Clifford Fourier neural operator as a tool to learn and describe elastic wavefield displacements

Tianze Zhang, Kris Innanen, Daniel Trad

Banff, Canada.
2023-12-08

NSERC CRSNG
UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
Department of Earth, Energy, and Environment

Content

1. Fourier neural operators

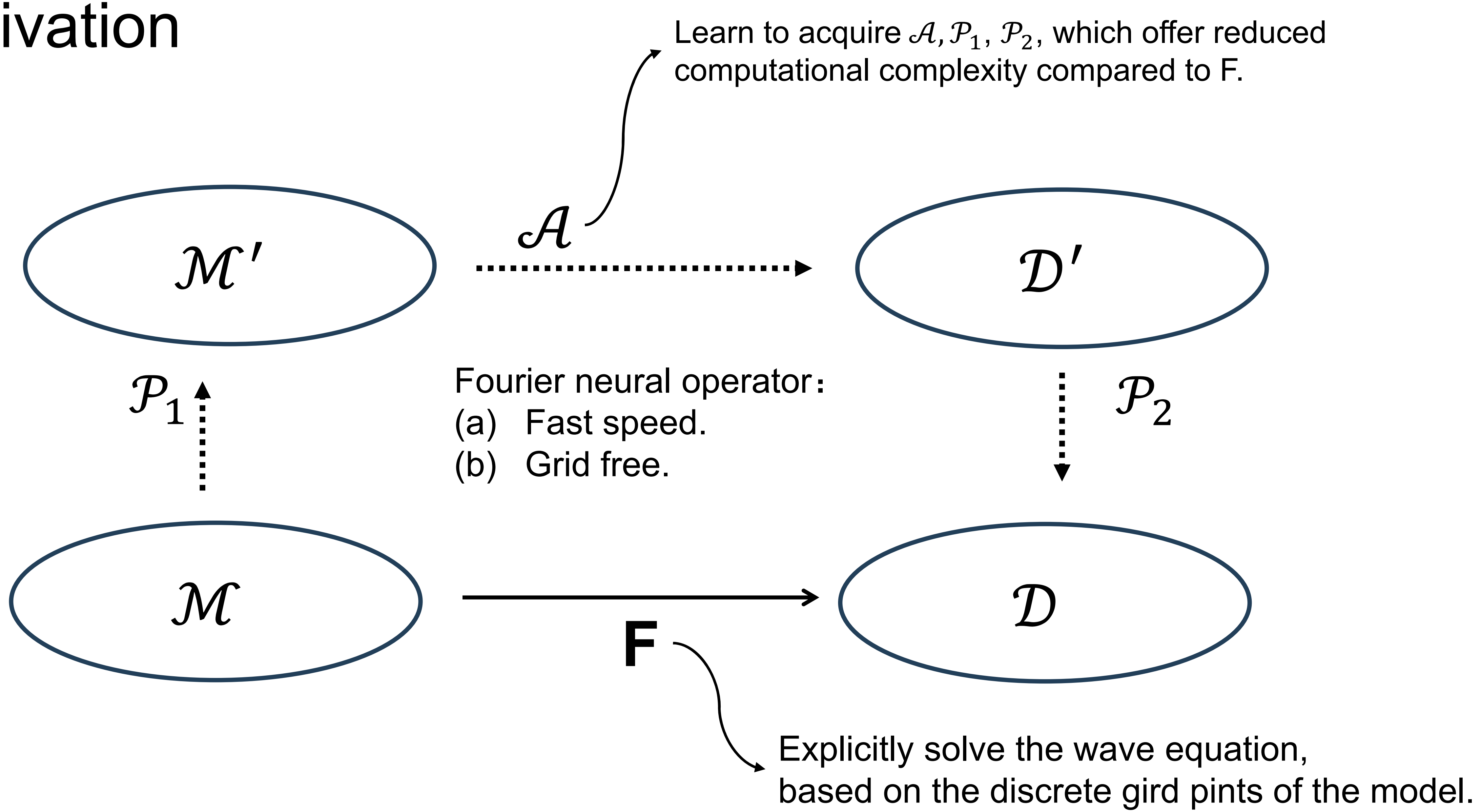
2. Clifford Fourier neural operators.

3.Numerical training results.

