) developed by [15]. In seismology, the FNO has been applied for earthquake localization [17], to predict high-frequency terms from low-frequency simulations [18], and to solve the acoustic wave equation [19, 20]. To our best knowledge, the first application on the elastic wave equation, *i.e.* solving all components of ground motion, was done by [21, 22].

However, all those applications were limited to one-dimensional (1D) or two-dimensional (2D) domains, while it is known that three-dimensional (3D) effects may be crucial [23, 24]. Designing a 3D FNO is a technical challenge due to the memory requirements of the model. The very few studies that attempted to do so either used linear attention to reduce the dimensionality [25] or advanced model parallelization [26, 27].

In this study, we conducted the first prediction of elastic waves propagation in a 3D domain. We used a FNO architecture called the U-shaped neural operator (UNO, [28]) that allows a deep architecture with ten layers. Solving a time-dependent PDE on a 3D domain would ultimately represent a 4D problem (three spatial and one temporal dimensions) which is not affordable. To circumvent this issue, we propose a dimension conversion from the input's depth dimension to the output's temporal dimension. We applied our UNO to seismological data, but our conclusions can be extrapolated to other fields involving propagation equations.

2 Neural Operator architecture

2.1 Problem formulation

The propagation of seismic waves in a heterogeneous isotropic material can be described by the elastic wave equation

$$\rho\frac{\partial^2 \boldsymbol{u}}{\partial t^2} = \nabla\lambda\left(\nabla\cdot\boldsymbol{u}\right) + \nabla\mu\left[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right] + (\lambda + 2\mu)\nabla\left(\nabla\cdot\boldsymbol{u}\right) - \mu\nabla\times\nabla\times\boldsymbol{u} \quad (1)$$

where ρ is the material density, $\boldsymbol{u}$ is the displacement, and λ, μ are the Lamé parameters characterizing the material. Alternatively, the material can be described by the velocity of shear waves, V_S, and the velocity of compressional waves, V_P. In this work, we assumed that the ratio V_P/V_S was constant and equal to α (where α=1.7) to describe the material with the only parameter V_S. Therefore, equation 1 can be reformulated in terms of V_S as

$$\frac{\partial^2 \boldsymbol{u}}{\partial t^2} = (\alpha^2 - 2)\nabla V_S^2\left(\nabla\cdot\boldsymbol{u}\right) + \nabla V_S^2\left[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right] + (\alpha^2 - 1)V_S^2\nabla\left(\nabla\cdot\boldsymbol{u}\right) - V_S^2\nabla\times\nabla\times\boldsymbol{u} \quad (2)$$

In a general framework, let us denote $D_a \subset \mathbb{R}^3$ the material domain and $\mathcal{A} = \mathcal{F}(D_a; \mathbb{R}^+)$ the space of functions describing the material property V_S. The displacement $\boldsymbol{u}$ intrinsically depends on four variables, $(x, y, z, t) \in D_a \times [0, T]$. However, this formulation is i) too demanding in terms of computational resources for a 4-dimensional neural network, ii) not realistic since the displacement is generally only measured with sensors at the surface. Therefore, we introduce the set $D_u = \{(x, y, t), (x, y) \in \partial D_a^{top}, t \in [0, T]\}$ where ∂D_a^{top} is the upper boundary of the material domain D_a and $T > 0$ is the final time. Then, we denote $\mathcal{U} = \mathcal{C}(D_u; \mathbb{R}^3)$ the space of continuous velocity functions $\boldsymbol{u}$ observed only at the surface of the domain. Note that the output space of $\boldsymbol{u}$ is $\mathbb{R}^3$ since the velocity is expressed along three components (East-West, North-South, vertical).

Then, denoting $a(\boldsymbol{x}) = V_S(\boldsymbol{x})^2$, equation 2 can be expressed in the general form $\mathcal{L}(a, u) = 0$ with initial conditions imposed by the seismic source and absorbing boundary conditions to represent an infinite spatial domain. In this setting, we finally aim at finding an approximation of

$$\mathcal{G}: \begin{array}{ccc} \mathcal{A} & \to & \mathcal{U} \\ a & \mapsto & \boldsymbol{u} \end{array} \quad (3)$$

defined as a neural network G_θ depending on a set of parameters θ.

2.2 Fourier Neural Operator

The FNO formulation originates from the theory of kernel integral operators [29, 14]. It has been further modified to benefit from the computational efficiency of the Fast Fourier Transform ([15]). The FNO can be decomposed into three components:

1. an uplift sub-network P that transforms the input a into an abstract and higher-dimensional vector v_0

2. a succession of L Fourier layers transforming abstract vectors $v_\ell \mapsto v_{\ell+1}$, described below

3. a projection sub-network Q that maps the abstract vector v_L resulting from the last Fourier layer to the physical output $\boldsymbol{u}$

A Fourier layer F_ℓ can be described, with $\mathcal{F}$ denoting the Fast Fourier Transform, by

$$v_{\ell+1} = \sigma \left(\mathcal{F}^{-1}(R_\ell \cdot \mathcal{F}(v_\ell)) + W_\ell v_\ell \right) \tag{4}$$

It first performs the convolution of the Fourier coefficients of its input vector $\mathcal{F}(v_\ell)$ with a kernel operator $\mathcal{K}_\ell$ learnt during the neural network training. This is similar to the operation performed by usual convolutional layers, except that the kernel is no longer a matrix but an operator. To alleviate the computational cost, the convolution is performed in the Fourier space, where it is equivalent to multiplication. Therefore, the kernel operator $\mathcal{K}_\ell$ can be represented by a fixed number of Fourier coefficients $R_\ell := \mathcal{F}(\mathcal{K}_\ell)$. To maintain the physical meaning throughout the process, the convolution is followed by the inverse Fourier transform $\mathcal{F}^{-1}$. Meanwhile, a bias is added through a linear operation parametrized by W_ℓ. Finally, a non-linear activation function σ is applied.

