# Rapid Seismic Waveform Modeling and Inversion With Neural Operators

Yan Yang, Angela F. Gao, Kamyar Azizzadenesheli, Robert W. Clayton, and Zachary E. Ross

***Abstract*— Seismic waveform modeling is a powerful tool for determining earth structure models and unraveling earthquake rupture processes, but it is usually computationally expensive. We introduce a scheme to vastly accelerate these calculations with a recently developed machine learning paradigm called the neural operator. Once trained, these models can simulate a full wavefield at negligible cost. We use a U-shaped neural operator to learn a general solution operator to the 2-D elastic wave equation from an ensemble of numerical simulations performed with random velocity models and source locations. We show that full-waveform modeling with neural operators is nearly two orders of magnitude faster than conventional numerical methods, and more importantly, the trained model enables accurate simulation for velocity models, source locations, and mesh discretization distinctly different from the training dataset. The method also enables convenient full-waveform inversion with automatic differentiation.**

***Index Terms*— Full-waveform inversion, geophysics, machine learning, partial differential equations (PDEs), waveform modeling.**

## I. Introduction

THE seismic wave equation relates displacement fields to external forces and the density and elastic structure in the Earth. Solutions to the wave equation form the basis of ground-shaking simulations of large earthquakes [1], [2], [3] and full-waveform inversion for Earth's structure [4], [5], [6]. Due to the highly heterogeneous nature of the Earth, as exemplified by subduction zones and sedimentary basins, there are no exact analytical solutions for these wavefields. Instead, approximate solutions are made possible by approximating derivatives through discretized spatial and time or frequency domains. Finite-difference methods (FDMs) have been popular since the early 1980 s due to their relatively straightforward formulation [7], [8], [9]. The spectral-element method (SEM), a particular case of finite-element methods (FEMs), which was introduced to seismology in the early 2000 s, combined the flexibility of FEMs with the accuracy of spectral approaches [10], [11], [12], [13]. These numerical solvers impose a trade-off between resolution and computation speed, with the computational cost proportional to the fourth power of frequency [14]. Thus, the cost of wave simulation is a major barrier to using full-waveform techniques for seismic inversion and updating models of the subsurface with new data.

A number of machine learning-based methods have been proposed in the past few years to provide a faster alternative for tackling seismological problems, such as signal denoising [15], [16], [17], event detection [18], [19], [20], and phase association [21], [22]. Deep neural networks have also recently been used to solve partial differential equations (PDEs), such as the Eikonal equation and wave equation [23], [24], [25], [26], [27]. These approaches to solve PDEs offer not only speedup in computational capabilities, but also low-memory overhead, differentiability, and on-demand solutions. Such advantages facilitate deep learning being used for seismic inversion [28], [29], [30], [31], [32]. However, one major limitation of these approaches is that the solutions generated by these models are dependent on the specific spatial and temporal discretization in the numerical simulation training set.

Recently, a paradigm named "neural operator" was developed to address the mesh-dependent shortcoming of classical neural networks by creating a single deep learning model that can be applied to different discretizations [33], [34], [35], [36]. This is made possible because neural operators can provably learn mappings between infinite-dimensional function spaces [37] and, therefore, are suitable for learning general solution operators to PDEs, which are valid even when the PDE coefficients (e.g., elastic properties) are varied. Since first introduced [33], a variety of neural operator models have been developed. In particular, the Fourier neural operator (FNO) is a model that uses the fast Fourier transform as an integral operator, and has been shown to outperform other neural operators in terms of efficiency and accuracy [38]. The FNO has been applied to many types of scientific problems, including weather forecasting [39], $CO_2$ sequestration [40], and coastal flooding [41].

Within the domain of seismology, neural operators were also recently used to learn general solution operators to the 2-D acoustic wave equation, a simplified case of the elastic wave equation [42]. This pilot study demonstrated that it was possible for a single FNO model to predict a complete wave-

Manuscript received 10 October 2022; revised 12 January 2023 and 3 March 2023; accepted 26 March 2023. Date of publication 3 April 2023; date of current version 13 April 2023. *(Corresponding author: Yan Yang.)*

Yan Yang, Robert W. Clayton, and Zachary E. Ross are with the Seismological Laboratory, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: yanyang@caltech.edu; clay@gps.caltech.edu; zross@caltech.edu).

Angela F. Gao is with the Computing and Mathematical Sciences Department, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: afgao@caltech.edu).

Kamyar Azizzadenesheli was with the Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA. He is now with Nvidia, Santa Clara, CA 95051 USA (e-mail: kamyara@nvidia.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TGRS.2023.3264210, provided by the authors.

Digital Object Identifier 10.1109/TGRS.2023.3264210

field given an arbitrary velocity model and mesh discretization. The success of this limited case highlights the potential of these methods; however, extending the method from the acoustic wave equation to the elastodynamic case requires substantially increased model complexity. By comparison with neural networks, FNO is not considered to be a deep architecture and is most analogous to the fully connected neural networks employed heavily until the 2010s. A U-shaped neural operator (U-NO) was recently proposed to enable very deep neural operators and facilitate fast training, data efficiency, and hyperparameter selection robustness [43].

In this article, we apply the U-NO architecture to full-seismic-waveform modeling. We train a U-NO model to learn a general solution operator to the 2-D elastic wave equation and demonstrate that the trained model enables fast and accurate simulation for source locations, velocity structures, and mesh discretization beyond the training dataset. The trained U-NO also allows for efficient full-waveform inversion with automatic differentiation.

## II. Methods

### A. Neural Operator Learning

Operators are maps between function spaces, and the purpose of operator learning is to learn the operator given a dataset of input–output pairs. In seismology, it is common to write solutions to the wave equation, $U(x)$, in terms of a linear integral operator acting on a source function, $A(x)$

$$U(x) = \int G(x, y)A(y)dy \tag{1}$$

where $x \in \mathbb{R}^4$ is the physical domain and $G$ is a so-called Green's function defined for a particular velocity model. Equation (1) holds so long as the velocity model is not varied, because the wave equation remains a linear operator.

Instead, if we consider the case where the input function, $A(x)$, is a velocity model, the solution operator, $L$, relating this to $U(x)$ is nonlinear and cannot be written in the form of (1)

$$U(x) = (\mathcal{L}A)(x).$$

The most general version of the nonlinear solution operator $L$ for the elastic wave equation is not known in closed form.

Neural operators are a class of models that aim to solve this problem, as they provably can learn a wide array of nonlinear operators. Their basic form consists of a composition of linear operators with nonlinear activations. More specifically, a neural operator with $L$ layers can be written as follows:

$$\begin{aligned} v_0(x) &= (PA)(x) \\ v_{l+1}(x) &= \sigma(W_l v_l(x) + \textstyle\int \kappa_l(x, y)v_l(y)dy), \quad l = 0, \ldots, L-1 \\ U(x) &= (Qv_L)(x) \end{aligned} \tag{2}$$

where $v_l$ is the input function at the $l^{\text{th}}$ layer, $P$ is a pointwise operator that lifts the input function to a higher dimensionality, $Q$ is a pointwise operator that projects the function back to the desired output dimensionality, $W_l$ is a linear pointwise transformation that can keep track of nonperiodic boundary behavior, $\sigma$ is a pointwise nonlinear activation operator, and $\kappa_l$ is a kernel function that acts along with the integral as a global linear operator.

A neural operator is parameterized by $P$, $Q$, $W_l$, and $\kappa_l$. A critical aspect of this class of models is that these parameters are independent of the numerical discretization of the physical domain; i.e., they are shared across all possible discretizations in a similar way that in convolutional networks, the parameters are shared across neurons. It is this property that allows for the learning of maps between infinite-dimensional function spaces, as the discretization can be chosen dynamically at inference time independently of what was used for training.

If we are given a dataset of $N$ numerical simulations, $\{A_i, U_i\}_{i=1}^N$, where the values of $A_i$ are chosen to span the range of the expected function space, we can train a neural operator in a supervised fashion to map from arbitrary $A$ into $U$.

Due to the expense of evaluating integral operators, neural operators may lack the efficiency of convolutional or recurrent neural networks in finite-dimensional settings. The FNO was proposed to mitigate this difficulty through the fast Fourier transform [38]. The kernel integral operator in (2) can be considered a convolution operator, defined in Fourier space as follows:

$$\textstyle\int \kappa_l(x, y)v_l(y)dy = \mathcal{F}^{-1}(\mathcal{F}(\kappa_l) \cdot \mathcal{F}(v_l)) \tag{3}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote Fourier transform and its inverse, respectively. However, FNO imposes that each layer is a map between functions spaces with identical domain spaces, which may cause a large memory usage. The U-NO, an analogy to the U-net architectures, was proposed to allow progressively transforming the input function space with respect to a sequence of varying domains [43], [44]. After the lifting operator $P$, a sequence of $L_1$ nonlinear integral operators $G_i$ is applied to $v_0$ and maps the input to a set of functions with decreasing dimensional domain. Then, a sequence of $L_2$ nonlinear integral operators $G_i$ is applied to $v_{L_1+1}$ and maps the input to a set of functions with increasing dimensional domain before the projection operator $Q$. Skip connections [44] are included to add vector-wise concatenation of $v_{L_1+i}$ and $v_{L_1-i}$. The contracting and expanding parts are symmetric. The architecture of the U-NO used in this study is illustrated in Fig. 1, and we refer the interested readers to [33], [34], [35], [36], [38], and [43] for more details.