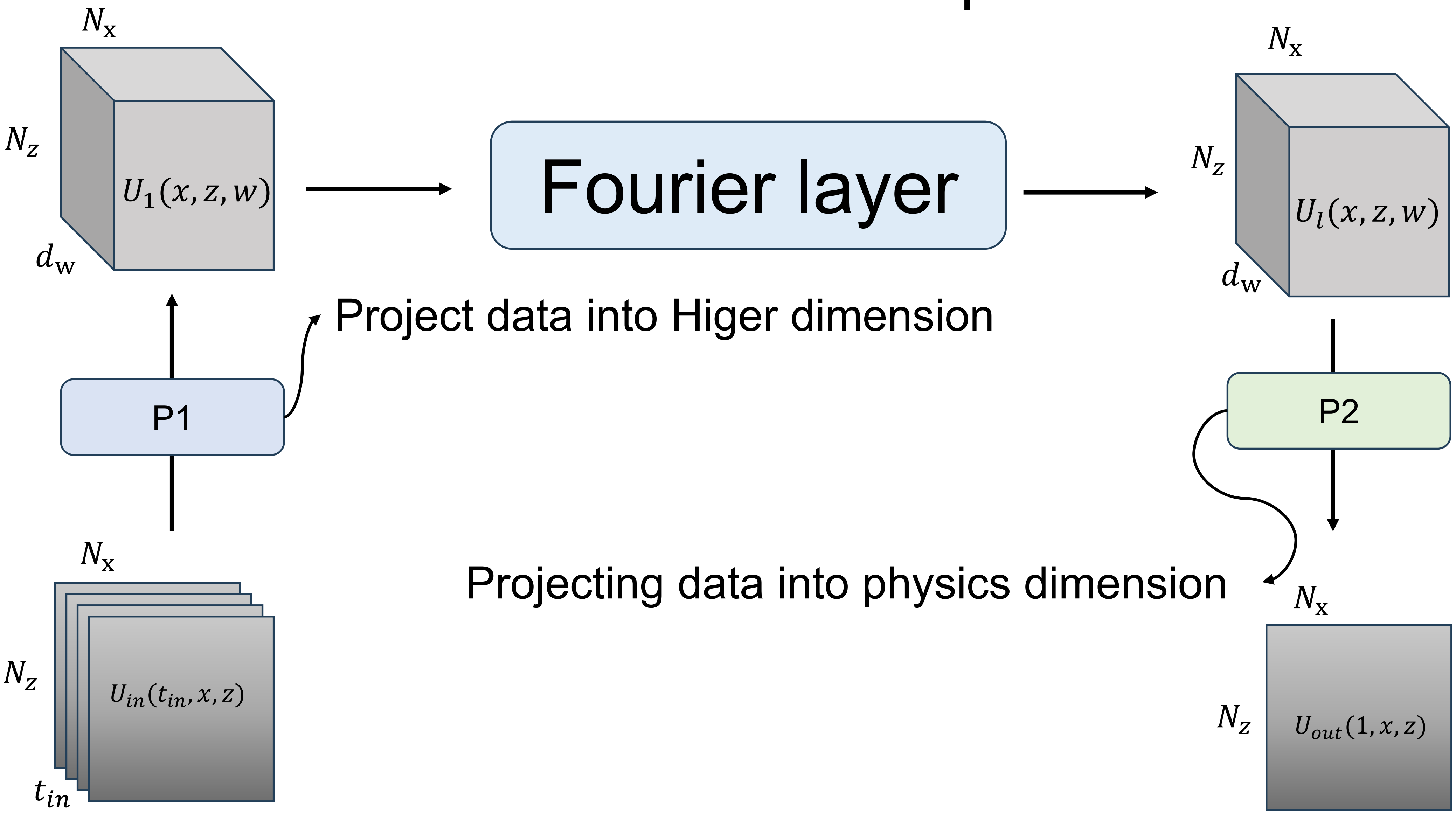
4. Conclusion and feature study.