One major advantage of neural operators is their discretization-invariance property. Indeed, the weights R_ℓ depend on the number of Fourier coefficients one chooses to select in each Fourier layer and not on the dimension of v_ℓ. Therefore, the neural operator can be trained with moderate dimensions and, once trained, be applied to higher-resolution data to highlight small-scale features.

2.3 Model used in this work

To increase the complexity of the neural network, and hence its expressivity, while avoiding the common problem of vanishing/exploding gradients, the UNO has been proposed by [28]. It arranges the Fourier layers in an encoder-decoder structure to allow skip connections from one Fourier layer in the encoder to its symmetric layer in the decoder. Our model comprises 8 Fourier layers, 4 in the encoder and 4 in the decoder.

In addition to adding skip connections to the original FNO, the UNO modifies the dimensions of vectors v_ℓ in each Fourier layer. In the encoder part, the physical dimensions are reduced from $64 \times 64 \times 64$ to $8 \times 8 \times 8$ while they are increased to $64 \times 64 \times 128$ in the decoder part.

The uplift sub-network P is made of two fully connected layers. It takes as input the geological description a concatenated with a positional encoding of coordinates in D_a. With this concatenation, P creates an abstract vector v_0 with the same dimensions as a and 16 additional channels. At the end of the model, there are three projection operators Q_E, Q_N, and Q_Z, each made of two fully connected layers. Each operator projects the vector v_L onto the respective velocity components u_E, u_N, u_Z.

3 Data

The UNO is trained in a fully supervised manner. Therefore, the training requires pairs $(a, \boldsymbol{u})$. Geological inputs were built from random fields to form a large and general database (Section 3.1). Then, finite elements simulations were performed to obtain the velocity fields at the surface (Section 3.2).

3.1 Geological database

Our geological database represents 3D S-wave velocity fields in a $9.6\,\text{km} \times 9.6\,\text{km} \times 9.6\,\text{km}$ cube. This database is built from layered random fields to provide a general decomposition basis to any 3D material [30]. Each geology contains a homogeneous bottom layer with constant velocity V_S=4500 m/s. On top of this layer, there are 1 to 6 layers whose thickness and mean
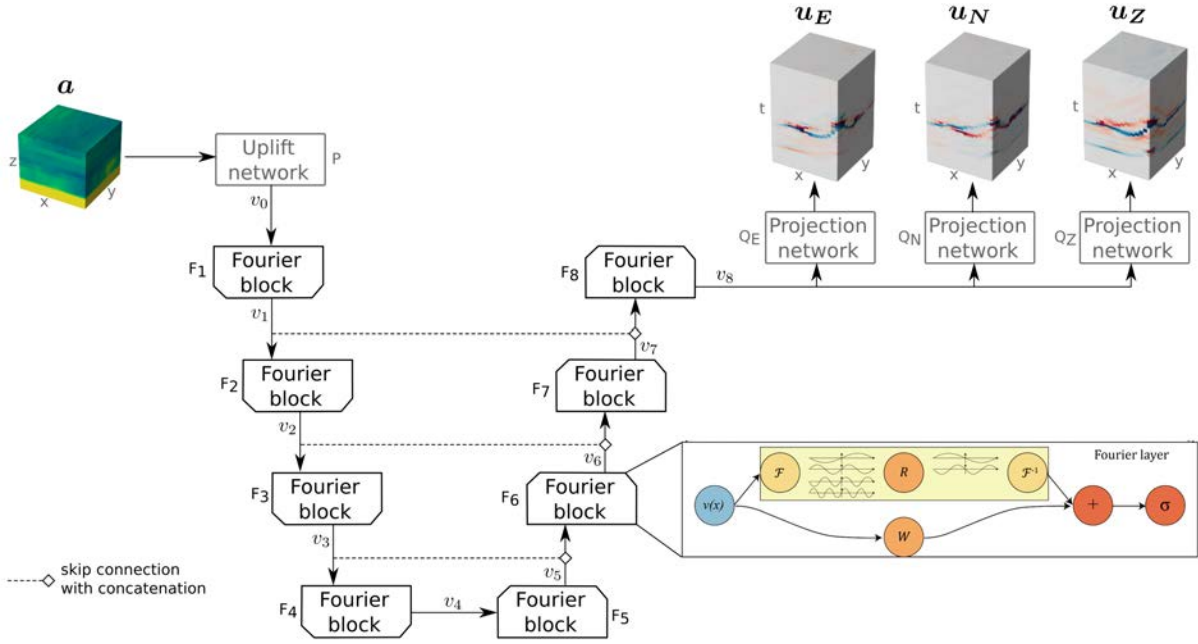
Figure 1: Architecture of our UNO. The input (3D geology a) is uplifted by the sub-network P, then transformed trough 8 Fourier layers $F_1, \cdots, F_8$. Finally, three sub-networks Q_E, Q_N, and Q_Z project the components of the velocity u_E, u_N, u_Z. The detail of the sixth Fourier block is shown in the right corner (image reproduced from [15]). Dotted lines show the skip connections with a concatenation of inputs.

value were randomly chosen. The mean value follows a uniform distribution $\mathcal{U}(1785$ m/s; 3214 m/s). There is no additional constraint, meaning that velocity inversion may be present.

