Learning to solve elastic wave equation with the Clifford Fourier neural operator

Tianze Zhang, Kris Innanen, Daniel Trad

ABSTRACT

Neural operators are extensions of neural networks which in supervised training learn how to map complex relationships, such as classes of PDE. Recent literature reports efforts to develop one type of these, the Fourier Neural Operator (FNO), such that it learns to create relatively general solutions to PDEs such as the Navier-Stokes equation. Clifford algebra is very useful for representing multidimensional data. In this study, we use the Clifford Fourier Neural Operator (CFNO) be trained to learn the elastic wave equation from a synthetic training data set. CFNO attempts to find a manifold for elastic wave propagation. On that manifold, wave fields are represented in lower dimensions than those needed for standard solutions, and the calculations for wave propagation are correspondingly simpler. The CFNO combines a linear Clifford fully connected transform, the Clifford Fourier transform, and a non-linear local activation to produce a network with sufficient freedom to map from a general parameterization of a forward wave problem to its solution. Post-training, the CFNO is observed to generate accurate elastic wave fields.

INTRODUCTION

Clifford algebras serve as a fundamental nexus where geometry melds with algebra, originally conceived to streamline the spatial and geometric interrelationships spanning a multitude of mathematical entities. These algebras seamlessly integrate various mathematical structures, including real numbers, vectors, complex numbers, quaternions, and exterior algebras. Remarkably, Clifford algebras extend beyond standard vector analysis's traditional scalar and vector elements by incorporating spatial elements that encapsulate planes and volumes. Consider the vector cross product in three-dimensional space, effectively represented in Clifford algebras as a plane segment delineated by the two vectors involved. Although represented by three unique components akin to a vector, the cross product distinguishes itself by its characteristic sign reversal upon reflection, which is not a property of true vectors. Clifford algebras encapsulate these diverse spatial elements into entities known as multivectors. In the current research, we supplant the standard feature field operations found in deep learning frameworks with operations derived from Clifford algebras that act on fields of multivectors. The principles of Clifford algebras govern the operational dynamics and interrelations within multivectors. To illustrate, convolutional kernels are endowed with multivector elements to facilitate convolution over multivector-valued feature maps.

The Fourier Neural Operator (FNO) represents a novel approach to learning operators that map between infinite-dimensional function spaces. This methodology is particularly significant in solving partial differential equations (PDEs) where traditional neural networks face challenges due to the high dimensionality and complexity of the function spaces involved. The advent of deep learning has provided powerful tools for function approximation in high-dimensional spaces. However, the application to PDEs has been limited by the need for architectures that can inherently handle the mapping of functions. The Fourier Neural Operator (FNO) is designed to address this by leveraging the expressive power of neural networks in the Fourier domain.

FNO operates by parameterizing the integral kernel, which is central to operator learning, in the Fourier space. This parameterization allows for the efficient handling of high-dimensional data and the learning of complex mappings. The FNO framework comprises a series of layers that perform a Fourier transform, a pointwise multiplication with a learned parameter, and an inverse Fourier transform. This process captures the global interactions of the input functions, making it highly effective for learning operators in PDEs. One of the primary advantages of FNO is its efficiency. Operating in the Fourier domain avoids the curse of dimensionality that plagues many neural network approaches. Furthermore, FNO can generalize across different geometries and boundary conditions, making it a versatile tool for various scientific computing applications. FNO has been successfully applied to various problems, from fluid dynamics to climate modelling. Its ability to learn complex operators has shown promise in accelerating traditionally computationally expensive simulations, opening new avenues for research and application in numerical analysis and beyond.

This study integrates Clifford algebra with the Fourier neural operator, yielding the Clifford Fourier Neural Operator (Clifford FNO). The aim is to train the Clifford FNO to adeptly solve the isotropic elastic wave equation, thereby generating the partial derivative wavefields denoted as $\partial_x V_x$, $\partial_z V_x$, $\partial_x V_z$, and $\partial_z V_z$. We evaluate the efficacy of two training methodologies: the rollout and one-step methods. The findings indicate that the rollout method outperforms the one-step method in generating more accurate wavefields, despite the latter initially seeming more congruent with our expectations of a waveform operator. This report is structured as follows: Firstly, a concise overview of Clifford algebra is provided, detailing the fundamental calculation principles inherent to the algebra, which is mostly cited from Brandstetter et al. (2022). Subsequently, the report delves into an exposition of the Clifford Fourier Operator, highlighting it as the pivotal distinction from the conventional FNO. Lastly, we expound upon the training methodologies employed and present the training outcomes.

Clifford algebras

Consider the vector space $\mathbb{R}^n$ with the standard Euclidean product $\langle \cdot,\ \cdot \rangle$, where $n = p + q$, and p and q are the non-negative integers. A real Clifford algebra $Cl_{p,q}(\mathbb{R})$ is an associate algebra generate by $p+q$ orthogonal basis elements $e_1, e_2, \cdots, e_{p+q}$ of the generating vector space $\mathbb{R}^\ltimes$, such that the following quadratic relations hold:

$$\begin{aligned} e_i^2 &= 1 \ \text{ for } \ 1 \le i \le p; \\ e_j^2 &= -1 \ \text{ for } \ p \le j \le p+q; \\ e_{ij} &= e_{ji} \text{ for } i \ne j \end{aligned} \tag{1}$$

The pair (p, q) is called signature and defines a Clifford algebra with the basis elements that span the vector space G^{p+q} of $Cl_{p,q}(\mathbb{R})$. The vector space of Clifford algebra has scalar elements and vector elements. Still, it can also have elements consisting of multiple basis elements of the generating vector space $\mathbb{R}^n$, which can be interpreted as the elements of the planes and volume segments. Extremely low-dimensional Clifford algebras are

(1)$Cl_{0,0}(\mathbb{R})$ which is a one-dimensional algebra that is spanned by the basis element 1 and therefore isomorphic to $\mathbb{R}$ of all real numbers;

(2) $Cl_{1,0}(R)$ which is as two-dimensional algebra with vector space G^1 spanned by $1, e_1$, and the basis vector e_1 squares to -1, and this is isomorphic to $\mathbb{C}$, the field to the complex numbers;

(3) $Cl_{0,2}(\mathbb{R})$ which is a 4-dimensional algebra with vector space G^2 spanned by $\{1, e_1, e_2, e_{12}\}$, where e_1, e_2, e_1e_2 are all squared to -1 and anti-commute. Thus $Cl_{0,2}(\mathbb{R})$ is isomorphic to quaternions $\mathbb{H}$.

Grade

The grade of the Clifford algebra basis element is the dimension of the subspace it represents. For example, the basis element $\{1, e_1, e_2, e_{12}\}$ of vector space G^2 of the Clifford algebra $Cl_{0,2}(\mathbb{R})$ have grades $\{0, 1, 1, 2\}$. Using the concept of grades, we can divide Clifford algebra into subspaces comprising each grade's elements. The subspace of the smallest dimension is M_0, the subspace of all the scalars. Elements of M_1 are vectors, elements of M_2 are called bivectors, and so on. The vector space G^{p+q} of a Clifford algebra can be written as the sum of all these subspaces: $G^{p+q} = M_0 \oplus M_1 \oplus M_2 ... \oplus M_{p+q}$. The elements of the Clifford algebra are called the multivectors, containing elements of the subspaces, i.e., scalars, bivectors, trivectors, ..., k-vectors. The basis with the highest grade are called the pseudoscalar, i.e., in $\mathbb{R}^2$ the pseudoscalar is e_1e_2, in $\mathbb{R}^3$ the pseudoscalar is $e_1e_2e_3$.