Part one Introduction to Fourier neural operator

Motivation

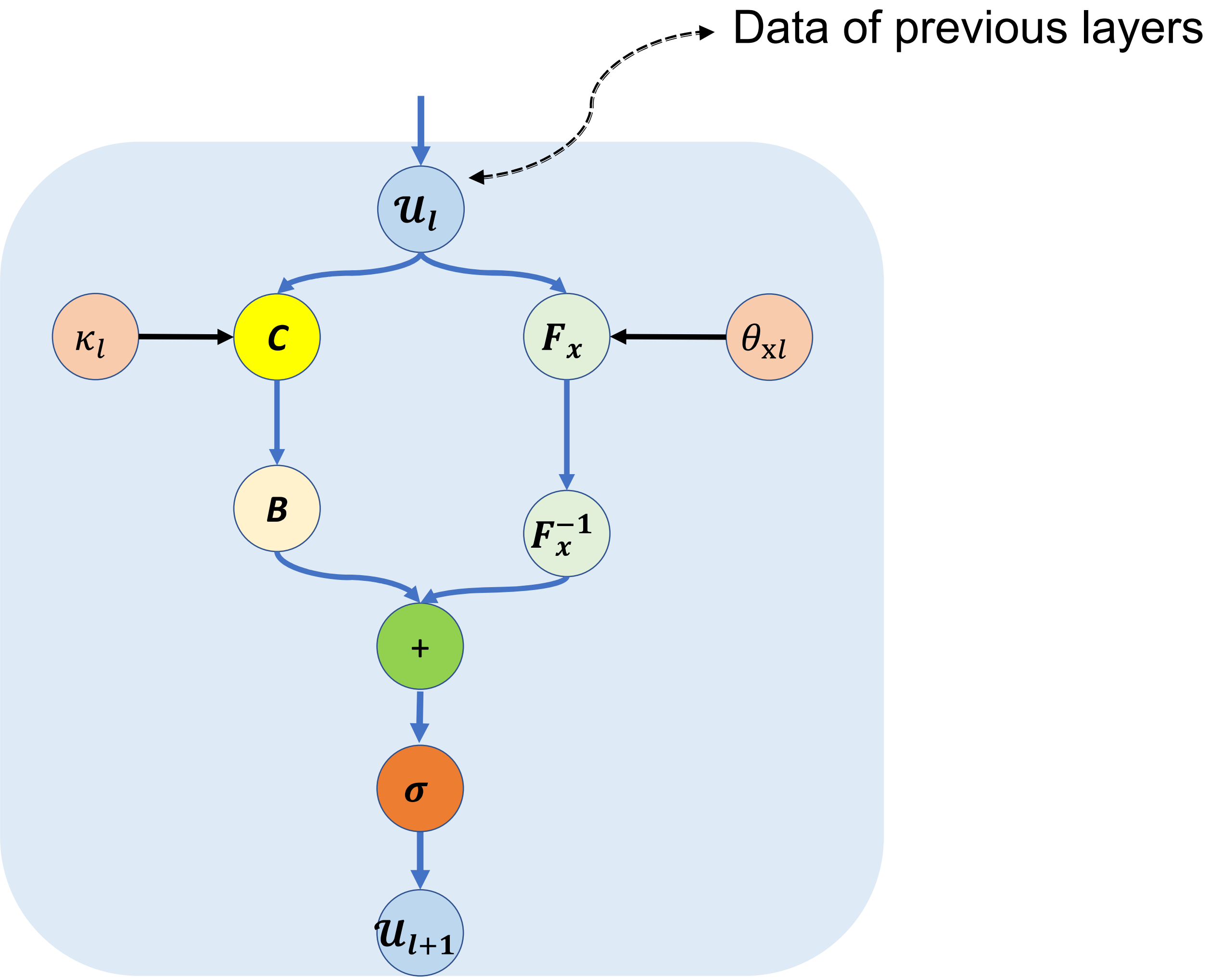

Learn to acquire $\mathcal{A}, \mathcal{P}_1, \mathcal{P}_2$, which offer reduced computational complexity compared to F.
$\mathcal{A}$
$\mathcal{M}'$
$\mathcal{D}'$
$\mathcal{P}_1$
$\mathcal{P}_2$
Fourier neural operator：
(a) Fast speed.
(b) Grid free.
$\mathcal{M}$
$\mathcal{D}$
F
Explicitly solve the wave equation,
based on the discrete gird pints of the model.

The basic structure of the Fourier neural operator.