Then, heterogeneities were added independently inside each layer with von Karman random fields which are known to best represent crustal heterogeneities. The correlation length was randomly chosen between 1.5km and 6km, and the coefficient of variation followed a normal distribution $\mathcal{N}(0.2, 0.1)$ of mean 0.2 and standard deviation 0.1. Finally, the velocities were clipped to the interval [1071 m/s; 4500 m/s] to ensure realistic values and an appropriate discretization of wave lengths.

For the neural operator training, geologies were represented as $64 \times 64 \times 64$ matrices. The geological database amounts to 30Go.

3.2 Finite elements 3D simulations

To obtain ground motion time series, we used SEM3D, a High-Performance Computing (HPC) code based on the Spectral Element Method [31, 32]. Each geology was discretized to a hexahedral mesh with elements of size 300m. A seismic source was placed in the middle of the bottom layer at position (4.8km, 4.8km, -8.4km). It is parametrized as a moment tensor with strike = 50°, dip = 45°, and rake = 88°. Its source time function is given by $t \mapsto 1 - \left(1 + \frac{t}{\tau}\right) e^{-\frac{t}{\tau}}$ with $\tau = 0.127$ s. These parameters correspond to the Le Teil earthquake (France, 2019, [33]).

Each simulation runs for 20s and can accurately reproduce frequencies up to 5 Hz. The generation of 30,000 simulations represents 1.6 million CPU hours (50 min per simulation on 64 CPUs). During the simulation, ground motion velocities were recorded on a grid of 256 virtual sensors equally spaced at the surface (the space between two consecutive sensors was 600m). The acquisition frequency is 20 Hz.

For the neural operator training, we used the ground motion time series between 1s and 7.4s (128 time points). In addition, we performed a 2D spatial interpolation to refine ground motion to a 64×64 spatial grid. The velocity database amounts to 176 Go.

3.3 Neural operator training

The database of 30,000 instances was split into 90% training data and 10% validation data. The geological inputs were normalized to a Gaussian with mean 0 and standard deviation 0.25. No normalization was applied to the outputs.

The neural operator has been trained with the Adam optimizer for 110 epochs. The learning rate was initially 10^{-3} and was reduced by 0.5 when the validation loss did not improve for 20 epochs. The activation function was the relu function and the loss function was defined as the Mean Absolute Error (MAE). In total, the neural operator contains 87 million parameters. The training took 11 hours on 4 Nvidia A100 GPUs.

4 Results

4.1 General prediction results

Figure 2 shows the convergence of the training and validation loss functions. The exponential decreasing curve illustrates the quality of the neural operator learning. At the end of the training, the validation loss is slightly higher than the training loss, which may indicate a slight overfitting of the data. However, when comparing the distribution of the prediction error for 1,000 instances in the training and validation datasets, it appears that the distributions are very similar (Figure 3). Therefore, if overfitting is present, it should have a minimal effect on the validation results' accuracy.
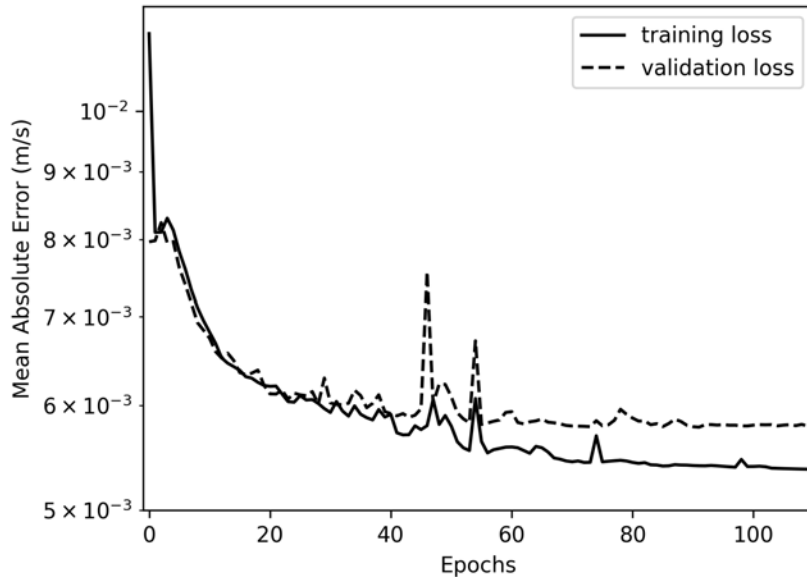

Figure 2: Evolution of the training loss (line) and the validation loss (dashed line) as a function of epochs. The loss is summed over the three components.