Dual

The dual of the multivector $\mathbf{a}$ is defined as $\mathbf{a}^*$, where $\mathbf{a}^* = \mathbf{a}i_{p+q}$ where i_{p+q} represents the psedoscalar of the Clifford algebra $Cl_{p,q}(\mathbb{R})$. This definition helps us relate different multivectors to each other, which is useful when defining the Clifford Fourier transforms. For example in $\mathbb{R}^2$ the psedoscalar is a bivector, and in $\mathbb{R}^3$ the pseudoscalar is a trivector.

Clifford product

The Clifford product is a bilinear operation on the multivectors. For any arbitrary multivectors, $\mathbf{a}, \mathbf{b}, \mathbf{c} \in G^{p+q}$, and scalar λ the Clifford product has the following features: (1) closure, i.e., $\mathbf{ab} \in G^{p+q}$; (2) associativity, i.e., $\mathbf{a}(\mathbf{bc}) = (\mathbf{ab})\mathbf{c}$; (3) communicative scalar multiplication, i.e., $\lambda\mathbf{a} = \mathbf{a}\lambda$ (4) distribution over addition $\mathbf{a}(\mathbf{b} + \mathbf{c}) = \mathbf{ab} + \mathbf{ac}$. The Clifford product is generally non-communicative, i.e., $\mathbf{ab} \neq \mathbf{ba}$,

Clifford algebras $Cl_{2,0}(\mathbb{R})$, and $Cl_{0,2}(\mathbb{R})$

The 4-dimensional vector spaces of these Clifford algebras have the basis vectors $\{1, e_1, e_2, e_1e_2\}$ where e_1, e_2 are squared to +1, and to -1 for $Cl_{0,2}(\mathbb{R})$. For $Cl_{0,2}(\mathbb{R})$, the Clifford product of two multivectors $\mathbf{a} = a_0 + a_1e_1 + a_2e_2 + a_{12}e_1e_2$ and $\mathbf{b} = b_0 + b_1e_1 + b_2e_2 + b_{12}e_1e_2$ is given by:

$$\begin{aligned}\mathbf{ab} =&(a_0b_0 + a_1b_1 + a_2b_2 - a_{12}b_{12})1+\\&(a_0b_1 + a_1b_0 - a_2b_{12} + a_{12}b_2)e_1+\\&(a_0b_2 + a_1b_{12} - a_2b_0 - a_{12}b_1)e_2+\\&(a_0b_{12} + a_1b_2 - a_2b_{12} + a_{12}b_0)e_{12}.\end{aligned} \tag{2}$$

A vector $x = (x_1, x_2) \in \mathbb{R}^2$ with standard Euclidean product $\langle \cdot\ ,\ \cdot\rangle$ can be related to $x_1e_1 + x_2e_2 \in \mathbb{R}^2 \subset G^2$. The Clifford multiplication of the two vectors $x, y \in \mathbb{R}^2 \subset G^2$ yields the Clifford product xy:

$$\begin{aligned}xy =&(x_1e_1 + x_2e_2)(y_1e_1 + y_2e_2)\\=&x_1y_1e_1^2 + x_2y_2e_2^2 + x_1y_1e_1e_2 + x_2y_1e_2e_1\\=&\langle x, y\rangle + \langle x \wedge y\rangle,\end{aligned} \tag{3}$$

where $\wedge$ is the wedge product or the exterior product. The asymmetric quantity $x \wedge y = -y \wedge x$ is associated with the bivector, which can be considered as the oriented plan segment. A unit bivector i_2, spanned by the (orthogonal) basis vectors e_1, and e_2 is determined by the product:

$$i_2 = e_1e_2 = \langle e_1, e_2\rangle + e_1 \wedge e_2 = -e_2 \wedge e_1 = -e_2e_1, \tag{4}$$

which if squared yields $i_2^2 = -1$. Thus i_2 represents a geometric $\sqrt{-1}$. From equation 4 we have $e_2 = e_1i_2 = -i_2e_1$ and $e_1 = i_2e_2 = -e_2i_2$, Using the pseudovector of a scalar is a bivector and the dual of a vector is a vector. The dual pairs in $Cl_{0,2}(\mathbb{R})$ and $Cl_{2,0}\mathbb{R}$ are $1 \leftrightarrow e_1e_2$, and $e_1 \leftrightarrow e_2$. For $Cl_{2,0}(\mathbb{R})$ these dual pairs allow us to write an arbitrary multivector $\mathbf{a}$ as:

$$\mathbf{a} = a_0 + a_1e_1 + a_2e_2 + a_{12}e_1e_2 = 1(a_0 + a_{12}i_2) + e_1(a_1 + a_2i_2) \tag{5}$$

which can be regarded as two complex-valued parts: the spinor parts, which commute with the base elements 1, i.e., $1i_2 = i_21$, and the vector parts, which anti-commutes with the respective base elements e_1, i.e., $e_1i_2 = e_1e_1e_2 = -e_1e_2e_1 = -i_2e_1$. For $Cl_{0,2}(\mathbb{R})$ we have:

$$\mathbf{a} = 1(a_0 + a_{12}i_2) + e_1(a_1 - a_2i_2). \tag{6}$$

This decomposition will be used as the basis for Clifford Fourier transforms. The Clifford algebra $Cl_{0,2}(\mathbb{R})$ is isomorphic to the quaternions $\mathbb{H}$, which is an extension of the complex numbers and are commonly represented as $a + b\hat{i} + c\hat{j} + d\hat{k}$. Quaternions form a 4-dimensional algebra spanned by $\hat{i}$, $\hat{j}$, and $\hat{k}$, where $\hat{i}, \hat{j}, \hat{k}$ are all squared to -1. The basis element 1 is called the scalar, and the elements of $\hat{i}, \hat{j}, \hat{k}$ are called the vector parts of the quaternions.

Clifford algebra $Cl_{3,0}(\mathbb{R})$

The 8-dimensional vector space G^3 of the Clifford algebra $Cl_{3,0}(\mathbb{R})$ has the basis vectors $\{1, e_1, e_2, e_3, e_1e_2, e_2e_3, e_3e_1, e_2e_3, e_1e_2e_3\}$, it is consisited with one scalar 1, three vectos $\{e_1, e_2, e_3\}$, three bivectors $\{e_1e_2, e_2e_3, e_3e_1\}$, and one trivector $e_1e_2e_3$. The trivector is also the pseudoscalar i_3. The dual pair of the $Cl_{3,0}(\mathbb{R})$ are $1 \leftrightarrow e_1e_2e_3$, $e_1 \leftrightarrow$

e_2e_3, $e_2 \leftrightarrow e_3e_1$ $e_3 \leftrightarrow e_1e_2$. The intriguing example of duality of the multivaectors of $Cl_{3,0}(\mathbb{R})$ is the expression of the electromagnetic wavefields, $\mathbf{F} = E + Bi_3$, where the $E = E_xe_1 + E_ye_2 + E_ze_3$, and $B = B_xe_1 + B_ye_2 + B_ze_3$. In this way, the electromagnetic wavefields are decomposed in the electronic vector fields and imaginative bivector fields via the pseudoscalar i_3. For example, for the base component B_xe_1 of B it holds $B_xe_1e_1e_2e_3 = B_xe_2e_3$, which is the dual of E_xe_1. In summary, the electromagnetic wavefield contains three parts, which are three vectors (the electric field componients) and three bivectors (the magnetic field components) multiplied by i_3. This viewpoint gives Clifford neural layers a natural advantage over their default counterparts.