2. Clifford Fourier neural operators

Conventional 1 layer of FL

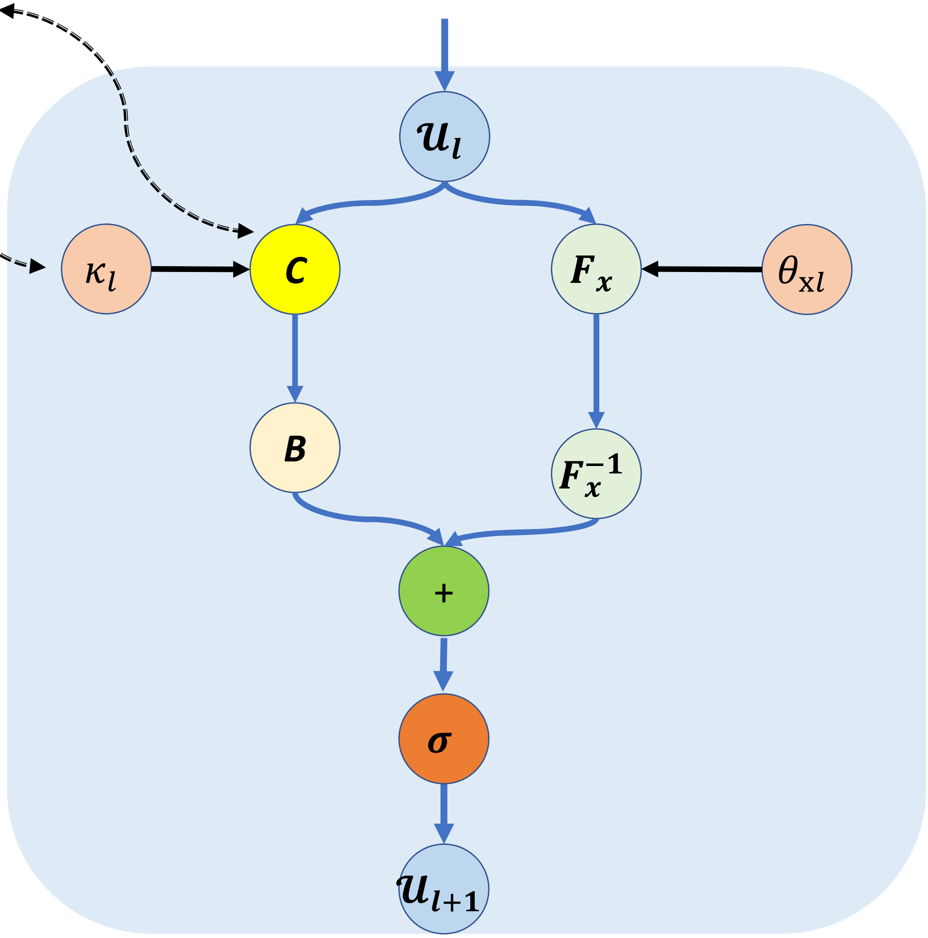


Original FNO Fourier Layer L_l

Conventional 1 layer of FNO

1D convolution operation

1D convolution kernel



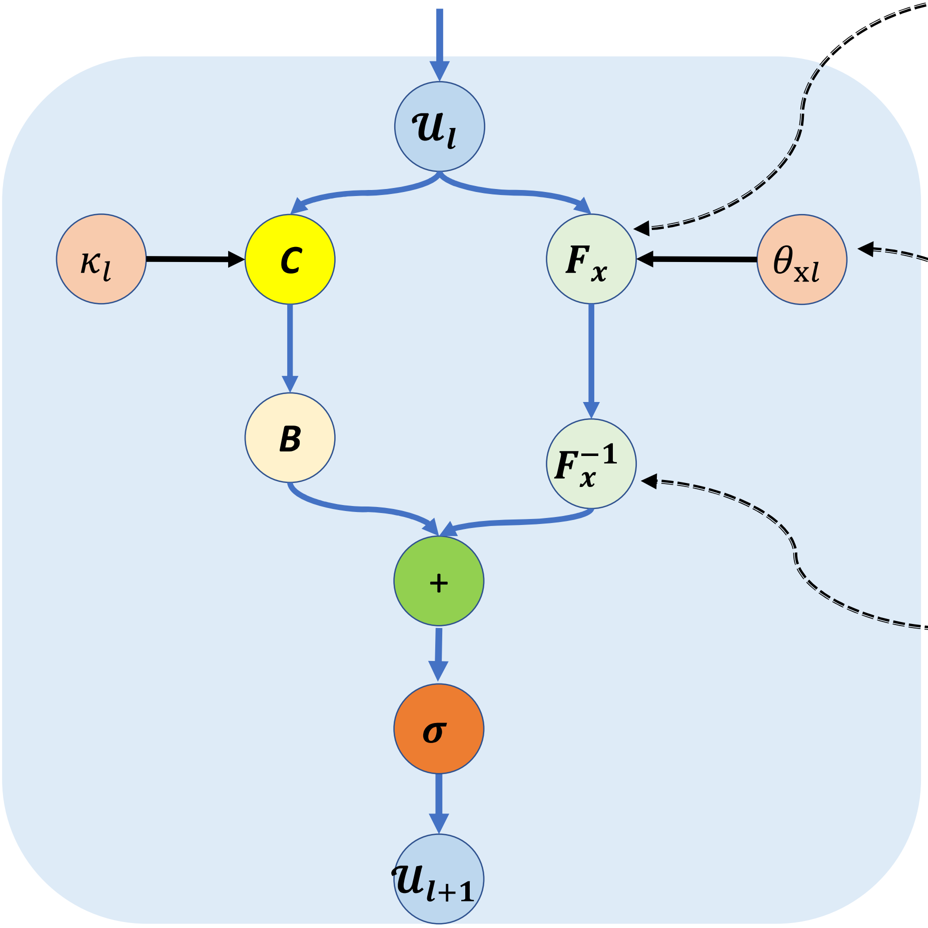
Learning localized wavefield features, i.e., how the phases of the wavefields changes at the geological interfaces.