### B. Numerical Simulation

We set up a training dataset of random source locations, $S$-wave velocity ($V_S$) models, and $P$- to $S$-wave velocity ratios ($V_P/V_S$). We define the velocity model on a 64 × 64 mesh with 0.16-km grid spacing. The source is set as an isotropic explosive source randomly distributed on the mesh. $V_S$ has an average background of 3 km/s and perturbed by random fields with a von Kármán covariance function with the following parameters: Hurst exponent $\kappa = 0.5$, correlation length $a_x = a_y = 8$ grids, and the fractional magnitude of the fluctuation $\varepsilon = 10\%$ background velocity. The power spectral density function of the von Kármán-type random field follows a power law (fractal randomness) and can accurately

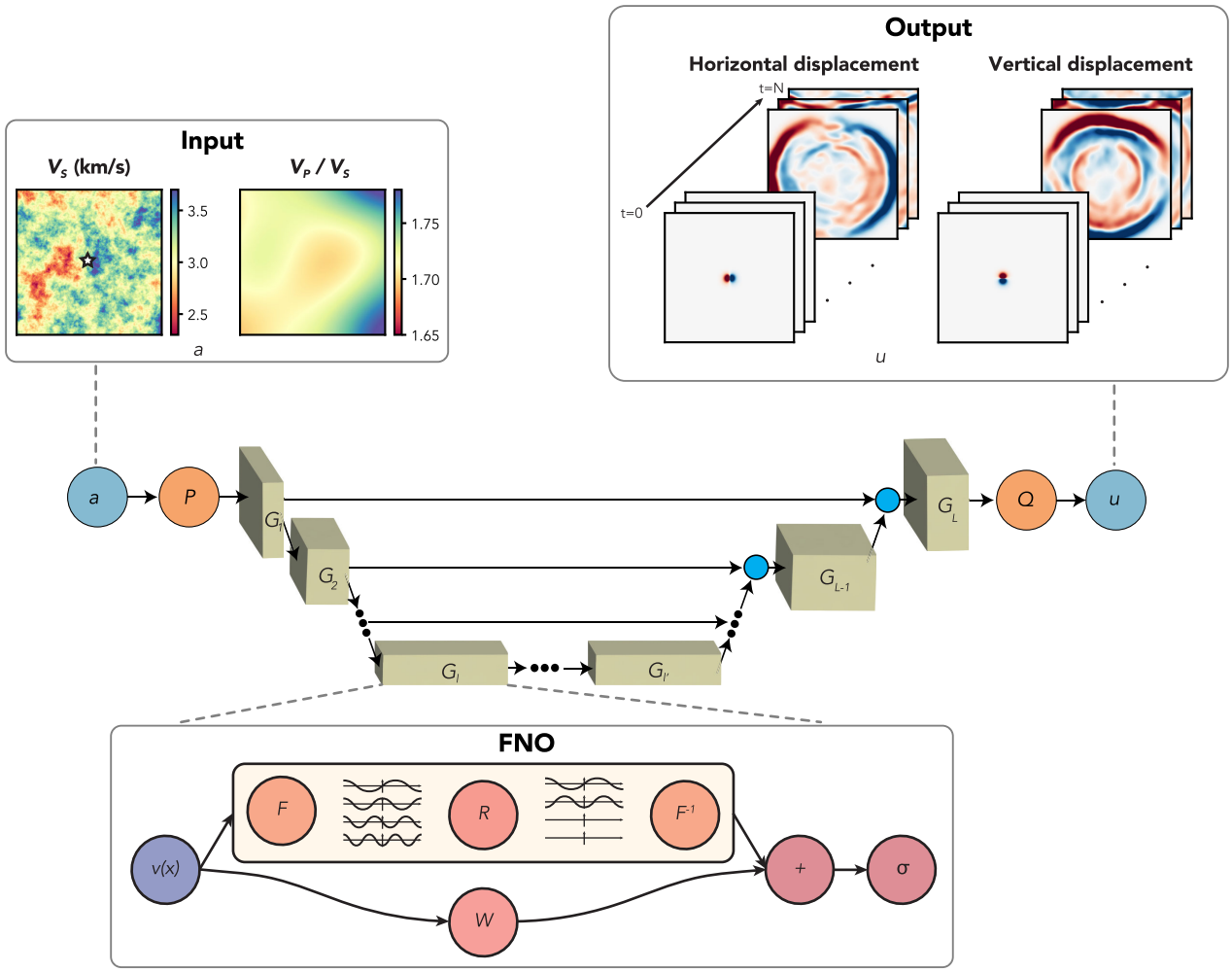



Fig. 1. Overview of our method for solving the elastic wave equation with neural operators. The inputs, $a$, to the U-NO model are the $P$- and $S$-wave velocity ($V_P$ and $V_S$) models and the source location (indicated by the white star). (Top left) $V_P$ is calculated from $V_S$ and $V_P/V_S$, examples of which are shown. (Top right) Outputs, $u$, are the horizontal and vertical displacements at each time step, examples of which are shown. (Middle) U-NO architecture, orange circles $P$ and $Q$ denote point-wise operators, rectangles $G$ denote general operators, and smaller blue circles denote concatenations in function space. (Bottom) Architecture of each FNO layer, where $v$ is the input of the layer, $F$ and $F^{-1}$ are Fourier transform and its inverse, respectively, $R$ and $W$ are a linear transform, and $\sigma$ is the nonlinear activation function. The expansion (or contraction) factors in (5) are set as follows: $c^s_{1,2} = \frac{3}{4}, c^s_{3,4} = \frac{1}{2}, c^s_{5,6} = 2, c^s_{7,8} = \frac{4}{3}, c^t_1 = \frac{3}{4}, c^t_2 = \frac{2}{3}, c^t_{3,4} = \frac{1}{2}, c^t_{5,6} = 2, c^t_7 = \frac{3}{2}, c^t_8 = \frac{4}{3}, c^c_{1,2,3,4} = 2,$ and $c^c_{5,6,7,8} = \frac{1}{2}$..

represent the distribution of Earth's heterogeneity [45]. $V_P/V_S$ is simplified to an average background of 1.732 perturbed by a smooth Gaussian random field with the following parameters: correlation length $\lambda = 32$ grids and standard deviation $\sigma = 2\%$ background. This work, as our very first experiment to evaluate the feasibility of solving 2-D elastic wave equations, wants to focus on the parameters that the wavefield is most sensitive to. Therefore, we use the empirical relation between density and $V_S$ to compute the density [46]. Other input parameters, such as density and attenuation, may be explored in future work. A total of 20 000 random sets of models are generated, and each of them is input to a GPU-based 2-D finite difference code in Cartesian coordinates to simulate the 2-D displacement field [47]. For simulation, the top boundary is set with a free-surface boundary condition, and the other three edges have absorbing boundary conditions. A total of 4-s wavefield with a time step of 0.01 s and a major frequency content up to 6 Hz is simulated. Each simulation takes about 1.23 s with a GPU memory usage of 0.3 GB.

### C. U-NO Model Training

We developed a framework that applies U-NO to the 2-D elastic wave equation. The architecture is depicted schematically in Fig. 1. U-NO takes the source location and $V_P$ and $V_S$ as inputs, where $V_P$ is calculated from $V_S$ and $V_P/V_S$. $V_P$ and $V_S$ are then passed through a point-wise lifting operator. A sequence of nonlinear integral operators (encoders) are applied that gradually contract the physical domain size after each inverse Fourier transform step, while simultaneously increasing the number of channels in the co-domain. These operators are followed by a sequence of nonlinear integral operators (decoders) that progressively expand the physical domain and decrease the number of channels. Finally, a point-wise projection operator leads to the output function [43]. The output of the U-NO model is the complete horizontal and vertical displacement wavefield functions over the medium domain, which can be queried at any mesh points desired, regardless of the input and output training mesh used.