CLIFFORD NEURAL NETWORK

Clifford convolution

The difference between Clifford algebra convolution and traditional convolution lies in that both the input and convolution kernel of geometric algebra convolution are multivectors. Suppose the input of the l^{th} Clifford algebra convolution layer is a multivector which corresponds to the output of the $l-1^{th}$ layer can be expressed as:

$$X^{l-1} = X_r^{l-1} + X_{12}^{l-1}e_{12} + X_{23}^{l-1}e_{23} + X_{31}^{l-1}e_{31}, \tag{7}$$

where X^{l-1} means the output of the previous layer, and it is composed with multiple elements. The term r is the scalar part of X^{l-1}, X_{12}, X_{23}, and, X_{31} are the corresponding bivector parts. The convolution filters are also multivectors which can be expressed as:

$$W^l = W_r^l + W_{12}^le_{12} + W_{23}^le_{23} + W_{31}^le_{31}, \tag{8}$$

where the W_r is the scalar part of W^l, and W_{12}, W_{23}, W_{31} are the bivector part of W^{l-1}. The multiplication of X^{l-1} and W^l with the Clifford product gives X^l, after adding with the bias term B^l and applying the activation function σ, can be expressed as:

$$\begin{aligned} X^l =& \sigma\left(\sum X^{l-1} \otimes W^l + B^l\right) \\ =& \sigma\left(\sum(\langle X^{l-1}, W^l\rangle + X^{l-1} \wedge W^l) + B^l\right), \end{aligned} \tag{9}$$

where $\otimes$ represents the geometric product. From equation 9, we can see that the conventional convolution operation based on the inner product calculation is a part of the Clifford convolution. Compared with the conventional convolution operation, Clifford product could bring a richer detailed description of the correlation between multiple inputs. A more detailed expression can be expressed as:

$$X^{l-1} \otimes W^l = \langle X^{l-1}, W^l\rangle + X^{l-1} \wedge W^l = G_r + G_{12}e_{12} + G_{23}e_{23} + G_{31}e_{31}, \tag{10}$$

$$G_r = X_r^{l-1}W_r^l - X_{12}^{l-1}W_{12}^l - X_{23}^{l-1}W_{23}^l - X_{31}^{l-1}W_{31}^l, \tag{11}$$

$$G_{12} = X_r^{l-1}W_{12}^l + X_{12}^{l-1}W_{23}^l + X_{23}^{l-1}W_{31}^l - X_{31}^{l-1}W_{23}^l, \tag{12}$$

$$G_{23} = X_r^{l-1}W_{23}^l - X_{12}^{l-1}W_{31}^l + X_{23}^{l-1}W_r^l + X_{31}^{l-1}W_{12}^l, \tag{13}$$

$$G_{31} = X_r^{l-1}W_{31}^l + X_{12}^{l-1}W_r^l - X_{23}^{l-1}W_{12}^l + X_{31}^{l-1}W_r^l, \tag{14}$$

Clifford Fourier neural network

The Clifford Fourier transform can be summarized as follow:

$$
\begin{aligned}
\mathcal{F}\{\boldsymbol{f}\}(\xi) =& \frac{1}{2\pi}\int_{\mathbb{R}_2} \boldsymbol{f}(x) e^{-2\pi i_2 \langle x,\xi\rangle} dx, \\
=& \frac{1}{2\pi}\int_{\mathbb{R}_2} \left[1(\underbrace{f_0(x) + f_{12}(x)i_2}_{\text{spinor part}}) + e_1(\underbrace{f_1(x) + f_2(x)i_2}_{\text{vector part}}) \right] e^{-2\pi i_2 \langle x,\xi\rangle} dx \\
=& \frac{1}{2\pi}\int_{\mathbb{R}_2} 1\left(f_0(x) + f_{12}(x)i_2\right) e^{-2\pi i_2 \langle x,\xi\rangle} dx \\
&+ \frac{1}{2\pi}\int_{\mathbb{R}_2} e_1\left(f_1(x) + f_2(x)i_2\right) e^{-2\pi i_2 \langle x,\xi\rangle} dx \\
=& 1\left[\mathcal{F}\left(f_0(x) + f_{12}(x)i_2\right)(\xi)\right] + e_1\left[\mathcal{F}\left(f_1(x) + f_2(x)i_2\right)(\xi)\right].
\end{aligned}
\tag{15}
$$

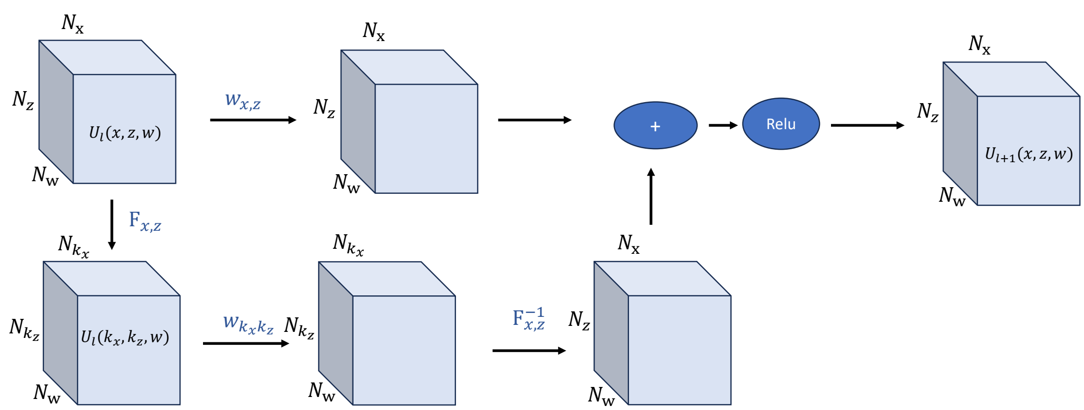
We obtain a Clifford Fourier transform by applying two standard Fourier transforms for the dual pairs $\boldsymbol{f}_0 = f_0(x) + f_{12}(x)i_2$ and $\boldsymbol{f}_1 = f_1(x) + f_2(x)i_2$, which both can be treated as a complexvalued signal $\boldsymbol{f}_0, \boldsymbol{f}_1 : \mathbb{R}^2 \to \mathbb{C}$. Consequently, $f(x)$ can be understood as an element of $\mathbb{C}^2$. The 2D Clifford Fourier transform is the linear combination of two classical Fourier transforms.