Original FNO Fourier Layer L_l

2. Clifford Fourier neural operators

Conventional 1 layer of FNO

Learning nonlocalized wavefield features, i.e., the speed and the direction of the wavefields.



FFT with respect to x direction

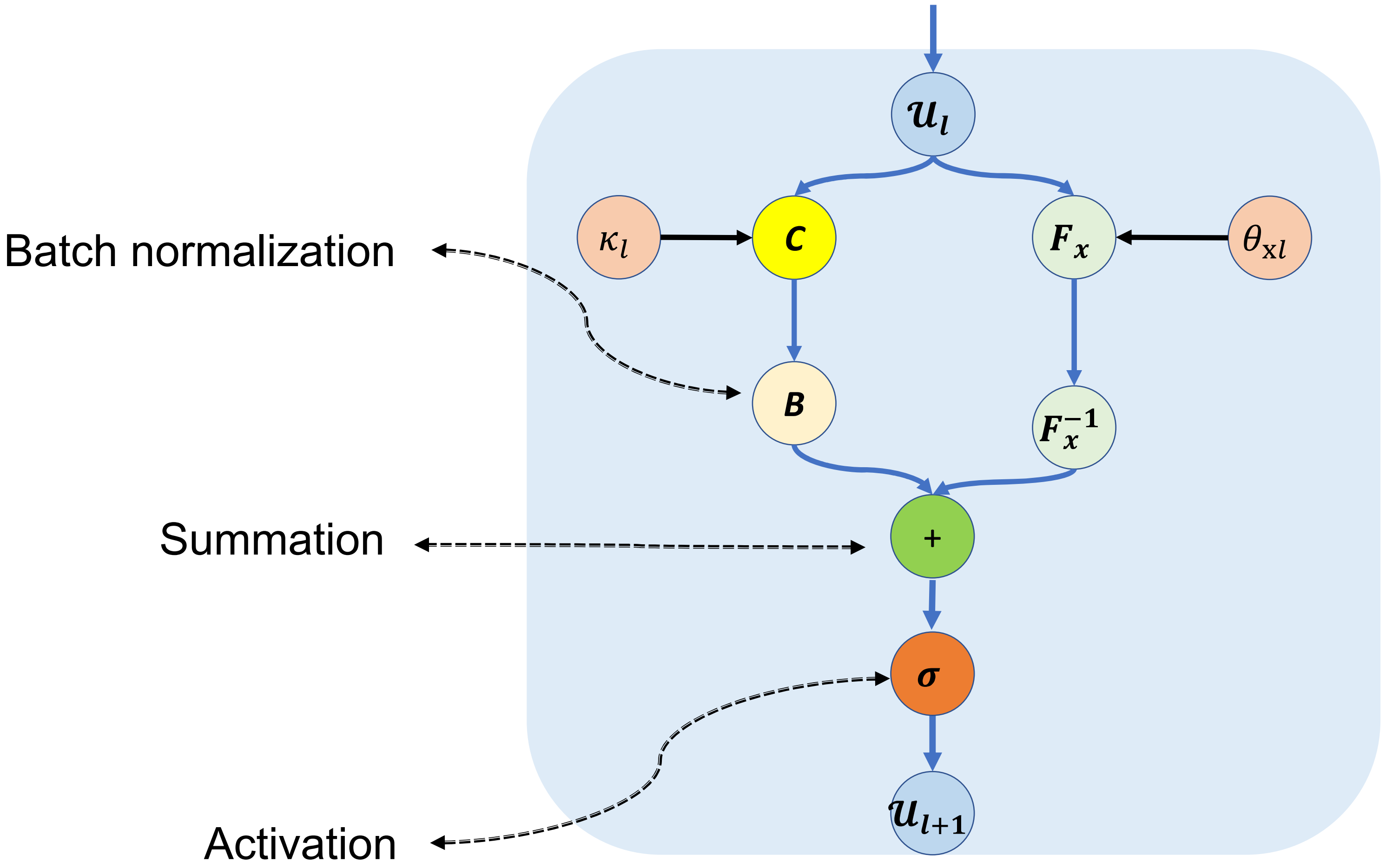
Multiplication with a matrix with the FFT data $\theta_{xl} \in \mathbb{C}^{k_r \times k_r}$