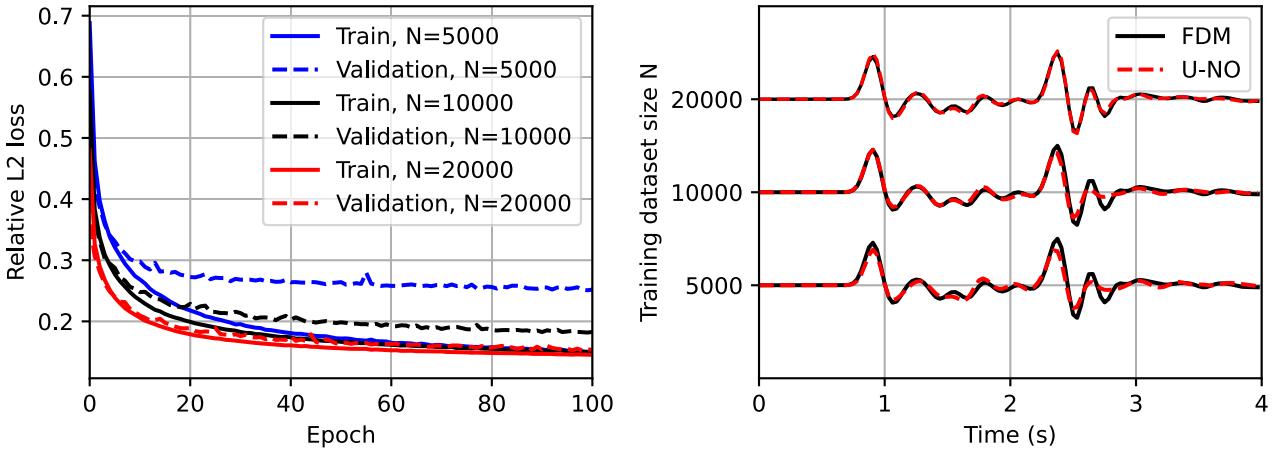



Fig. 2. Model performance as a function of the number of training samples. (Left) Relative $L2$ loss curves for the training and validation data. (Right) Example of simulated waveform comparison between FDM (black solid lines) and U-NO (red dashed lines). $N$ means the number of simulations in the training dataset, including 90% for training and 10% for validation.

We describe the detailed parameters used in U-NO below following the notations in Fig. 1. The goal is to learn an operator mapping from the input function $a$ to the output function $u$. The training is on an input mesh of $X_{\text{in}} \times Y_{\text{in}} \times T_{\text{in}}$ and an output mesh of $X_{\text{out}} \times Y_{\text{out}} \times T_{\text{out}} \times C_{\text{out}}$, where $X_{\text{in}} = Y_{\text{in}} = X_{\text{out}} = Y_{\text{out}} = 64$, $T_{\text{in}} = 3$ representing source, $V_S$, and $V_P/V_S$ distribution on the mesh, $T_{\text{out}} = 128$ for 32-Hz data output, and $C_{\text{out}} = 2$ representing two displacement components (horizontal and vertical). This work applies the U-NO architecture designed for mapping between 3-D spatiotemporal function domains $(x, y, t)$ without any recurrent composition in time [43]. The fourth dimension $C_{\text{out}}$ of the output function $u$ can be created in the last step through the projection operator $Q$. Constructing the operator to learn the mapping between 3-D spatiotemporal function domains

$$G : \left\{a : [0,1]^2 \times [0, T_{\text{in}}] \to \mathbb{R}^{d_A}\right\} \to \left\{u : [0,1]^2 \times [0, T_{\text{out}}] \to \mathbb{R}^{d_U}\right\}. \tag{4}$$

The operators $\{G_i\}_{i=0}^{L}$, as shown in Fig. 1, that are used to construct the U-NO are defined as follows:

$$G_i : \left\{v_i : [0, \alpha_i]^2 \times \mathcal{T}_i \to \mathbb{R}^{d_{vi}}\right\} \to \left\{v_{i+1} : \left[0, c_i^s \alpha_i\right]^2 \times c_i^t \mathcal{T}_i \to \mathbb{R}^{c_i^c d_{vi}}\right\} \tag{5}$$

where $[0, \alpha_i]^2 \times T_i$ is the domain of the input function $v_i$ to the operator $G_i$, and $c_i^s, c_i^t$, and $c_i^c$ are the expansion or contraction factors for the spatial domain, temporal domain, and co-domain for the $i$th operator, respectively. Note that $\mathcal{T}_0 = [0, T_{\text{in}}]$, $\alpha_0 = \alpha_{L+1} = 1$, and $\mathcal{T}_{L+1} = [0, T_{\text{out}}]$. In this work, we set the number of layers to $L = 8$. The details of the expansion and contraction factors $c_i^s, c_i^t$, and $c_i^c$ are in Fig. 1. The lifting operator $P$ to convert the input to a higher dimension channel space is a fully connected neural network with channel number $d_0 = 16$. The projection operator $Q$ to the output domain is also a fully connected neural network. The activation function used in each FNO block is the Gaussian error linear unit (GELU) [48].

With the simulation dataset and the U-NO design, we train the U-NO model in a supervised manner with the objective of learning the general solution operator to the wave equation for arbitrary inputs. We divide the training dataset into 90% training and 10% validation. The model is trained with a batch size of 8. After hyperparameter tuning, the loss function we use in model training is the 90% relative $L1$ loss plus 10% relative $L2$ loss. The incorporation of $L1$-norm loss is more resistant to outliers. We use an Adam optimizer [49] with a learning rate of $10^{-3}$ and a weight decay of $10^{-5}$. We trained for 100 epochs, which takes approximately 40 min/epoch using a single NVIDIA Tesla V100 GPU with 24-GB memory usage. A 70% of loss decrease is achieved in the first ten epochs. Once the U-NO model is trained, the model parameters require GPU usage of 3.8 GB, and the time for an evaluation on a new source and velocity model takes only 0.02 s with GPU usage of 0.9 GB.

## III. Results

### A. Number of Simulations Needed for Training

Once completely trained, the U-NO model can be evaluated on a new input with very little computational cost (0.02 s compared with the FDM runtime of 1.23 s). The number of training simulations is the main factor in the computational cost. In the training process, we split the entire training dataset to 90% for training and 10% for validation. We test the performance of the model on the velocity models out of the training dataset. We can see that the U-NO model trained on a dataset of 5000 simulations can already predict the major phase arrivals, while increasing the dataset size from 5000 simulations to 20 000 provides better fit to the amplitudes (Fig. 2). With a training dataset of 20 000 simulations, the validation and training loss are very close, indicating there is no overfitting of the training data.

### B. Generalizability to Arbitrary Velocity Structure or Discretization

The U-NO model is trained on random velocity models generated with the von Karman correlation function, which can best mimic the Earth's heterogeneous velocity distribution [45], [50]. We show by example that the U-NO model,