Algorithm 1 Pseudocode for 2D Clifford Fourier layer using $\mathrm{Cl}_{2,0}$

function CLIFFORDSPECTRALCONV2D($W, x, m1, m2$)
    $x_v, x_s \leftarrow$ VIEW_AS_DUAL_PARTS(x)
    $f(x_v) \leftarrow$ FFT2(x_v)
    $f(x_s) \leftarrow$ FFT2(x_s)
    $f^*(x_v) \leftarrow [f(x_v)[...; m1; m2], f(x_v)[...; -m1; -m2 :]]$
    $f^*(x_s) \leftarrow [f(x_s)[...; m1; m2], f(x_s)[...; -m1; -m2 :]]$
    $f^*(x) \leftarrow f^*(x_s).r + f^*(x_v).r + f^*(x_v).i + f^*(x_s).i$
    $f^*(x) \leftarrow f^*(x)W$
    $\tilde{x_v} \leftarrow$ IFFT2($f^*(x)[1] + f^*(x)[2]$)
    $\tilde{x_s} \leftarrow$ IFFT2($f^*(x)[0] + f^*(x)[3]$)
    $\tilde{x} \leftarrow$ VIEW_AS_MULTIVECTOR($\tilde{x_v}, \tilde{x_s}$)
    return $\tilde{x}$
end function
function CLIFFORDFOURIERLAYER2D(W_f, W_c, x)
    $y1 \leftarrow$ CLIFFORDSPECTRALCONV2D($W_f, x, m1, m2$)
    $x2 \leftarrow$ VIEW_AS_REALVECTOR(x)
    $y2 \leftarrow$ CLIFFORDCONV($W_c, x2$)
    $y2 \leftarrow$ VIEW_AS_MULTIVECTOR($y2$)
    $out \leftarrow$ ACTIVATION($y1 + y2$)
    return out
end function

The function CLIFFORDSPECTRALCONV2D takes a weight matrix W, an input matrix x, and the dimensions $m1, m2$ as its parameters. The function processes the input matrix using a series of Fourier transforms and operations specific to Clifford algebras. The input matrix x is split into its dual parts x_v and x_s, representing the vector and scalar parts, respectively. A 2D fast Fourier transform (FFT2) is applied to both x_v and x_s. Fourier modes are extracted for both parts. For the vector part, the modes are $f^*(x_v)$, and for the scalar part, the modes are $f^*(x_s)$. A geometric product is applied in the Fourier space, involving the weight matrix W and the Fourier modes. An inverse 2D FFT is performed on both the vector and scalar Fourier modes to transform them back to the spatial domain. The results are then combined to view them as a multivector $\tilde{x}$. The function returns this multivector, which contains the transformed image data.

The function `CLIFFORDFOURIERLAYER2D` represents a layer in a neural network that uses the Clifford Fourier transform within its structure. It calls the `CLIFFORDSPECTRALCONV2D` function to apply a spectral convolution to the input. The real vector part of the input is retrieved. A Clifford convolution is then applied to the real vector part. The spectral convolution and the Clifford convolution results are combined using an activation function. The final result is the output from the function, representing the processed layer output.



1

FIG. 1. The architecture of the Fourier neural layer obtained from Li et al. (2020). including the Fourier transform $\mathrm{F}_{\mathrm{x,z}}$, the wavenumber-domain multiplication with filter w_{k_x,k_z} , the inverse Fouriertransform $\mathrm{F}^{-1}_{\mathrm{x,z}}$, the space-domain filter $w_{x,z}$, and the ReLU activation function, obtained from Wei and Fu (2022).

TRAINING PROBLEM SET UP

The problem we focus on here is simulating the isotropic elastic waveform within a specific investigation area. Therefore the target of the neural network is in the field $\mathcal{D} = \{(x,y,t)|x \in [a,b], y \in [c,d], t \in [0,T]\}$, approximate the following mapping $G : (\Sigma, \Omega) \to u$, where $\Omega = (V_p, V_s, \rho)$, are the elastic media parameters, including P-wave velocity V_p, S-wave velocity V_s, and density ρ. The term Σ represents the input wavefields within the self-regression model or the space-time coordinate vectors within the none-self-regression model. The term u represents the output, which is the elastic wave-

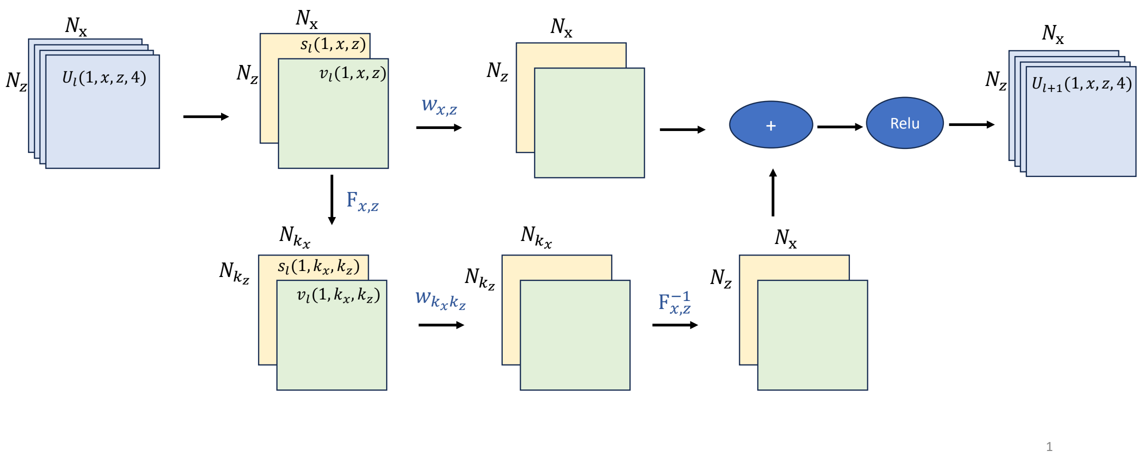


FIG. 2. The architecture of the Clifford Fourier neural operator, using one Clifford multivector U_l, with a dimension of $\mathbb{C}^{1\times \mathrm{N_z},\times \mathrm{N_x}\times 4}$ for the input as an exmaple. The Clifford multivector U_l is firstly being viewed as dual parts, which means that it will be changed into the spinor part s_l and vector part v_l. The complex-valued spinor part s_l and vector part v_l, each of which has the dimension of $\mathbb{C}^{1\times \mathrm{N_z},\times \mathrm{N_x}\times 1}$, will undergo the 1D convolution, and the Fourier domain multiplication just like Figure 2. Finally the spinor part s_l and vector part v_l are transformed back into the Clifford multivector with dimension $\mathbb{C}^{1\times \mathrm{N_z},\times \mathrm{N_x}\times 4}$, which will be regarded as the updated data U_{l+1}.

fields.