k_r : Hyperparameter of training

Inverse FFT with respect to x direction

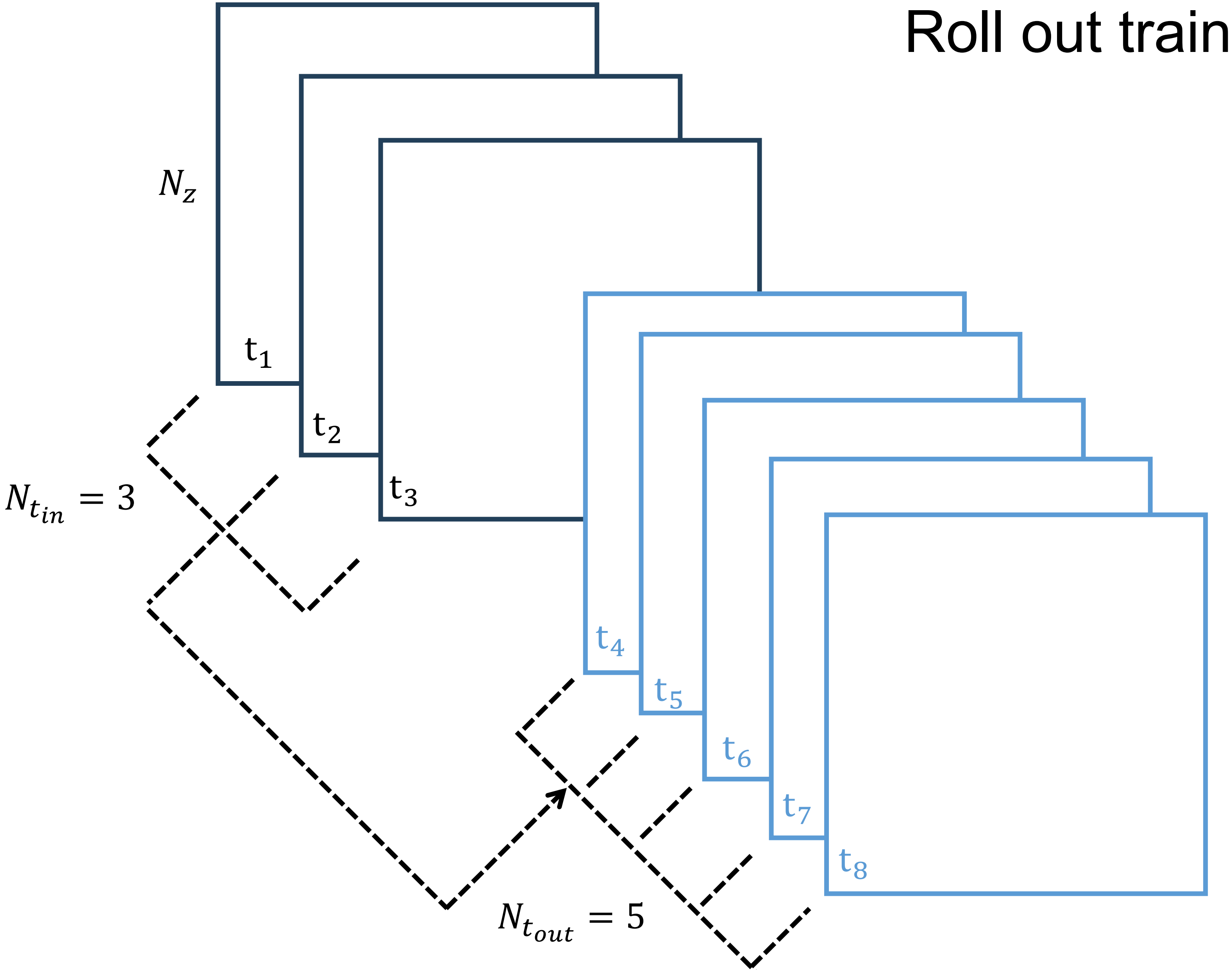
Original FNO Fourier Layer L_l

Conventional 1 layer of FNO



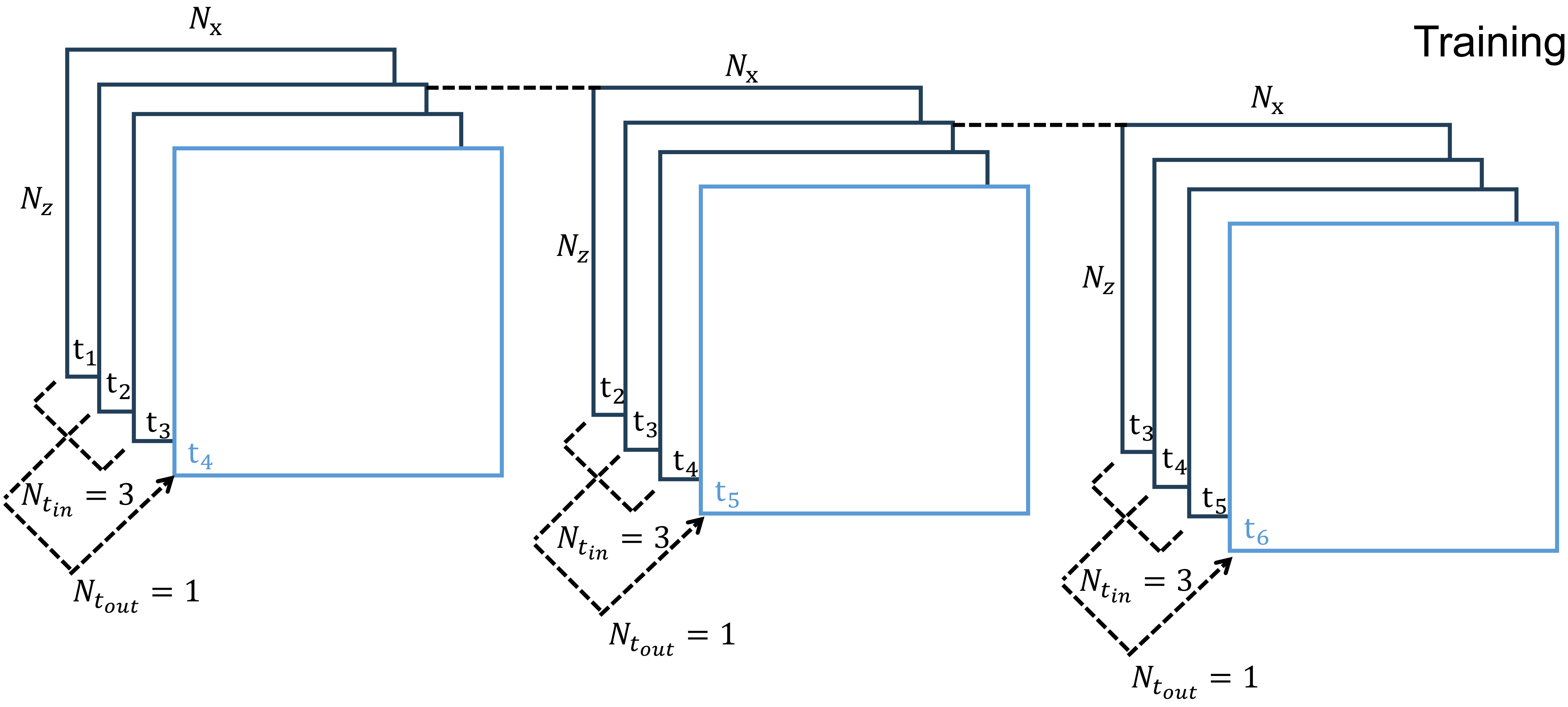
Original FNO Fourier Layer L_l

Roll out training method



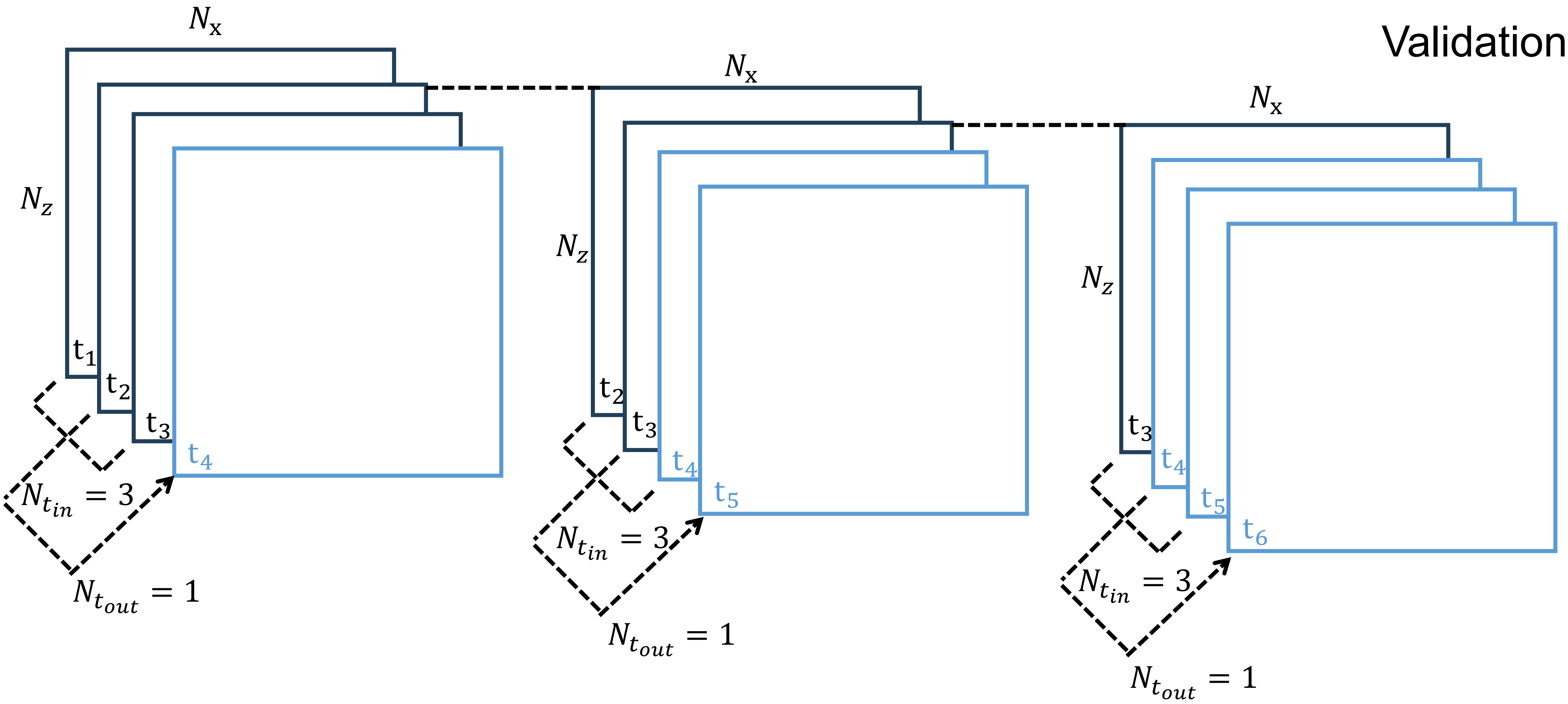
One step training method

Training stage



One step training method

Validation stage



Part two Clifford Fourier neural operators.

2. Clifford(Quaternion) Fourier neural operators

Clifford Fourier neural operator is the complex number extension of the real number valued Fourier neural operator.

Quaternion algebra can be seen as a specific instance of a Clifford algebra.

A quaternion Q is a complex number defined in a four-dimensional space as $\mathrm{Q} = r + x\mathbf{e_1} + y\mathbf{e_2} + z\mathbf{e_1}\mathbf{e_2}$

r, x, y, z real numbers

1, $\mathbf{e_1}, \mathbf{e_2}$, $\mathbf{e_1}\mathbf{e_2}$ quaternion unit basis

Complex number

$$\mathrm{C} = r + x\mathbf{e_1}$$

Real part — r; Imaginary part — $x\mathbf{e_1}$

Conjugate

$$\mathrm{C}^* = r - x\mathbf{e_1}$$

Quaternion number

$$\mathrm{Q} = r + x\mathbf{e_1} + y\mathbf{e_2} + z\mathbf{e_1}\mathbf{e_2}$$