Additionally, Figure 3 shows that the prediction accuracy is similar for the three components (East-West, North-South, vertical). This means that the projection sub-networks Q_E, Q_N, and

Q_Z are sufficient to capture the specificities of each velocity component from the last abstract vector v_8. Therefore, the following results are presented only for the East-West component, and their interpretation can be easily transferred to the other components.
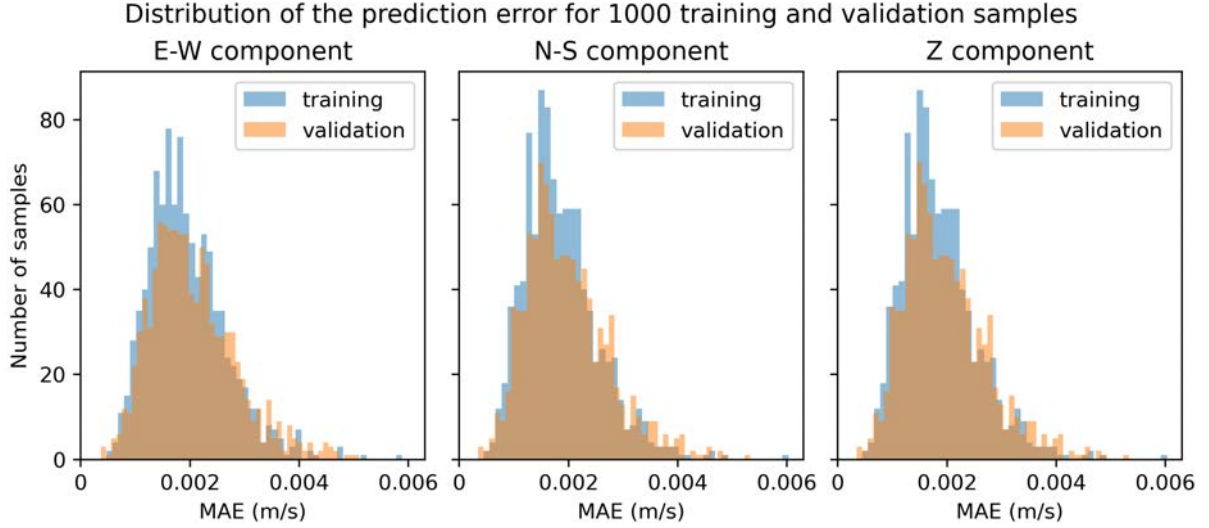

Figure 3: For 1000 elements of the training dataset (blue) and validation dataset (orange), distribution of the Mean Absolute Error (MAE) between the neural operator prediction and the ground truth. Each subpanel shows one velocity component (E-W: East-West), (N-S: North-South), (Z: vertical).

4.2 Ground motion prediction for validation data samples

Figure 4 shows the neural operator prediction for one geology in the validation dataset. This geology has four layers with large velocity contrasts between layers (Figure 4a). The MAE on the E-W velocity component is $7.8 \ 10^{-4}$m/s. The time evolution at individual snapshots shows that the neural operator can accurately capture the time arrival of P-waves (the first arrival) and S-waves (the second and most prominent peak), despite their different amplitudes (Figure 4b). In addition, the velocity magnitude is also spatially close to the reference, as seen on the snapshots in Figure 4c.

We found that the neural operator tends to underestimate the Peak Ground Velocity (PGV). This is visible in Figure 4b where the prediction peaks are generally smaller than the ground truth.

Figure 5 shows that predictions are also accurate when the neural operator input is a heterogeneous geology like depicted in Figure 5a. Heterogeneities diffracted the seismic waves, leading to more variations in the velocity time series. However, the wave arrival times are still accurately captured by the neural operator (Figure 5b), and the spatial distribution of ground motion is very satisfactory (Figure 5c).

Quantitatively, one can compute the Goodness-of-Fit criteria [34]. This scale evaluates the difference between two time series regarding envelope and phase. Scores above 6 are generally considered good and above 8 considered very good (10 is a perfect match). Figure 6 shows the GOF corresponding to the predictions with the heterogeneous geology in Figure 5. One can observe that the envelope GOF is larger than 6 for 95% of the points, and the phase GOF is larger than 8 for 92% of points. These scores assess the quality of the UNO prediction. In
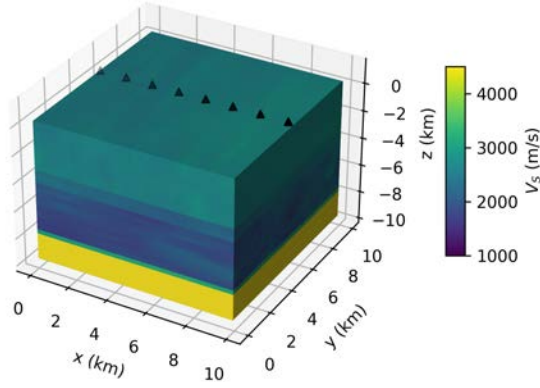
(a) 3D geological description (S-wave velocity V_S). The black triangles represent the sensors where the traces are shown in Figure 4b.
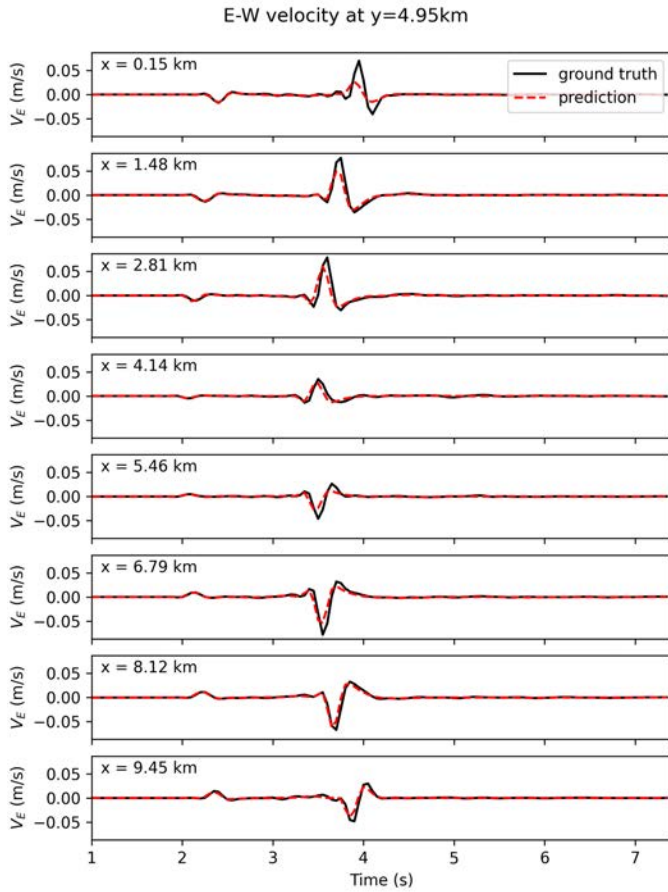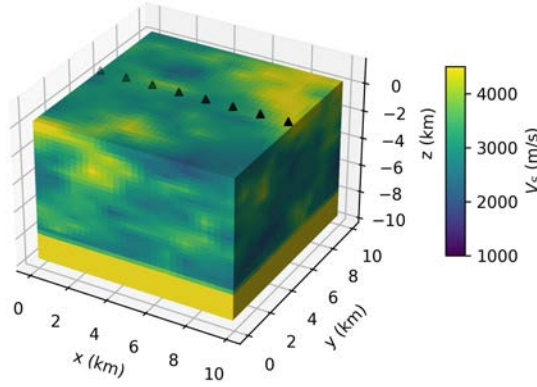