Training in autoregression way and none-autoregression way

Training the neural network within the none-autoregression way, or the roll out way, means that the input of the neural network is Σ, which are position coordinates (x, y, t), and the neural network is trained to give the results:

$$\hat{u}(x, y, t) = f_\theta((x, y, t), \Omega) \in \mathbb{R}. \tag{16}$$

The training methodology is also presented in Figure 3. As wavefield simulation needs the source information. Thus, despite the coordinate information, I also use the initial time steps of the wavefields as the input for network training. These initial wavefields carry the source information which are essential for waveform modelling. Supposing that we have a maximum of 8 time steps for training. The roll-out training method uses several time steps of the wavefield as input (the three-time steps Figure 3 represented as black boxes) and is trained to generate the following time steps of the wavefield(the five-time steps Figure 3 represented as blue boxes).

If we were to train the neural network in an autoregression way, or the one-step way, the neural network would be trained to learn how to map the wavefield at the previous time step to the wavefield at the next time step:

$$\hat{u}(t) = f_\theta(u(t - \Delta t), \Omega) \in \mathbb{R}^{n\times m}. \tag{17}$$

The target of the training process is to find θ^* which can minimize the objective function $\mathcal{L}$ on the training dataset $\mathcal{T}$ which can be formulated as:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}[u(x,y,t), \hat{u}(x,y,t)]\ \forall x,y,t \in \mathcal{D}, u \in \mathcal{T} \tag{18}$$

The one-step training methodology is also presented in Figure 4. In the training stage of the one-step method, the network uses several time steps of the wavefield as input (3-time steps in this Figure) and is trained to generate the wavefield of one-time step. In the validation stage of the one-step training, the network uses several time steps of the wavefield as input (3-time steps in this Figure) and is trained to generate the wavefield of one-time step, but the wavefields generated by the neural network will be used as the input for generating the next time step. A demonstration of the validation stage of one step training is plotted in Figure 5, and more details about the one-step training can be found in Wei and Fu (2022).

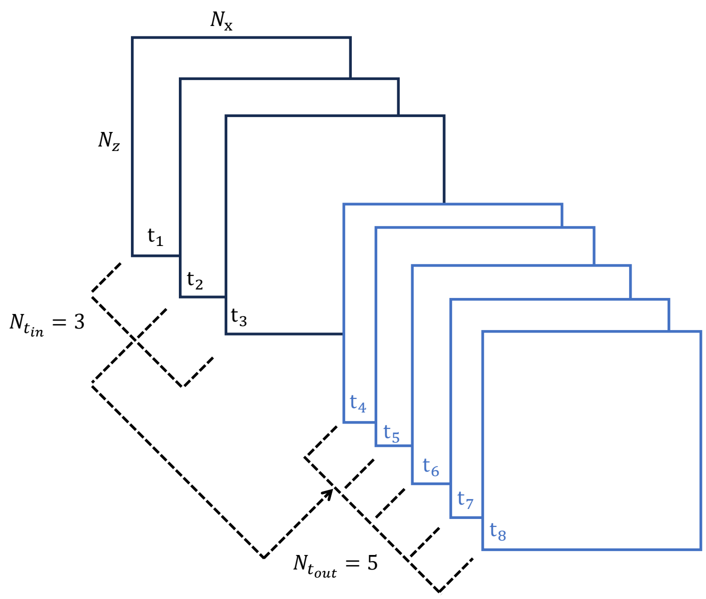


FIG. 3. Roll-out training method. Supposing that we have a maximum of 8 time steps for training. The roll-out training method uses several time steps of the wavefield as input (3 time steps in this Figure), and is trained to generate the following time steps of the wavefield(5 time steps in this Figure).

Clifford Fourier neural network roll-out training results

The presented figure 6 illustrates a comparison between predicted wavefields generated by a neural network and the corresponding true wavefield data at six discrete time steps: $t = 0.116s, 0.146s, 0.196s, 0.294s, 0.392s$, and $0.586s$. The predictions pertain to partial derivatives of the velocity field in the x and y directions, specifically $\partial_x V_z$, $\partial_z V_x$, $\partial_z V_z$, and $\partial_z V_z$. The Clifford Fourier Neural Operator (Clifford FNO), has been trained via a roll-out training methodology, which takes in the coordinate information, the source information and generates the whole time steps wavefields under consideration. Analyzing the snapshots at various time frames, we observe a high degree of fidelity in the network's predictions. Initially, at $t = 0.116s$, the predicted wavefields exhibit a remarkable resemblance to the true data, capturing both the major wavefronts and the subtler wave patterns.

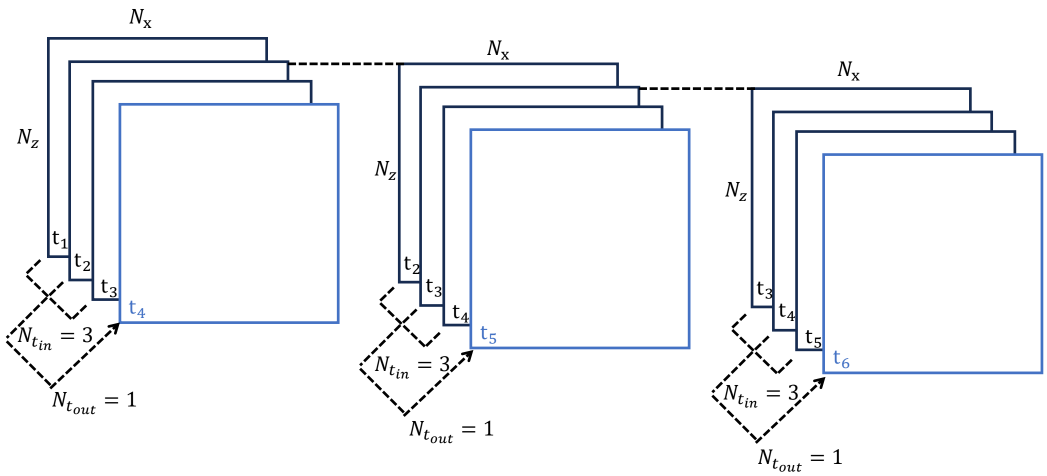


FIG. 4. One-step training method training stage. In the training stage of the one-step method, the network uses several time steps of the wavefield as input (3-time steps in this Figure) and is trained to generate the wavefield of the next one time step.

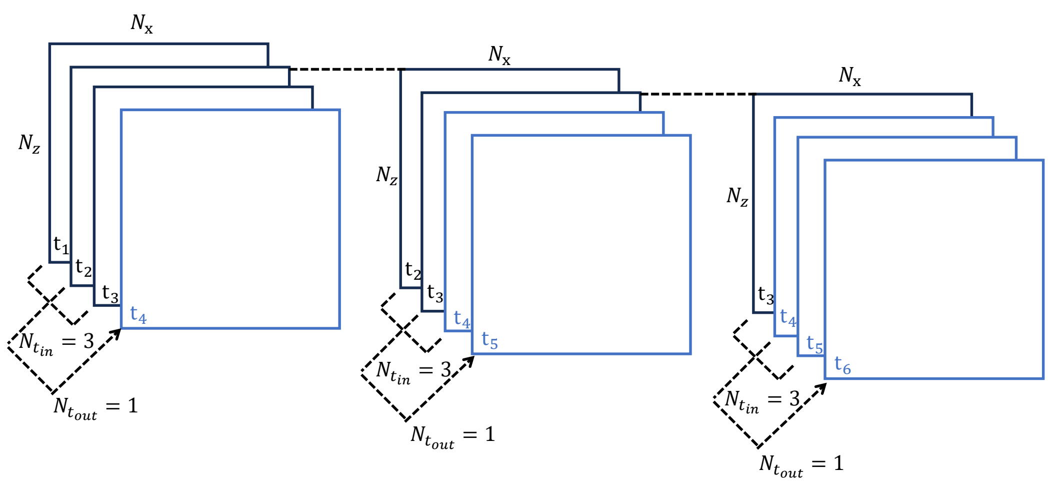


FIG. 5. One-step training method validation stage. In the validation stage of the one-step training, the network uses several time steps of the wavefield as input (3-time steps in this Figure) and is trained to generate the wavefield of one time step, but the wavefields generated by the neural network will be used as the input for generating the next time step.