Real part — r; Imaginary part — $x\mathbf{e_1} + y\mathbf{e_2} + z\mathbf{e_1}\mathbf{e_2}$

Conjugate

$$\mathrm{Q}^* = r - x\mathbf{e_1} - y\mathbf{e_2} - z\mathbf{e_1}\mathbf{e_2}$$

Quaternion unit basis:

$$\mathbf{e_1}^2 = \mathbf{e_2}^2 = -1 \qquad (\mathbf{e_1}\mathbf{e_2})^2 = -1$$

Pseudoscalar ($\mathbf{e_1}\mathbf{e_2}$)

Quaternion product

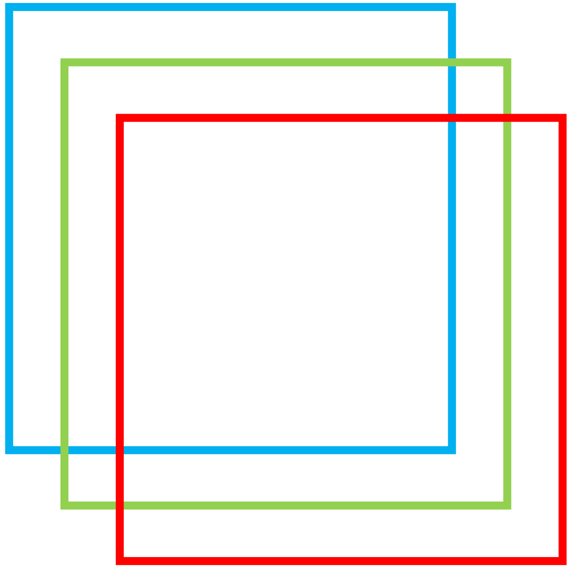
$$Q_1 = r_1 + x_1\mathbf{e_1} + y_1\mathbf{e_2} + z_1\mathbf{e_1}\mathbf{e_2} \quad Q_2 = r_2 + x_2\mathbf{e_1} + y_2\mathbf{e_1} + z_2\mathbf{e_1}\mathbf{e_2}$$

$$Q_1 \otimes Q_2 = (r_1 r_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + (r_1 x_2 + x_1 r_2 + y_1 z_2 - z_1 y_2)\mathbf{e_1} + (r_1 y - x_1 z_2 + y_1 r_2 + z_1 x_2)\mathbf{e_2} + (r_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 r_2)\mathbf{e_1}\mathbf{e_2}$$

$$Q_1 \otimes Q_2 = \underbrace{Q_1 \cdot Q_2}_{\text{Inner product}} + \underbrace{Q_1 \wedge Q_2}_{\text{Wedge product}}$$

The nature ability to represent multi-dimensional data

RGB image



Conventional ML methods represent multi-dimensional data along the channel dimension

Batch size | Channel size | Feature dimension

$$\mathbb{R}^{1\times 3\times N_z\times N_x}$$

Quaternion represent multi-dimensional data as an entity .