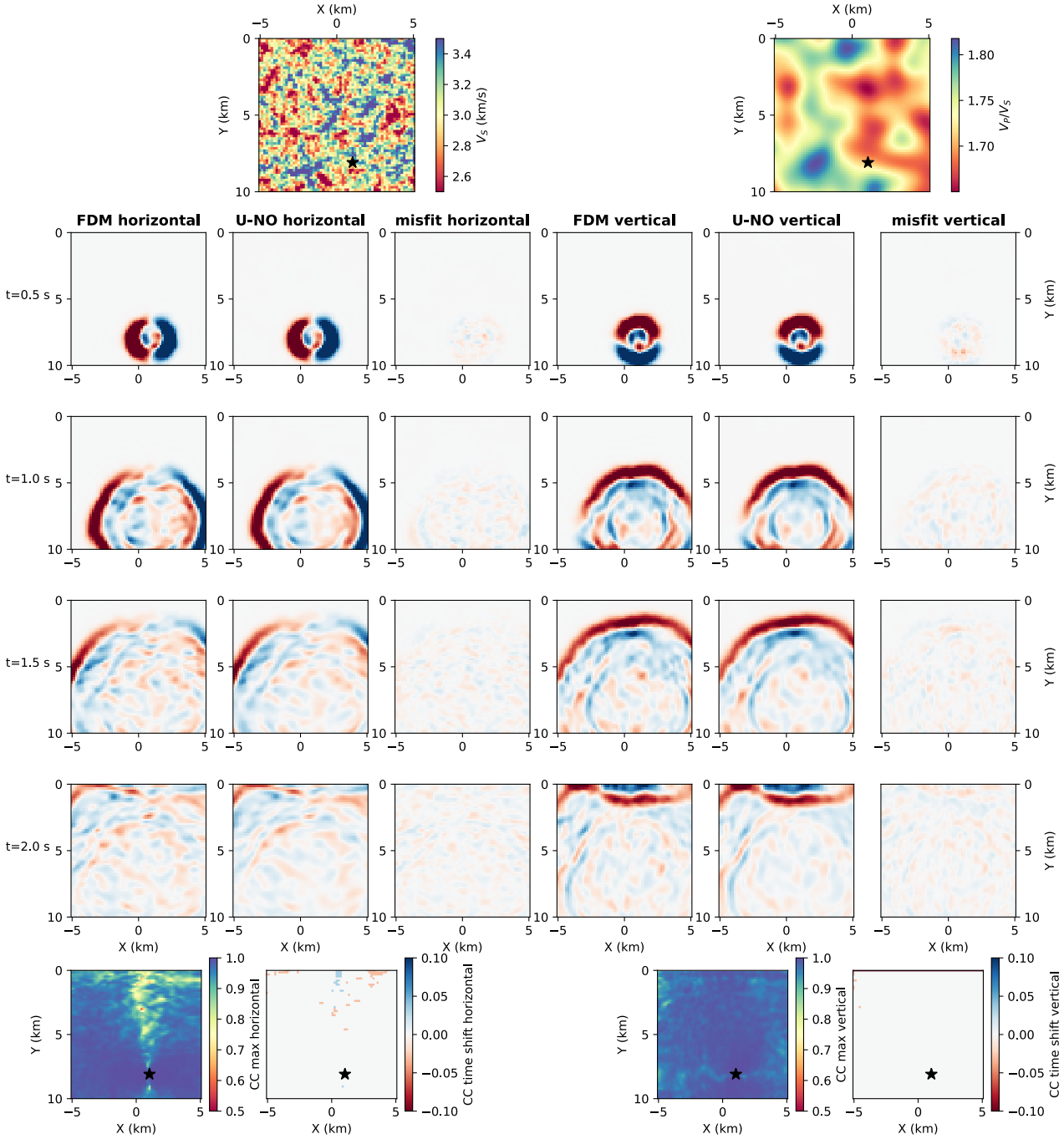



Fig. 3. Model generalization experiments 1: random fields of $V_S$ and $V_P/V_S$ model with four-times roughness of the training data. The top row shows $V_S$ and $V_P/V_S$. From the second row to the fifth row, the wavefield snapshots at 0.5–2.0 s are shown. From left to right, the first three columns show the horizontal displacement of the FD simulation, U-NO prediction, and their misfit in the same color scale. The latter three columns show the vertical component. The horizontal and vertical displacement waveforms at each grid are cross correlated between FD and U-NO, with the maximum cross-correlation value and its associated time shift shown in the bottom row. For this case, the relative $L2$ loss of the U-NO simulation is 0.182.

although trained on random velocity models with some certain parameters, is applicable to arbitrary velocity models. These outcomes are, in fact, expected from theoretical grounds, because most physical functions can be approximated to arbitrary accuracy by random fields.

Our first example is with velocity models from a von Karman-type random distribution, but with a different covariance function than the one used for the training data. We increase the roughness of the velocity structure by a factor of 4 by decreasing the correlation length of $V_S$ and $V_P/V_S$ to only one-fourth that of the training data. As shown in Fig. 3, the wavefield snapshot has more coda than with the smoother models because of the scattering from increased heterogeneity. However, the coda waves are well modeled by U-NO when compared with the ground-truth simulation by FDM.

The velocity models used in the training data do not have coherent structures with discontinuities as in the real Earth, but wavefields for such models can still be simulated with

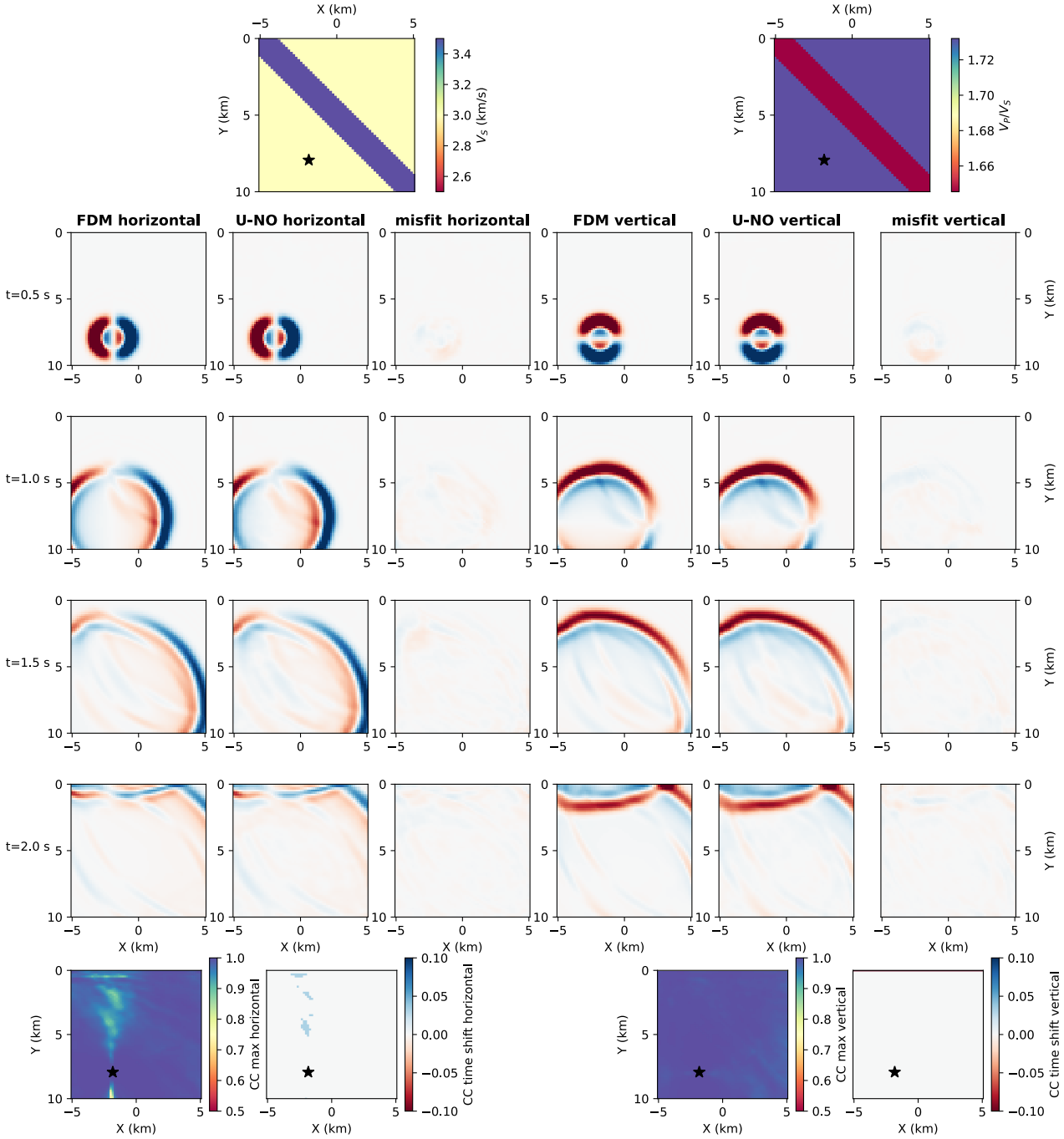



Fig. 4. Model generalization experiments 2: similar to Fig. 3, but for a homogeneous background model embedded with a "slab" with 20% higher $V_S$ and 5% lower $V_P/V_S$. Relative $L2$ loss of the U-NO simulation is 0.090.

our method. As mentioned before, this is because discontinuous functions can be approximated to arbitrary accuracy by random fields. Fig. 4 shows a simple model with a dipping "slab" embedded in a homogeneous background. The slab has 20% higher $V_S$ and 5% lower $V_P/V_S$. The wavefield snapshots show that the reflections from the high velocity anomaly are clearly predicted by U-NO. A more complex example is shown in Fig. 5, where a random subpanel of the Marmousi model, a 2-D velocity model with complex vertical and horizontal structures used in exploration studies [51], is used. The reflected and refracted waves are very complicated due to the presence of folding and faulting, but the U-NO predictions still closely approximate the numerical solutions (Fig. 5).

One of the most important advantages of a neural operator compared with a neural network is its mesh-free nature, since it intrinsically learns the mapping between function spaces. A model trained on a particular mesh can be evaluated on any other mesh, even at finer spacing. The Fourier layers may learn from and evaluate functions on any discretization, because parameters are directly learned in Fourier space, and resolving the functions in physical space is simply projecting on the