As time progresses, the complexity of the wavefield interactions increases, yet the network demonstrates a proficient level of accuracy up to $t = 0.294s$, after which, a slight divergence in the finer details becomes discernible. Particularly, in the $\partial_x V_z$ and $\partial_z V_x$ fields, the prediction retains the overall structure of the wavefronts but begins to differ slightly in intensity and the exact positioning of the peaks and troughs. This divergence is more

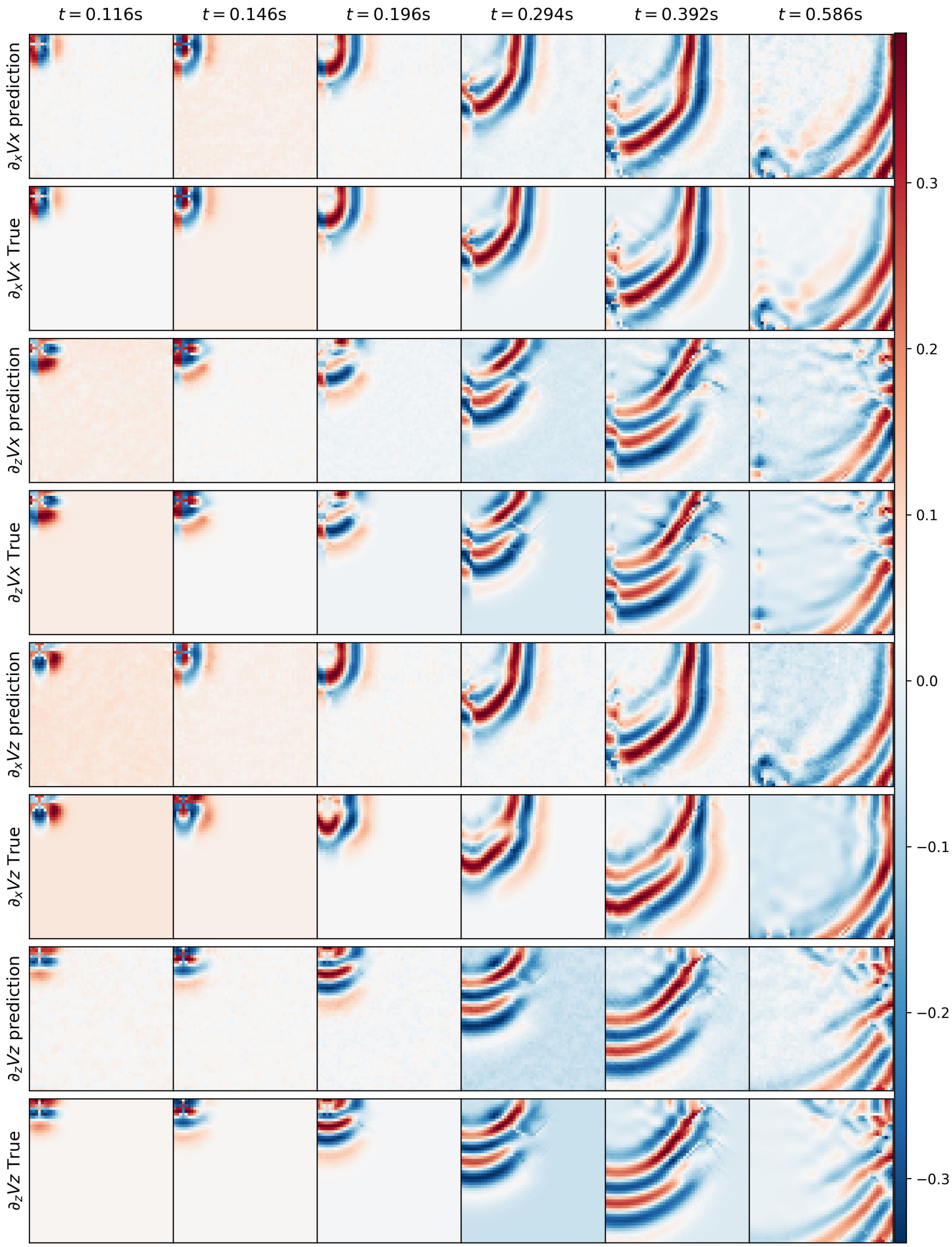


FIG. 6. A example of the snapshots of the wavefields at six-time steps, $\partial_x V_x$, $\partial_z V_x$, $\partial_x V_z$, $\partial_z V_z$, generated by the Clifford FNO. The Clifford FNO is trained by using the roll-out training methodology.

noticeable in the $\partial_z V_z$ component, where the intensity of the predicted waveforms appears to be slightly attenuated compared to the true data. Nonetheless, at $t = 0.586s$, the network