(b) E-W ground velocity time series at 8 sensors aligned at y=4.95km and represented in Figure 4a.
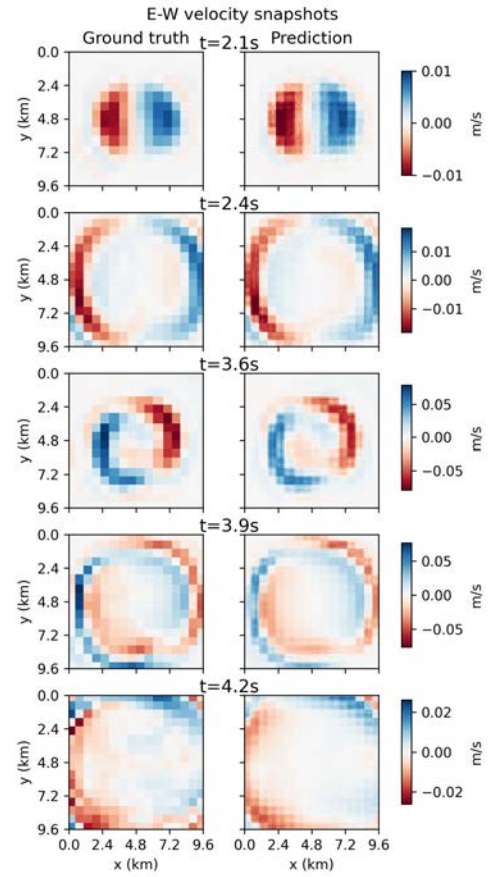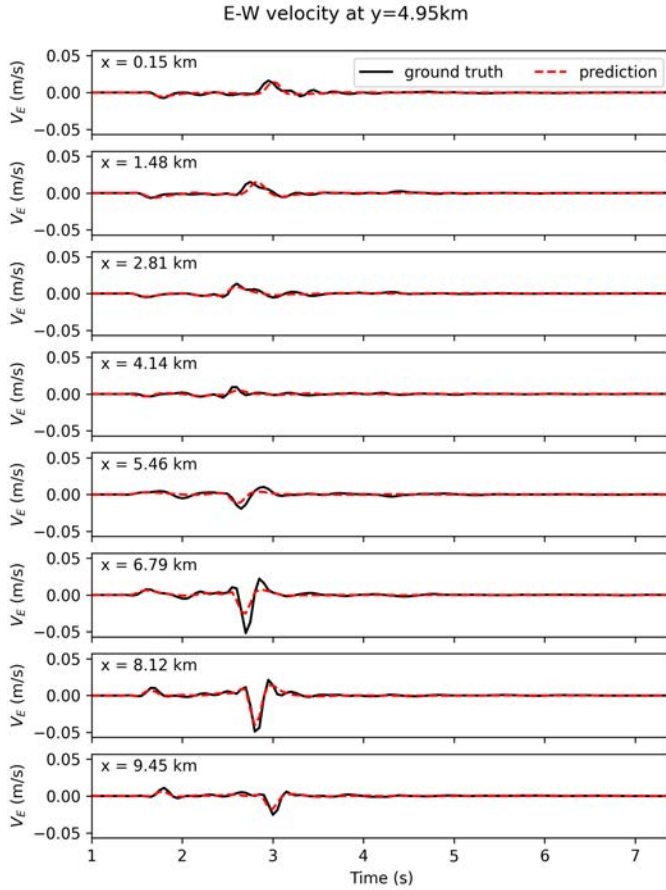
(c) Comparison of reference velocities (left column) and velocities predicted by the neural operator (right column) for 5 time points.
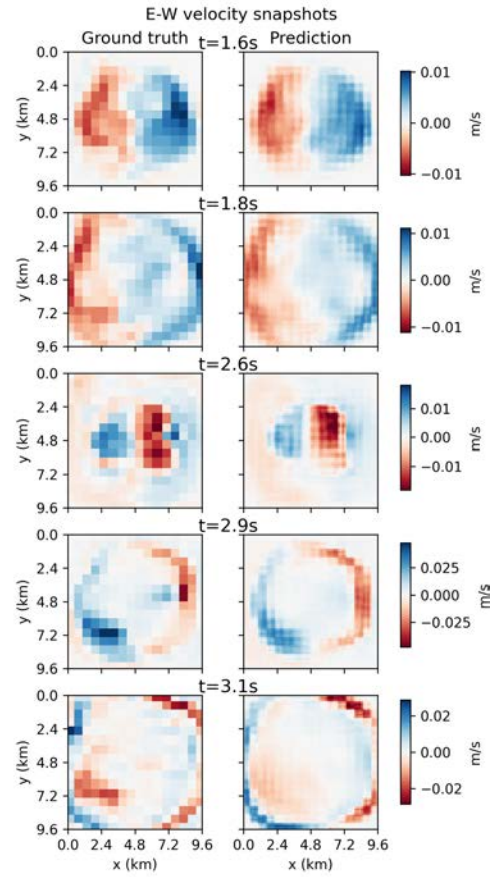
Figure 4: Comparison of simulations (considered as ground truth) and neural operator predictions for one geology in the validation dataset.