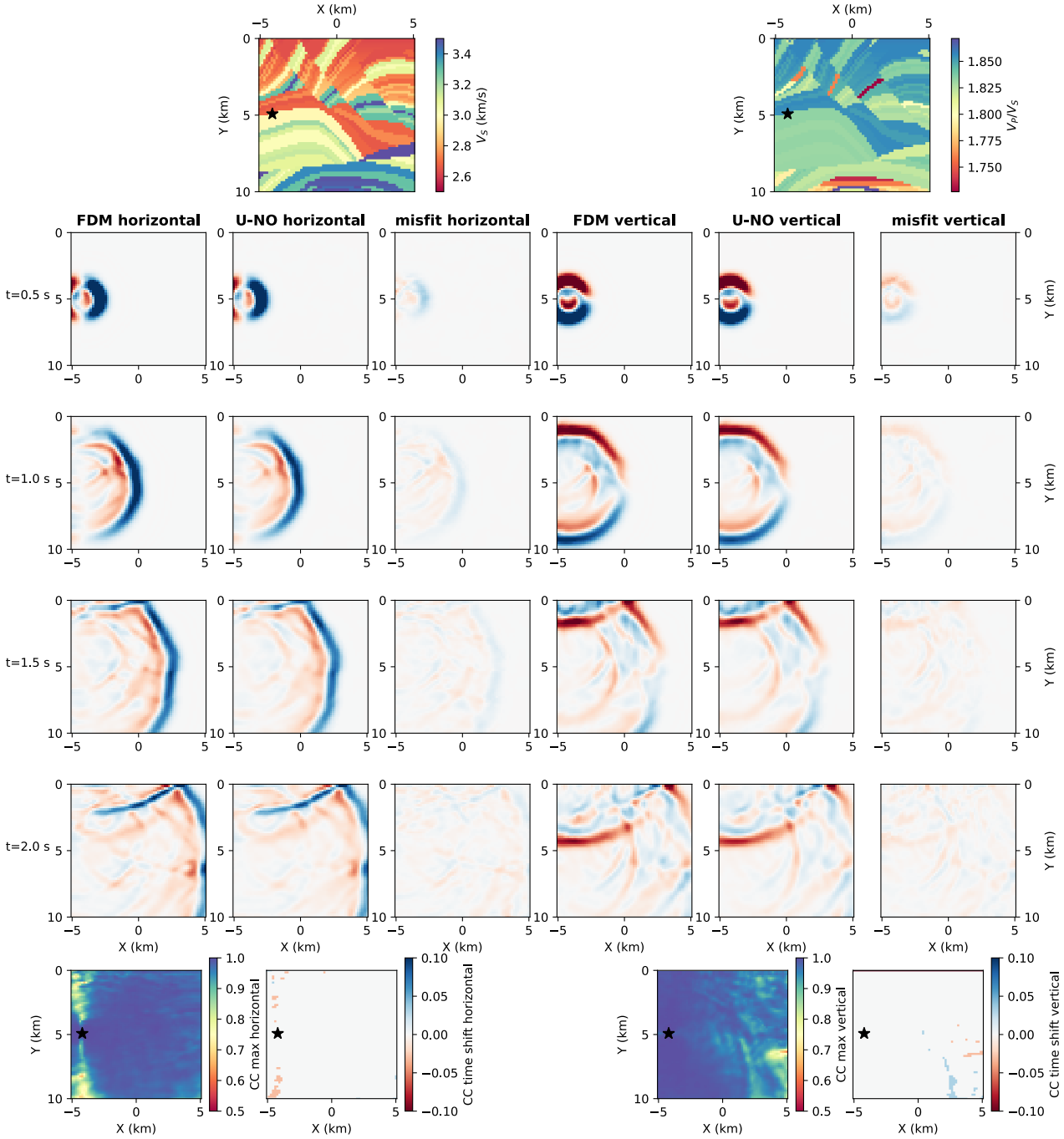



Fig. 5. Model generalization experiments 3: similar to Fig. 3, but for a random subpanel from the Marmousi model. The velocity perturbation range is normalized to 30% of the average velocity. Relative $L2$ loss of the U-NO simulation is 0.225.

basis [38]. The example in Fig. 6 shows the U-NO trained on a grid of $64 \times 64$ nodes applied to an input velocity model with $160 \times 160$ nodes. Here, both the input velocity model and the output wavefield can be seen at a much higher resolution, yet U-NO provides comparable prediction with the FDM solver. Note that if the resolution is increased by a factor of 2, a grid-based numerical solver, such as FDM, takes about six times greater computational time; however, the evaluation using U-NO takes only about 2.5 times longer, providing additional computational efficiency.

We evaluate the overall generalization performance of the trained U-NO by performing one thousand random realizations on each of these cases. The distribution of the relative $L2$-norm misfit and cross-correlation coefficient are plotted in Fig. 7. In the case of the Marmousi model, the extended tail of the histogram is attributed to the model's imbalanced complexity. In general, however, we see a very high cross-correlation coefficient ($>0.95$) between U-NO prediction and ground truth, confirming its robust generalizability.

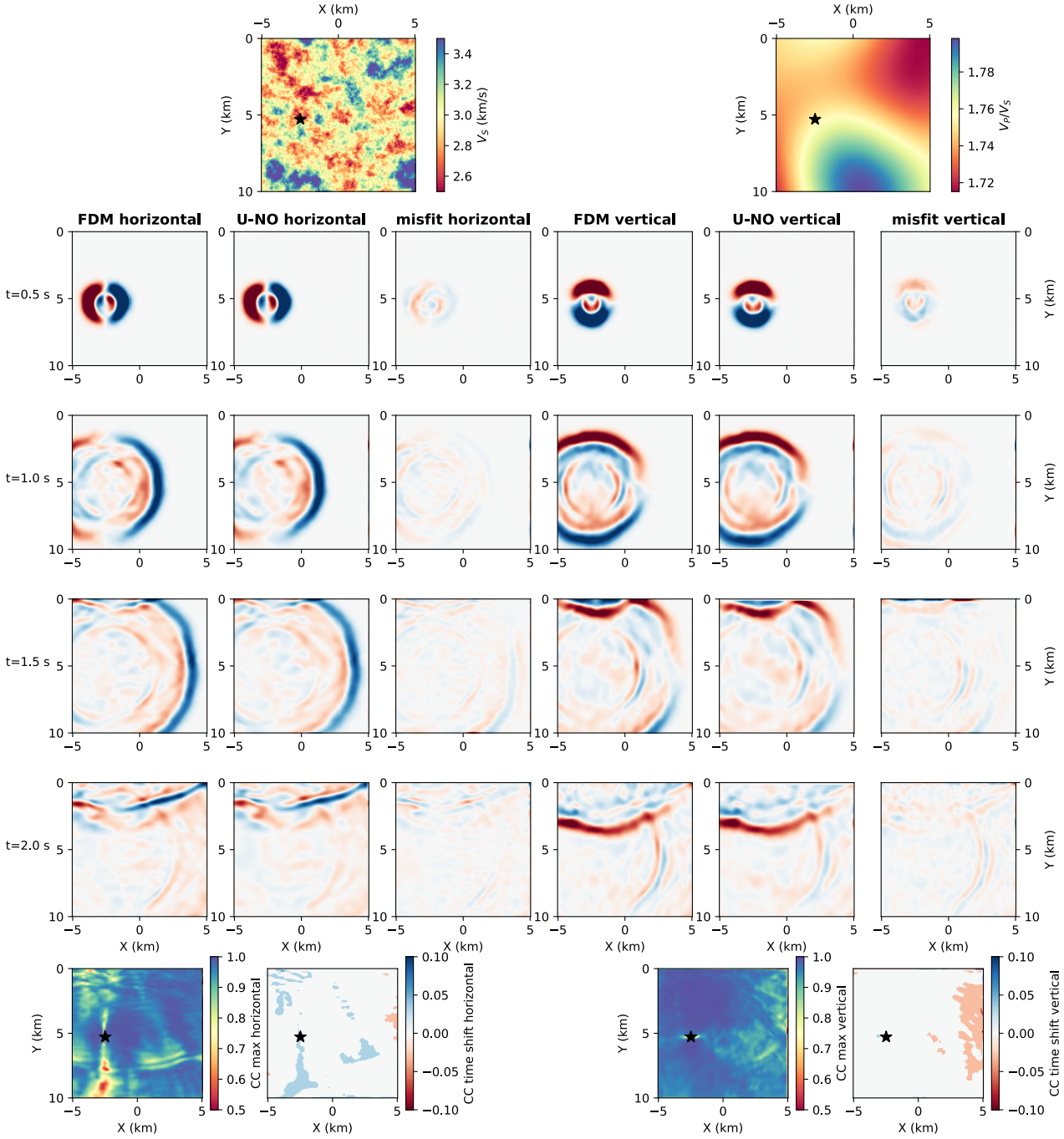



Fig. 6. Model generalization experiments 4: similar to Fig. 3, but $V_P$ and $V_S$ model mesh discretization is increased from 64 × 64 to 160 × 160. Relative $L2$ loss of the U-NO simulation is 0.385.

### C. Application to Full-Waveform Inversion

One of the most important applications of wavefield simulations is in full-waveform inversion (FWI), which uses the full recorded waveform to image the Earth's interior. The adjoint-state method is the traditional approach for computing the gradients of an objective function with respect to parameters of interest [4], [5]. Neural operators are differentiable by design, which enables gradient computation with reverse-mode automatic differentiation. It has been shown that automatic differentiation and the adjoint approach are mathematically equivalent [28]. Hence, the trained U-NO model allows for convenient FWI, and the associated speed and accuracy should depend only on the forward modeling part.