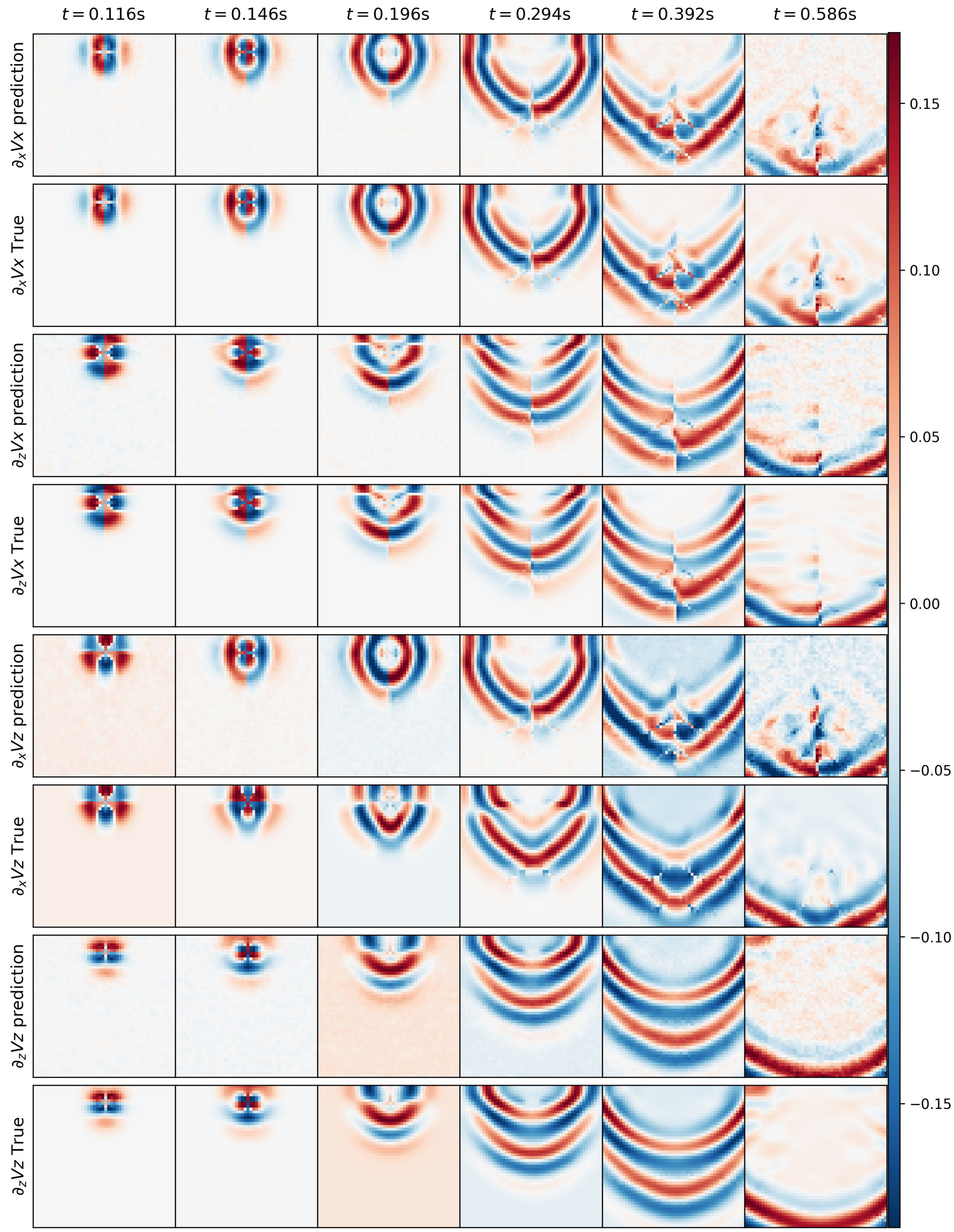


FIG. 7. Another example of the snapshots of the wavefields at six-time steps, $\partial_x V_x$, $\partial_z V_x$, $\partial_x V_z$, $\partial_z V_z$, generated by the Clifford FNO. The Clifford FNO is trained by using the roll-out training methodology.

manages to predict the general direction and movement of the waveforms.

These results indicate a sophisticated level of learning by the Clifford FNO, demonstrating its capability to not only grasp the fundamental dynamics of the wavefields but also to generalize well into future states, a non-trivial task given the inherent complexity and non-linearity of such systems. The minor discrepancies noted are expected and present areas for further refinement, potentially through enriched training data or an extended training period. Overall, the comparative analysis underscores the potential of advanced neural network architectures, such as the Clifford FNO, in accurately simulating physical phenomena, thereby serving as a potent tool for forecasting and analysis in various scientific and engineering domains. Similar observation can be seen in the wavefield generated on another model, plotted in Figure 9.

Clifford Fourier neural network one-step training results

Figure 8 represents the efficacy of a Clifford Fourier Neural Operator (FNO) that has been trained using the one-step training methodology to predict wavefields. This figure displays a series of snapshots at six time steps: $t = 0.002s, 0.004s, 0.008s, 0.01s, 0.012s$, and $t = 0.02s$. The predictions and true values are shown for the partial derivatives of the velocity field components: $\partial_x V_z, \partial_z V_x, \partial_z V_z$, and $\partial_z V_z$.

The one-step training approach focuses on training the network by feeding it single steps of the wavefield at a time, a departure from training on entire sequences. This method has implications for the neural network's ability to predict immediate future states with higher precision. In the initial time steps ($t = 0.002s$ to $t = 0.008s$), the Clifford FNO shows a high degree of accuracy, with the predicted wavefields closely matching the true data in both pattern formation and intensity levels. The clear definition of wavefronts suggests that the network has successfully captured the essential dynamics of the physical system at these early stages. As time advances to $t = 0.01s$ and beyond, subtle discrepancies begin to emerge. Specifically, the predictions show slight deviations in wavefront sharpness and positioning, indicating a slight model-performance degradation as the prediction horizon extends. At the final time step shown ($t = 0.02s$), although the network still captures the general direction and behaviour of the wavefields, there is a noticeable difference in the intensity and complexity of the wave patterns. The predicted wavefields appear smoother and less nuanced compared to the intricate patterns present in the true data.

This analysis suggests that while the Clifford FNO is capable of capturing the immediate dynamics of the wavefields with high fidelity, its performance diminishes when predicting further into the future. This could be due to the limited temporal context provided by the one-step training approach. Training on longer sequences or employing a multi-step training strategy might provide the network with a more comprehensive understanding of the temporal evolution of the wavefields, potentially improving long-term prediction accuracy.

CONCLUSION

In this study, we train a fast-forward modeling method with the Cliffrod Fourier Neural Operator(FNO). The network consists of three parts, which are two dimensions projection layers that operate on time and several Fourier layer that learns the spatial partial deriva-