(a) 3D geological description (S-wave velocity). The black triangles represent the sensors where the traces are shown in Figure 5b.

(b) E-W ground velocity time series at 8 sensors aligned at y=4.95km and represented in Figure 5a.

(c) Comparison of reference velocities (left column) and velocities predicted by the neural operator (right column) for 5 time points.

Figure 5: Comparison of simulations (considered as ground truth) and neural operator predictions for one geology in the validation dataset.

addition, the better agreement on the phase confirms that the predictions are more accurate for the wave arrival times than for the peaks' magnitude.
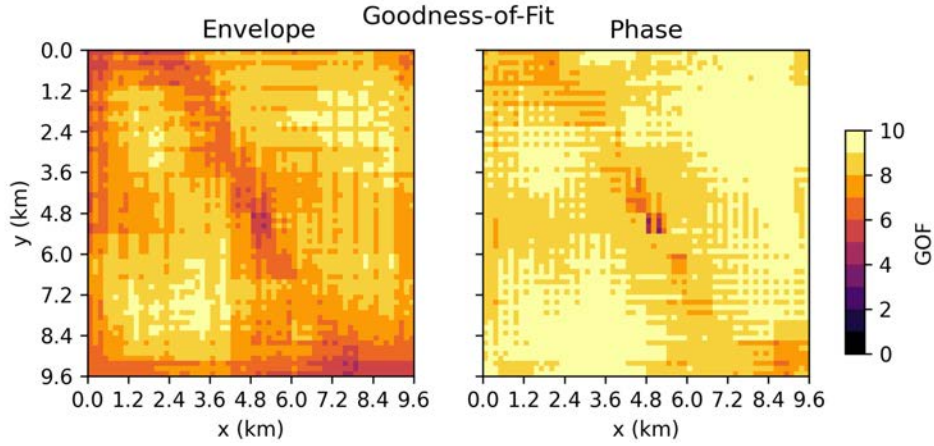

Figure 6: Goodness-of-Fit (GOF) criteria for the envelope and phase of all time series predicted with the heterogeneous geology (Figure 5a). 10 means a perfect agreement.

When examining the velocity fluctuations in Figure 5b, it appears that the smallest ones are generally missing from the neural operator prediction. This can also be observed in Figure 7 where the Fourier coefficients are shown as a function of the frequency. Although the prediction is very close to the ground truth at low frequency, the disagreement increases above 1 Hz. However, the amplitude decay has approximately the same slope for the prediction and the reference signal, meaning that the neural operator lacks accuracy on high-frequency variations but does not introduce data filtering.

5 Discussion and conclusion

We used a database of 3D heterogeneous geologies based on random fields to simulate the propagation of seismic waves. We showed that the UNO is able to learn the relationship between the geology and the surface ground motion represented by 3-component velocity time series. Therefore, our model maps the depth dimension of the geology to the time dimension of the velocity. This is an efficient way to model the 3D elastic wave equation while limiting the memory requirements of the model.

Contrary to a common belief in machine learning that it is better to start with easy tasks, we found that the training significantly improved when predicting the three components instead of a single one. This may be explained by the fact that adding two projection sub-networks Q_N and Q_Z does not significantly increase the number of parameters in the model but triples the amount of training data. Therefore, predicting multiple components was beneficial for the deep learning model.

The generalization gap between the validation loss and the training loss (Figure 2) could probably be reduced by using a larger database. Indeed, we observed that the validation loss was higher when using 20,000 samples instead of 30,000. Due to the computational cost of generating the database with high-fidelity simulations, we accepted this trade-off as a first step to validate the 3D UNO.