We demonstrate the inversion performance using the velocity structure of random subpanels in the Marmousi model [51]. The synthetic waveform data are simulated with FDM [47] using 14 sources distributed in a ring shape. In Fig. 8, we use the true source location, receivers on all 64 × 64 grids, and noise-free waveform data; the goal here is not to demonstrate resolution, but rather the computational accuracy of the method. We then invert for $V_P$ and $V_S$ simultaneously by starting with homogeneous initial $V_P$ and $V_S$ models and forward propagating the wavefield with the

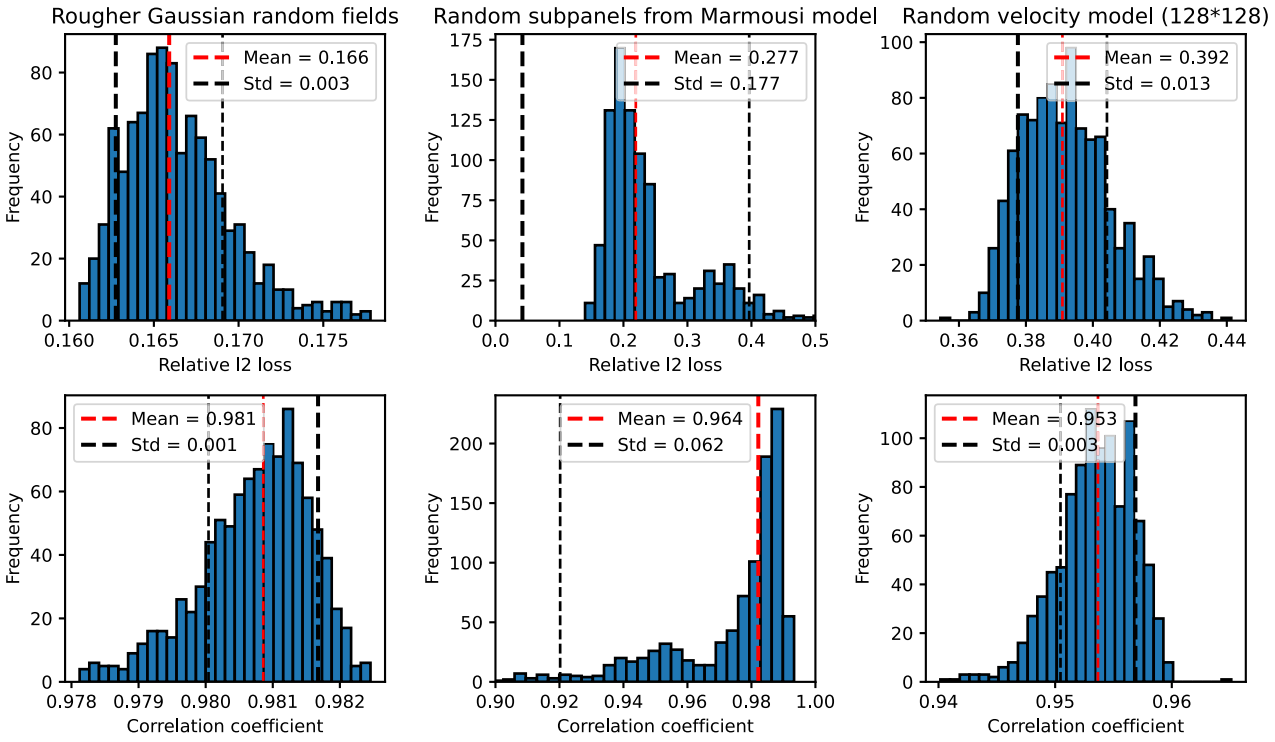



Fig. 7. Output evaluation. (Top) Distribution of relative $L2$ loss and (bottom) correlation coefficient between the U-NO predictions and ground truth. From left to right, the columns are corresponding to the experiments in Figs. 3, 5, and 6. The red and black dashed lines mark the mean and standard deviation of the histograms.

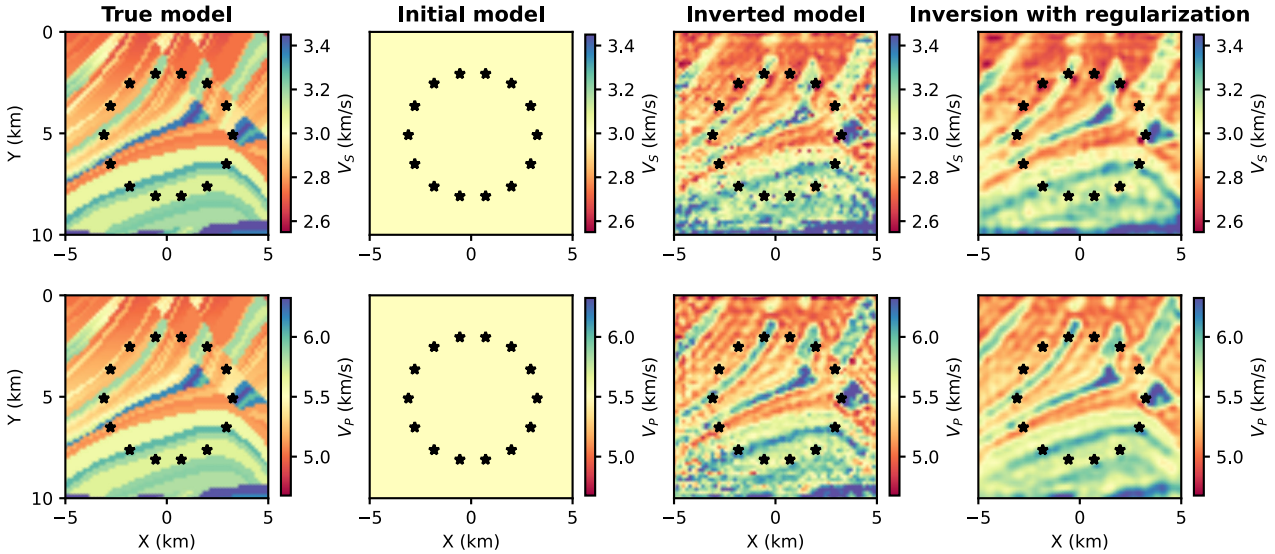



Fig. 8. Full-waveform inversion. The inversion for a random subpanel from the Marmousi model; 14 sources are placed in a ring shape (black stars), and receivers are placed at every node of the 64 × 64 grid. From left to right, the columns represent true velocity model, initial model for inversion, inverted model without regularization, and inverted model with the zeroth- and first-order Tikhonov regularization. The top and bottom rows are the models for $V_S$ and $V_P$, respectively.

U-NO for each source. The misfit is defined by the mean-square error between the forward modeled and true wavefield. The gradient of the misfit with respect to $V_P$ and $V_S$ can be computed through automatic differentiation. $V_P$ and $V_S$ are then iteratively updated with gradient descent for 100 iterations using the Adam optimizer [49] with a learning rate of 0.01. Each iteration takes only about 1.4 s by taking advantage of U-NO forward computation. The results in Fig. 8 show a relative $L2$-norm misfit between the true and inverted model of only 3%. This successful inversion, in turn, further validates the accuracy of forward modeling with U-NO.

Besides the fact that the inversion target velocity model is quite different from the smooth random fields in the training dataset, this experiment itself is difficult due to conventional problems in full-waveform inversion, such as cycle skipping (multiple local maxima in the least-squares misfit function).

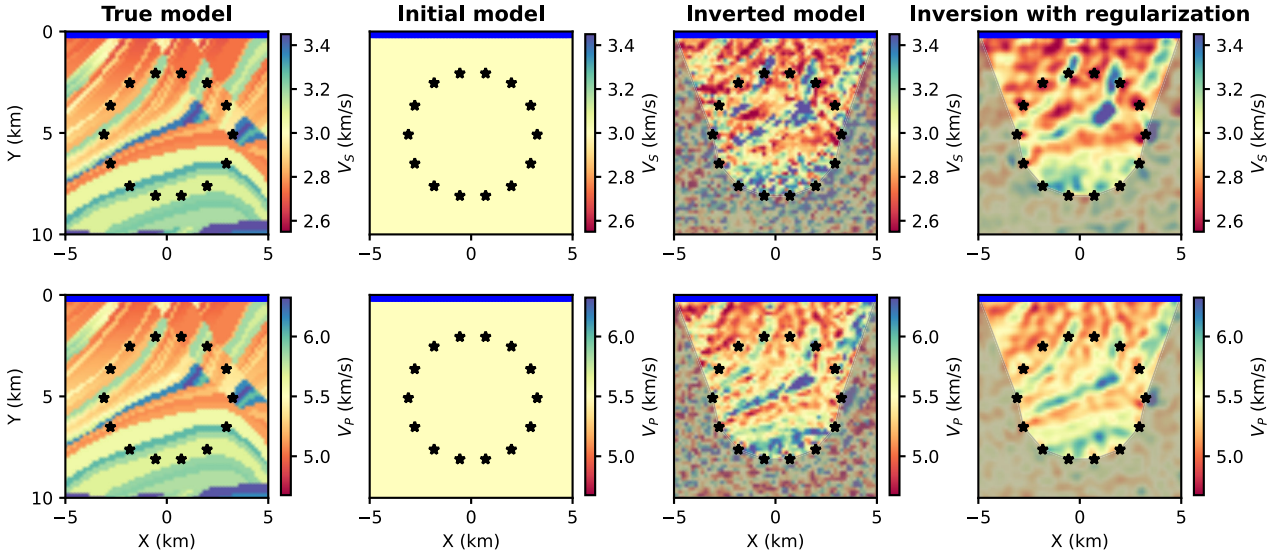



Fig. 9. Full-waveform inversion. Similar to Fig. 8, but the receivers are only placed on the 64 grids on the surface (blue line on the top of each subpanel). The gray shaded areas mask the areas without ray path coverage.

We also show that if we only use 64 receivers on the surface, the inversion results in the region with ray path coverage are still reasonably accurate (Fig. 9). Inversion with a biased homogenous initial model is also capable of producing relatively accurate results (see Supplementary Material).