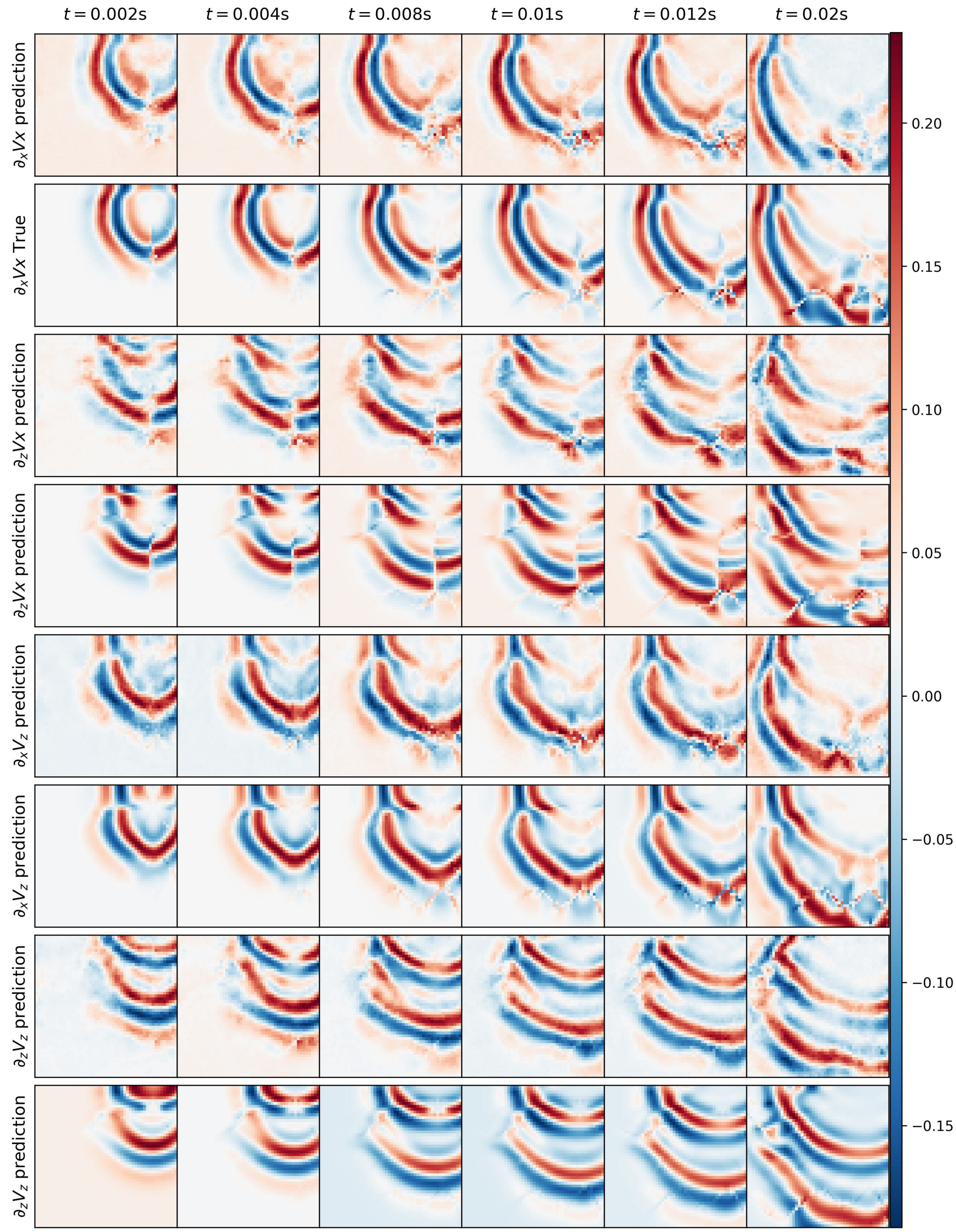


FIG. 8. A example of the snapshots of the wavefields at six-time steps, $\partial_x V_x$, $\partial_z V_x$, $\partial_x V_z$, $\partial_z V_z$, generated by the Clifford FNO. The Clifford FNO is trained by using the one-step training methodology.

tives. The power of the Fourier Neural operator comes from the combination of the Clifford linear operation, operators that resemble partial differential calculation (via the Clifford

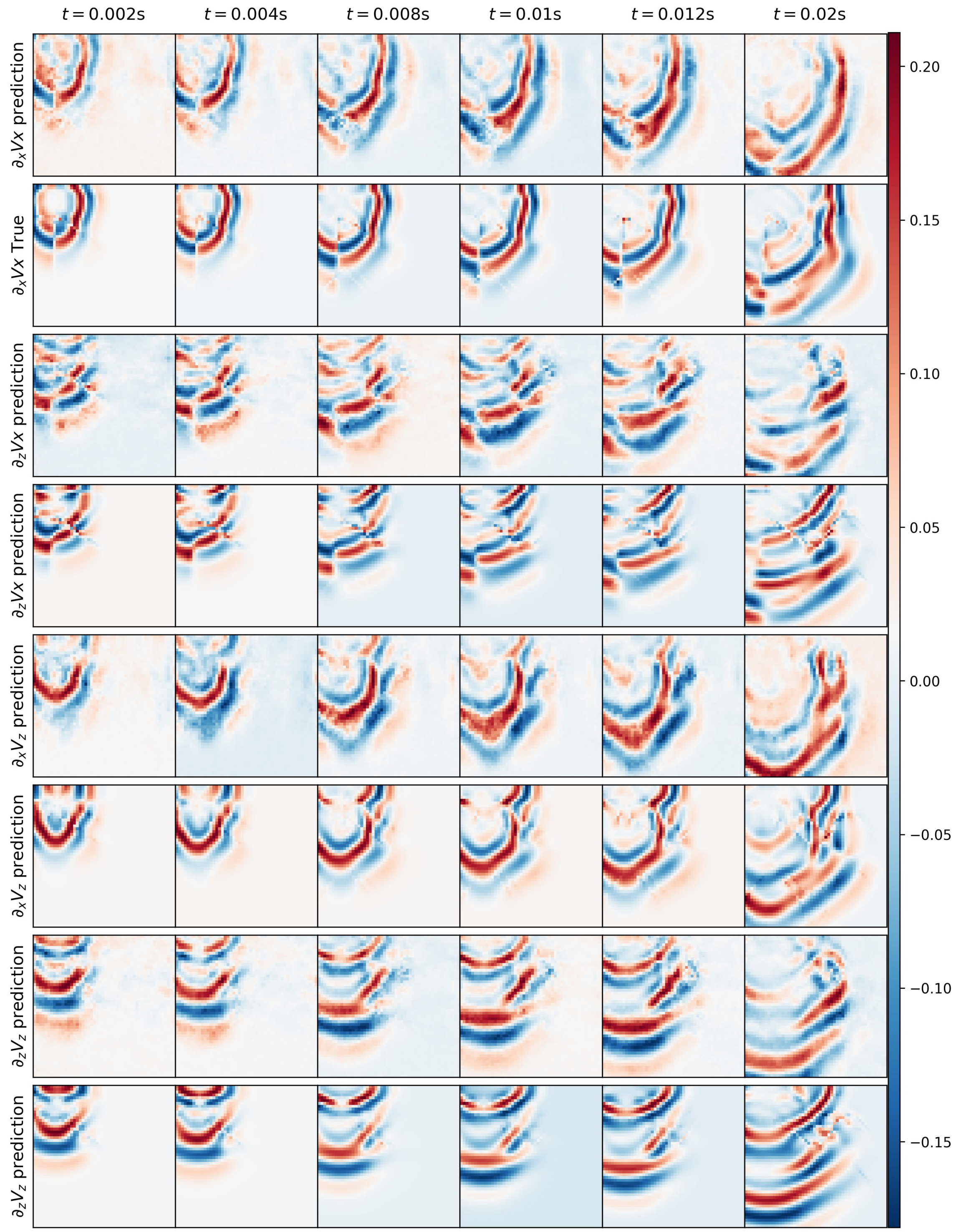


FIG. 9. Another example of the snapshots of the wavefields at six-time steps, $\partial_x V_x$, $\partial_z V_x$, $\partial_x V_z$, $\partial_z V_z$, generated by the Clifford FNO. The Clifford FNO is trained by using one-step training methodology.

Fourier transform), and the non-linear local activation. The numerical tests suggest that FNO could generate promising wavefields within certain prediction steps, however, with decreasing accuracy as time propagates.

ACKNOWLEDGMENTS

The sponsors of CREWES are gratefully thanked for continued support. This work was funded by CREWES industrial sponsors, NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19.

REFERENCES

Brandstetter, J., Berg, R. v. d., Welling, M., and Gupta, J. K., 2022, Clifford neural layers for pde modeling: arXiv preprint arXiv:2209.04934.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A., 2020, Fourier neural operator for parametric partial differential equations: arXiv preprint arXiv:2010.08895.

Wei, W., and Fu, L.-Y., 2022, Small-data-driven fast seismic simulations for complex media using physics-informed fourier neural operators: Geophysics, **87**, No. 6, T435–T446.