Despite this limitation, the predictions on validation data (Figures 4 and 5) show the accuracy of our model. It cannot be used at this point to predict maximum values since they tend to be underestimated. This reflects the findings of [17] that amplitudes are more difficult to predict
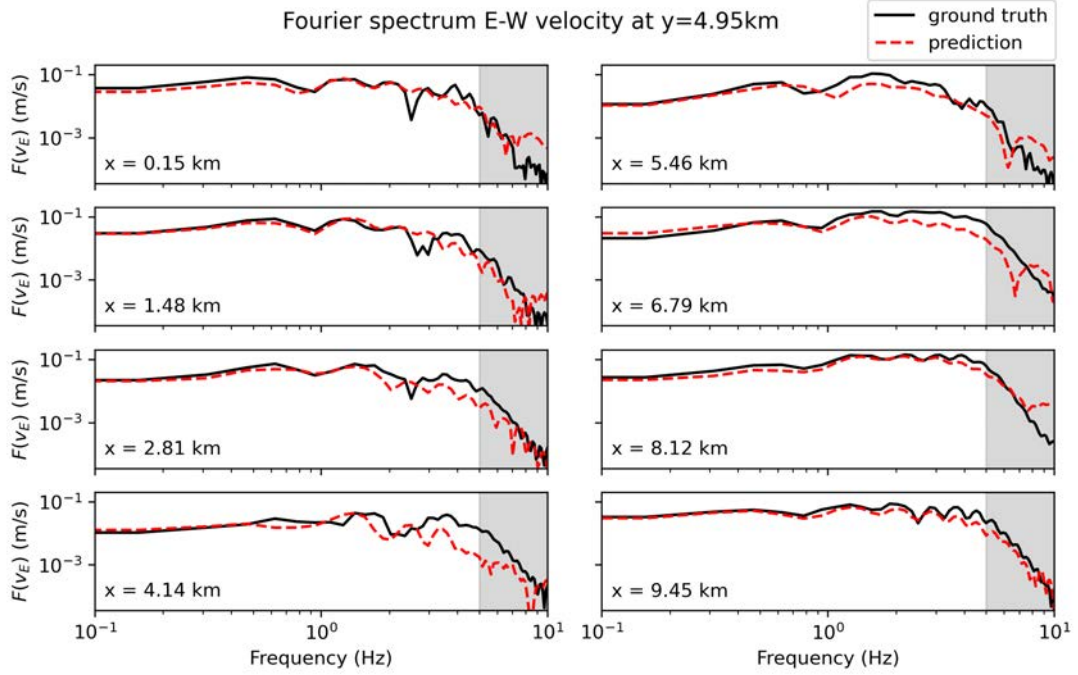
Figure 7: Amplitude of Fourier coefficients for reference velocity time series (black line) and time series predicted by the neural operator (dashed red line). The grey area represents the maximum frequency (5 Hz) where the numerical simulations are valid.

than phases. However, the amplitude accuracy should be improved by using more training data. The network architecture can also be refined, especially by increasing the number of Fourier modes in each block. This should allow a better representation of high-frequency variations. Nevertheless, the wave arrival times are already accurate despite geological heterogeneities that created dispersion. This is important for future applications in earthquake early warning, where time predictions are crucial.

References

[1] Shunsuke Homma et al. "A Physics-based Monte Carlo Earthquake Disaster Simulation Accounting for Uncertainty in Building Structure Parameters". In: *Procedia Computer Science* 29 (2014), pp. 855–865. DOI: `10.1016/j.procs.2014.05.077`.

[2] Inseok Park and Ramana V. Grandhi. "Quantifying Multiple Types of Uncertainty in Physics-Based Simulation Using Bayesian Model Averaging". In: *AIAA Journal* 49.5 (2011), pp. 1038–1045. DOI: `10.2514/1.J050741`.

[3] Heng Xiao and Paola Cinnella. "Quantification of Model Uncertainty in RANS Simulations: A Review". In: *Progress in Aerospace Sciences* 108 (2019), pp. 1–31. DOI: `10.1016/j.paerosci.2018.10.001`.

[4] G. Abbiati et al. "A Global Sensitivity Analysis Framework for Hybrid Simulation". In: *Mechanical Systems and Signal Processing* 146 (2021), p. 106997. DOI: `10.1016/j.ymssp.2020.106997`.

[5] Hugo Cruz-Jiménez et al. "Bayesian Inference of Earthquake Rupture Models Using Polynomial Chaos Expansion". In: *Geoscientific Model Development* 11.7 (2018), pp. 3071–3088. DOI: `10.5194/gmd-11-3071-2018`.

[6] Marvin Pförtner et al. *Physics-Informed Gaussian Process Regression Generalizes Linear PDE Solvers*. 2022. preprint.

[7] Aidin Tamhidi et al. "Conditioned Simulation of Ground-Motion Time Series at Uninstrumented Sites Using Gaussian Process Regression". In: *Bulletin of the Seismological Society of America* 112.1 (2022), pp. 331–347. DOI: `10.1785/0120210054`.

[8] Pierre Sochala, F. De Martin, and O. Le Maître. "Model Reduction for Large-Scale Earthquake Simulation in an Uncertain 3D Medium". In: *International Journal for Uncertainty Quantification* 10.2 (2020), pp. 101–127. DOI: `10.1615/Int.J.UncertaintyQuantificatio 2020031165`.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: `10.1038/nature14539`.

[10] M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707. DOI: `10.1016/j.jcp.2018.10.045`.

[11] George Em Karniadakis et al. "Physics-Informed Machine Learning". In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440. DOI: `10.1038/s42254-021-00314-5`.