## IV. Discussion and Conclusion

We use the relative $L2$ loss between the FDM and U-NO predictions to evaluate the performance of the trained model for generalization. The relative $L2$ loss is defined as the $L2$-norm of the difference between the prediction and ground truth divided by the $L2$-norm of the ground truth. This ratio is used to evaluate the performance of the trained model. When using the same mesh discretization as the training data, the relative $L2$ loss is around 10%–20%, but this number rises to 30%–40% when the tests are performed on finer grids (Fig. 7). These values are misleading, however, because the relative $L2$ loss imposes equal weights to the entire sparse matrix of waveforms that is dominated by small amplitudes close to 0. Alternatively, the cross-correlation coefficient is a quantity that is more sensitive to the seismic phases with amplitudes larger than background noise. A cross-correlation coefficient larger than 0.95 suggests that the coherence of the U-NO prediction is excellent, even for the scenarios with large relative $L2$ loss (Fig. 7). In addition, the FWI results confirm that the large $L2$ loss is not so important, since even challenging models can still be properly recovered (Fig. 8).

Besides the more than an order of magnitude higher speed, the most important advantage of the neural operator-based full-waveform modeling is its generalizability to arbitrary velocity models or discretization. This is because the neural operator learns a general solution operator to the wave equation instead of a specific instance of input velocity models. Once the neural operator is trained, it can be used by the entire seismology community for any region of a similar size without the need for retraining. Since the full-waveform modeling with a neural operator has easily accessible gradients for convenient FWI, we anticipate that this approach will eventually make FWI as affordable as travel time tomography.

One of the main limitations of the method is the domain extent. For a trained neural operator, the function is defined on a fixed domain extent (e.g., it could be a unit cube). We can evaluate at a different grid size but cannot change the extent. We are now working on an extension of the work, where we recursively predict the wavefield. Through this way, a trained neural operator is essentially taking the first few time steps as input and then output the next few time steps, and there will be no need for retraining.

The scalability of evaluation using a trained neural operator with respect to the grid size and the number of time steps is a little different from conventional FDM. Assuming the original dimension is $(Nx_1, Ny_1, Nt_1)$, where $Nx_1$ and $Ny_1$ are the number of grids in the $x$ and $y$ domain, respectively, and $Nt_1$ is the number of time steps. If the new dimension is $(Nx_2, Ny_2, Nt_2)$, the memory becomes $(Nx_2 \cdot Ny_2 \cdot Nt_2)/(Nx_1 \cdot Ny_1 \cdot Nt_1)$ times the original memory, which is consistent with the FDM. In the example presented in this article, evaluation using U-NO takes three times the GPU memory of the FDM approach, and this scaling should be consistent with increasing grid points. In terms of computational cost, the majority of it for UNO is on the Fourier transform and its inverse. The computational cost of fast Fourier transform with dimension $(Nx_1, Ny_1, Nt_1)$ is proportional to $Nx_1 \cdot Ny_1 \cdot Nt_1 \cdot \log(Nx_1 \cdot Ny_1 \cdot Nt_1)$, and therefore, the new computational time becomes $(Nx_2 \cdot Ny_2 \cdot Nt_2 \cdot \log(Nx_2 \cdot Ny_2 \cdot Nt_2))/(Nx_1 \cdot Ny_1 \cdot Nt_1 \cdot \log(Nx_1 \cdot Ny_1 \cdot Nt_1))$ times the original computational time. This scaling is slightly higher than that of FDM; however, considering the 60 times acceleration in the example presented in this article, UNO evaluation on an increased dimension of $1024 \times 1024 \times 1024$ should still have $\approx$40 times the acceleration.

The most compute- and memory-intensive part of the UNO method is the one-time training process. The cost of training for the 2-D case is tractable on a single GPU. For the extension

from 2-D to 3-D modeling, the computation and memory will increase due to the larger dataset and the larger number of parameters to learn. Therefore, the next step is to enhance data compression and parallelization to accelerate the training process and reduce the storage. Since this is a learning-based approach, the model performance can be improved by fine-tuning the model parameters and increasing the size of the training dataset. More importantly, any future advancements made in neural operator model architectures will be able to be directly incorporated into the system as they occur. For example, the improvement from linear layers of FNO to U-NO enables faster training convergence. As a result, we should only take current performance metrics as a starting point.

## REFERENCES


[1] R. W. Graves and A. Pitarka, "Broadband ground-motion simulation using a hybrid approach," *Bull. Seismol. Soc. Amer.*, vol. 100, no. 5A, pp. 2095–2123, Oct. 2010, doi: 10.1785/0120100057.

[2] A. J. Rodgers, N. A. Petersson, A. Pitarka, D. B. McCallen, B. Sjogreen, and N. Abrahamson, "Broadband (0–5 Hz) fully deterministic 3D ground-motion simulations of a magnitude 7.0 Hayward fault earthquake: Comparison with empirical ground-motion models and 3D path and site effects from source normalized intensities," *Seismol. Res. Lett.*, vol. 90, no. 3, pp. 1268–1284, May 2019, doi: 10.1785/0220180261.

[3] R. Graves and A. Pitarka, "Kinematic ground-motion simulations on rough faults including effects of 3D stochastic velocity perturbations," *Bull. Seismol. Soc. Amer.*, vol. 106, no. 5, pp. 2136–2153, Oct. 2016, doi: 10.1785/0120160088.

[4] A. Fichtner, B. L. N. Kennett, H. Igel, and H.-P. Bunge, "Full seismic waveform tomography for upper-mantle structure in the Australasian region using adjoint methods," *Geophys. J. Int.*, vol. 179, no. 3, pp. 1703–1725, Dec. 2009, doi: 10.1111/J.1365-246X.2009.04368.X.

[5] C. Tape, Q. Liu, A. Maggi, and J. Tromp, "Adjoint tomography of the southern California crust," *Science*, vol. 325, no. 5943, pp. 988–992, Aug. 2009, doi: 10.1126/SCIENCE.1175298.

[6] L. Gebraad, C. Boehm, and A. Fichtner, "Bayesian elastic full-waveform inversion using Hamiltonian Monte Carlo," *J. Geophys. Res., Solid Earth*, vol. 125, no. 3, Mar. 2020, Art. no. e2019JB018428, doi: 10.1029/2019JB018428.

[7] K. B. Olsen, "Site amplification in the Los Angeles basin from three-dimensional modeling of ground motion," *Bull. Seismol. Soc. Amer.*, vol. 90, no. 6B, pp. S77–S94, Dec. 2000, doi: 10.1785/0120000506.

[8] K. R. Kelly, R. W. Ward, S. Treitel, and R. M. Alford, "Synthetic seismograms: A finite-difference approach," *Geophysics*, vol. 41, no. 1, pp. 2–27, Feb. 1976, doi: 10.1190/1.1440605.

[9] H. Igel, T. Nissen-Meyer, and G. Jahnke, "Wave propagation in 3D spherical sections: Effects of subduction zones," *Phys. Earth Planet. Interiors*, vol. 132, nos. 1–3, pp. 219–234, Sep. 2002, doi: 10.1016/S0031-9201(02)00053-5.

[10] A. Fichtner, H. Igel, H. P. Bunge, and B. L. N. Kennett, "Simulation and inversion of seismic wave propagation on continental scales based on a spectral-element method," *J. Numer. Anal., Ind. Appl. Math.*, vol. 4, nos. 1–2, pp. 11–22, 2009.

[11] Q. Liu and Y. J. Gu, "Seismic imaging: From classical to adjoint tomography," *Tectonophysics*, vols. 566–567, pp. 31–66, Sep. 2012, doi: 10.1016/J.TECTO.2012.07.006.

[12] D. Komatitsch and J. Tromp, "Spectral-element simulations of global seismic wave propagation—I. Validation," *Geophys. J. Int.*, vol. 149, no. 2, pp. 390–412, May 2002, doi: 10.1046/j.1365-246X.2002.01653.x.

[13] D. Komatitsch and J. Tromp, "Spectral-element simulations of global seismic wave propagation—II. Three-dimensional models, oceans, rotation and self-gravitation," *Geophys. J. Int.*, vol. 150, no. 1, pp. 303–318, Jul. 2002, doi: 10.1046/j.1365-246X.2002.01716.x.

[14] O. Pell, J. Bower, R. Dimond, O. Mencer, and M. J. Flynn, "Finite-difference wave propagation modeling on special-purpose dataflow machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 5, pp. 906–915, May 2013, doi: 10.1109/TPDS.2012.198.

[15] W. Zhu, A. C. Bovik, S. M. Mousavi, and G. C. Beroza, "Seismic signal denoising and decomposition using deep neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9476–9488, Nov. 2019, doi: 10.1109/TGRS.2019.2926772.

[16] L. Yang, X. Liu, W. Zhu, L. Zhao, and G. C. Beroza, "Toward improved urban earthquake monitoring through deep-learning-based noise suppression," *Sci. Adv.*, vol. 8, no. 15, p. 3564, Apr. 2022, doi: 10.1126/SCIADV.ABL3564.