[12] Zachary E. Ross et al. "Generalized Seismic Phase Detection with Deep Learning". In: *Bulletin of the Seismological Society of America* 108 (5A 2018), pp. 2894–2901. DOI: `10.1785/0120180080`.

[13] Chao Song, Tariq Alkhalifah, and Umair Bin Waheed. "A Versatile Framework to Solve the Helmholtz Equation Using Physics-Informed Neural Networks". In: *Geophysical Journal International* 228.3 (2022), pp. 1750–1762. DOI: `10.1093/gji/ggab434`.

[14] Zongyi Li et al. *Neural Operator: Graph Kernel Network for Partial Differential Equations*. 2020. arXiv: `arXiv:2003.03485`. preprint.

[15] Zongyi Li et al. *Fourier Neural Operator for Parametric Partial Differential Equations*. 2021. arXiv: `arXiv:2010.08895`. preprint.

[16] Lu Lu et al. "Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators". In: *Nature Machine Intelligence* 3.3 (2021), pp. 218–229. DOI: `10.1038/s42256-021-00302-5`.

[17]   Hongyu Sun et al. *Accelerating Time-Reversal Imaging with Neural Operators for Real-time Earthquake Locations*. 2022. arXiv: `arXiv:2210.06636`. preprint.

[18]   Chao Song and Yanghua Wang. "High-Frequency Wavefield Extrapolation Using the Fourier Neural Operator". In: *Journal of Geophysics and Engineering* 19.2 (2022), pp. 269–282. DOI: `10.1093/jge/gxac016`.

[19]   Bian Li et al. *Solving Seismic Wave Equations on Variable Velocity Models with Fourier Neural Operator*. 2022. preprint.

[20]   Wei Wei and Li-Yun Fu. "Small-Data-Driven Fast Seismic Simulations for Complex Media Using Physics-Informed Fourier Neural Operators". In: *Geophysics* 87.6 (2022), T435–T446. DOI: `10.1190/geo2021-0573.1`.

[21]   Yan Yang et al. *Rapid Seismic Waveform Modeling and Inversion with Universal Neural Operators*. 2022. preprint.

[22]   Yan Yang et al. "Seismic Wave Propagation and Inversion with Neural Operators". In: *The Seismic Record* 1.3 (2021), pp. 126–134. DOI: `10.1785/0320210026`.

[23]   Peter Moczo et al. "Key Structural Parameters Affecting Earthquake Ground Motion in 2D and 3D Sedimentary Structures". In: *Bulletin of Earthquake Engineering* 16.6 (2018), pp. 2421–2450. DOI: `10.1007/s10518-018-0345-5`.

[24]   Chuanbin Zhu et al. "Seismic Aggravation in Shallow Basins in Addition to One-dimensional Site Amplification". In: *Journal of Earthquake Engineering* 24.9 (2020), pp. 1477–1499. DOI: `10.1080/13632469.2018.1472679`.

[25]   Wenhui Peng et al. *Linear Attention Coupled Fourier Neural Operator for Simulation of Three-Dimensional Turbulence*. 2022. arXiv: `arXiv:2210.04259`. preprint.

[26]   Thomas J. Grady et al. *Model-Parallel Fourier Neural Operators as Learned Surrogates for Large-Scale Parametric PDEs*. 2022. preprint.

[27]   Philipp Witte, Russell Hewett, and Ranveer Chandra. "Industry-Scale CO2 Flow Simulations with Model-Parallel Fourier Neural Operators". In: *NeurIPS 2022 Workshop Tackling Climate Change with Machine Learning*. 2022.

[28]   Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. *U-NO: U-shaped Neural Operators*. 2022. arXiv: `arXiv:2204.11127`. preprint.

[29]   Owen Huang et al. *An Introduction to Kernel and Operator Learning Methods for Homogenization by Self-consistent Clustering Analysis*. 2022. arXiv: `arXiv:2212.00802`. preprint.

[30]   Fanny Lehmann et al. "Machine Learning Opportunities to Conduct High-Fidelity Earthquake Simulations in Multi-Scale Heterogeneous Geology". In: *Frontiers in Earth Science* 10 (2022). DOI: `10.3389/feart.2022.1029160`.

[31]   Dimitri Komatitsch and Jeroen Tromp. "Introduction to the Spectral Element Method for Three-Dimensional Seismic Wave Propagation". In: *Geophysical Journal International* 139.3 (1999), pp. 806–822. DOI: `10.1046/j.1365-246x.1999.00967.x`.

[32]   Sara Touhami et al. "SEM3D: A 3D High-Fidelity Numerical Earthquake Simulator for Broadband (0–10 Hz) Seismic Response Prediction at a Regional Scale". In: *Geosciences* 12.3 (2022), p. 112. DOI: `10.3390/geosciences12030112`.

[33]   Bertrand Delouis et al. "Constraining the Point Source Parameters of the 11 November 2019 Mw 4.9 Le Teil Earthquake Using Multiple Relocation Approaches, First Motion and Full Waveform Inversions". In: *Comptes Rendus. Géoscience* 353.S1 (2021), pp. 1–24. DOI: `10.5802/crgeos.78`.

[34]   Miriam Kristeková, Jozef Kristek, and Peter Moczo. "Time-Frequency Misfit and Goodness-of-Fit Criteria for Quantitative Comparison of Time Signals". In: *Geophysical Journal*

International 178.2 (2009), pp. 813–825. DOI: `10.1111/j.1365-246X.2009.04177.x`.