[17] C. Birnie, M. Ravasi, S. Liu, and T. Alkhalifah, "The potential of self-supervised networks for random noise suppression in seismic data," *Artif. Intell. Geosci.*, vol. 2, pp. 47–59, Dec. 2021, doi: 10.1016/j.aiig.2021.11.001.

[18] W. Zhu and G. C. Beroza, "PhaseNet: A deep-neural-network-based seismic arrival-time picking method," *Geophys. J. Int.*, vol. 216, no. 1, pp. 261–273, 2019, doi: 10.1093/gji/ggy423.

[19] S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, "Earthquake transformer—An attentive deep-learning model for simultaneous earthquake detection and phase picking," *Nature Commun.*, vol. 11, no. 1, pp. 1–12, Aug. 2020, doi: 10.1038/s41467-020-17591-w.

[20] Z. E. Ross, M.-A. Meier, and E. Hauksson, "P wave arrival picking and first-motion polarity determination with deep learning," *J. Geophys. Res., Solid Earth*, vol. 123, pp. 5120–5129, Jun. 2018, doi: 10.1029/2017JB015251.

[21] Z. E. Ross, Y. Yue, M. Meier, E. Hauksson, and T. H. Heaton, "PhaseLink: A deep learning approach to seismic phase association," *J. Geophys. Res., Solid Earth*, vol. 124, no. 1, pp. 856–869, Jan. 2019, doi: 10.1029/2018JB016674.

[22] W. Zhu, K. S. Tai, S. M. Mousavi, P. Bailis, and G. C. Beroza, "An end-to-end earthquake detection method for joint phase picking and association using deep learning," *J. Geophys. Res., Solid Earth*, vol. 127, no. 3, pp. 1–13, Mar. 2022, doi: 10.1029/2021JB023283.

[23] A. Siahkoohi, M. Louboutin, and F. J. Herrmann, "Neural network augmented wave-equation simulation," Sep. 2019, *arXiv:1910.00925*.

[24] B. Moseley, T. Nissen-Meyer, and A. Markham, "Deep learning for fast simulation of seismic waves in complex media," *Solid Earth*, vol. 11, no. 4, pp. 1527–1549, Aug. 2020, doi: 10.5194/SE-11-1527-2020.

[25] B. Moseley, A. Markham, and T. Nissen-Meyer, "Fast approximate simulation of seismic waves with deep learning," Jul. 2018, *arXiv:1807.06873*.

[26] B. Moseley, A. Markham, and T. Nissen-Meyer, "Finite basis physics-informed neural networks (FBPINNs): A scalable domain decomposition approach for solving differential equations," Jul. 2021, *arXiv:2107.07871*.

[27] J. D. Smith, K. Azizzadenesheli, and Z. E. Ross, "EikoNet: Solving the Eikonal equation with deep neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10685–10696, Dec. 2021, doi: 10.1109/TGRS.2020.3039165.

[28] W. Zhu, K. Xu, E. Darve, and G. C. Beroza, "A general approach to seismic inversion with automatic differentiation," *Comput. Geosci.*, vol. 151, Jun. 2021, Art. no. 104751, doi: 10.1016/J.CAGEO.2021.104751.

[29] B. Sun and T. Alkhalifah, "ML-descent: An optimization algorithm for full-waveform inversion using machine learning," *Geophysics*, vol. 85, no. 6, pp. R477–R492, Nov. 2020, doi: 10.1190/geo2019-0641.1.

[30] X. Zhang and A. Curtis, "Bayesian geophysical inversion using invertible neural networks," *J. Geophys. Res., Solid Earth*, vol. 126, no. 7, Jul. 2021, Art. no. e2021JB022320, doi: 10.1029/2021JB022320.

[31] M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions," *J. Geophys. Res., Solid Earth*, vol. 127, no. 5, pp. 1–21, May 2022, doi: 10.1029/2021jb023120.

[32] V. Kazei, O. Ovcharenko, P. Plotnitskii, D. Peter, X. Zhang, and T. Alkhalifah, "Mapping full seismic waveforms to vertical velocity profiles by deep learning," *Geophysics*, vol. 86, no. 5, pp. R711–R721, Sep. 2021, doi: 10.1190/geo2019-0473.1.

[33] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Mach. Intell.*, vol. 3, no. 3, pp. 218–229, Mar. 2021, doi: 10.1038/s42256-021-00302-5.

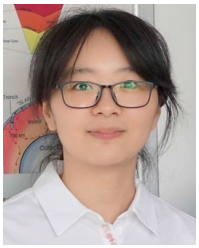
[34] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, "Model reduction and neural networks for parametric PDEs," *SMAI J. Comput. Math.*, vol. 7, pp. 121–157, May 2020, doi: 10.5802/smai-jcm.74.

[35] Z. Li et al., "Multipole graph neural operator for parametric partial differential equations," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2020, pp. 6755–6766.

[36] Z. Li et al., "Neural operator: Graph kernel network for partial differential equations," Mar. 2020, *arXiv:2003.03485*.

[37] N. Kovachki et al., "Neural operator: Learning maps between function spaces," Dec. 2021, *arXiv:2108.08481*. Accessed: Aug. 19, 2022.

[38] Z. Li et al., "Fourier neural operator for parametric partial differential equations," Oct. 2020, *arXiv:2010.08895*.
[39] J. Pathak et al., "FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators," Feb. 2022, *arXiv:2202.11214*.
[40] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, and S. M. Benson, "U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow," *Adv. Water Resour.*, vol. 163, May 2022, Art. no. 104180, doi: 10.1016/J.ADVWATRES.2022.104180.
[41] P. Jiang et al., "Digital twin Earth—Coasts: Developing a fast and physics-informed surrogate model for coastal floods via neural operators," Oct. 2021, *arXiv:2110.07100*.
[42] Y. Yang, A. F. Gao, J. C. Castellanos, Z. E. Ross, K. Azizzadenesheli, and R. W. Clayton, "Seismic wave propagation and inversion with neural operators," *Seismic Rec.*, vol. 1, no. 3, pp. 126–134, Oct. 2021, doi: 10.1785/0320210026.
[43] M. A. Rahman, Z. E. Ross, and K. Azizzadenesheli, "U-NO: U-shaped neural operators," Apr. 2022, *arXiv:2204.11127*.
[44] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9351, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.
[45] N. Nakata and G. C. Beroza, "Stochastic characterization of mesoscale seismic velocity heterogeneity in Long Beach, California," *Geophys. J. Int.*, vol. 203, no. 3, pp. 2049–2054, Dec. 2015, doi: 10.1093/gji/ggv421.
[46] T. M. Brocher, "Empirical relations between elastic wavespeeds and density in the Earth's crust," *Bull. Seismol. Soc. Amer.*, vol. 95, no. 6, pp. 2081–2092, Dec. 2005, doi: 10.1785/0120050077.
[47] D. Li, D. Helmberger, R. W. Clayton, and D. Sun, "Global synthetic seismograms using a 2-D finite-difference method," *Geophys. J. Int.*, vol. 197, no. 2, pp. 1166–1183, May 2014, doi: 10.1093/gji/ggu050.
[48] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," Jun. 2016, *arXiv:1606.08415*.
[49] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[50] P. M. Mai and G. C. Beroza, "A spatial random field model to characterize complexity in earthquake slip," *J. Geophys. Res., Solid Earth*, vol. 107, no. B11, pp. 10-1–10-21, Nov. 2002, doi: 10.1029/2001JB000588.
[51] R. Versteeg, "The Marmousi experience: Velocity model determination on a synthetic complex data set," *Lead. Edge*, vol. 13, no. 9, pp. 927–936, Sep. 1994, doi: 10.1190/1.1437051.



**Yan Yang** is currently pursuing the Ph.D. degree with the Seismological Laboratory, California Institute of Technology, Pasadena, CA, USA.

Her research interests focus on seismic imaging and monitoring of subsurface.

**Angela F. Gao** is currently pursuing the Ph.D. degree with the Computing and Mathematical Sciences Department, California Institute of Technology, Pasadena, CA, USA.

She is interested in inverse problems, computational photography, generative models, and deep learning, with applications in scientific imaging problems.

**Kamyar Azizzadenesheli** is a Senior Research Scientist at Nvidia, Santa Clara, CA, USA. Prior to his role at Nvidia, he was an Assistant Professor at the Department of Computer Science, Purdue University, West Lafayette, IN, USA. Prior to his Faculty position, he was at the California Institute of Technology, Pasadena, CA, USA, as a Post-Doctoral Scholar at the Department of Computing and Mathematical Sciences.

**Robert W. Clayton** is a Professor of geophysics at the California Institute of Technology, Pasadena, CA, USA, where he works in the areas of seismic wave propagation, earth structure, and tectonics. He has applied imaging methods to the Los Angeles region and to subduction zones around the world.

**Zachary E. Ross** is an Assistant Professor of geophysics with the California Institute of Technology, Pasadena, CA, USA, where he uses machine learning and signal processing techniques to better understand earthquakes and fault zones. He is interested in the dynamics of seismicity, earthquake source properties, and fault-zone imaging.