Batch size | Channel size | Feature dimension

$$\mathbb{H}^{1\times 1\times N_z\times N_x\times 3}$$

Quaternion neural network product

$$\mathbf{Y} = \boldsymbol{X} \otimes \mathbf{W}$$
$$= \boldsymbol{X} \cdot \mathbf{W} + \boldsymbol{X} \wedge \mathbf{W}$$
$$= \boldsymbol{Y}_{\mathrm{r}} + \mathbf{Y}_{\mathbf{1}}\mathbf{e}_{\mathbf{1}} + \boldsymbol{Y}_{\mathbf{1}}\mathbf{e}_{\mathbf{2}} + \boldsymbol{Y}_{\mathbf{12}}\,\mathbf{e}_{\mathbf{1}}\mathbf{e}_{\mathbf{2}}$$

Real valued neural network for 1 channel calculation

$$\boldsymbol{Y}_{\mathrm{r}} = X_r W_r - X_1 W_1 - X_2 W_2 - X_{12} W_{12}$$

$$\boldsymbol{Y}_1 = X_r W_1 + X_1 W_r + W_2 W_{12} - W_{12} W_2$$

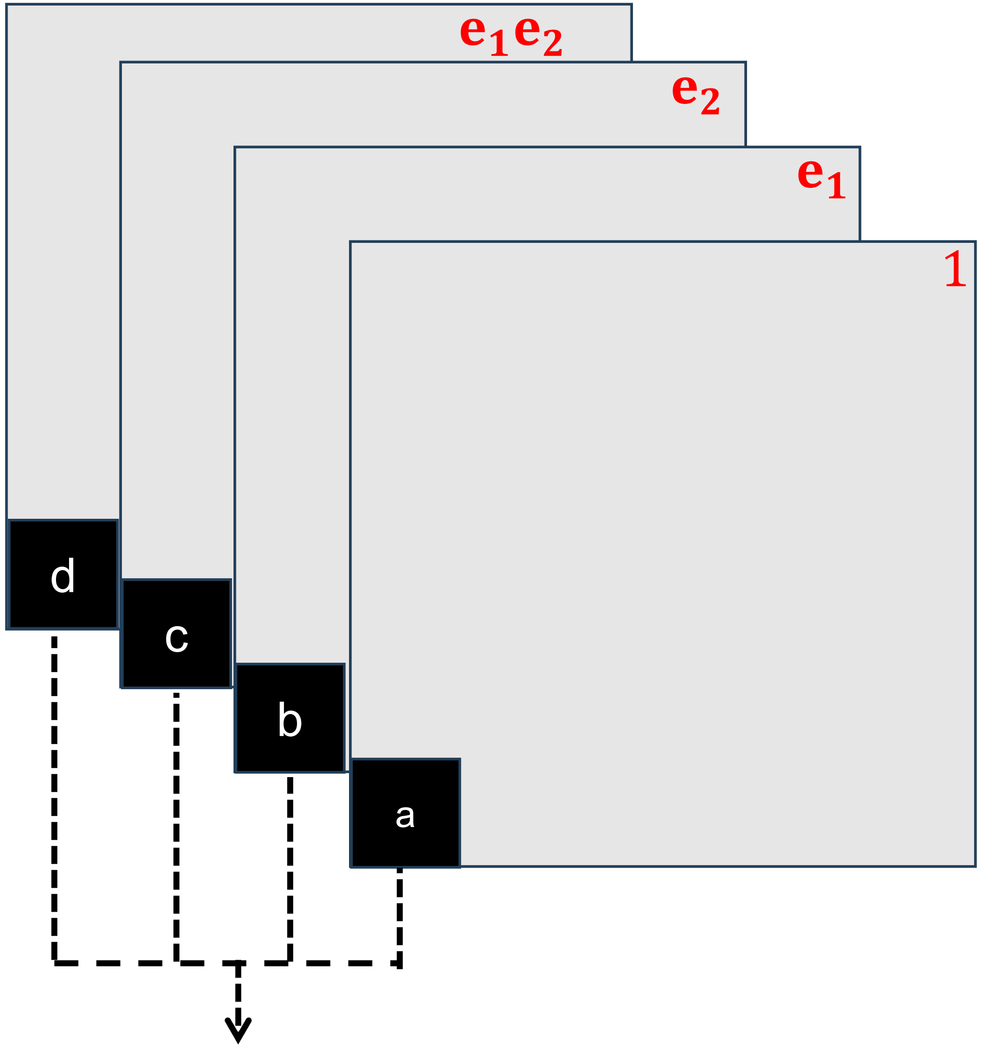
$$\boldsymbol{Y}_2 = X_r W_2 - X_1 W_{12} + W_2 W_r + W_{12} W_1$$

$$\boldsymbol{Y}_{12} = X_r W_3 + X_1 W_2 - W_2 W_1 + W_{12} W_r$$

Quaternion valued neural Network for 1 channel calculation.

View multidimensional data as multi-vector:

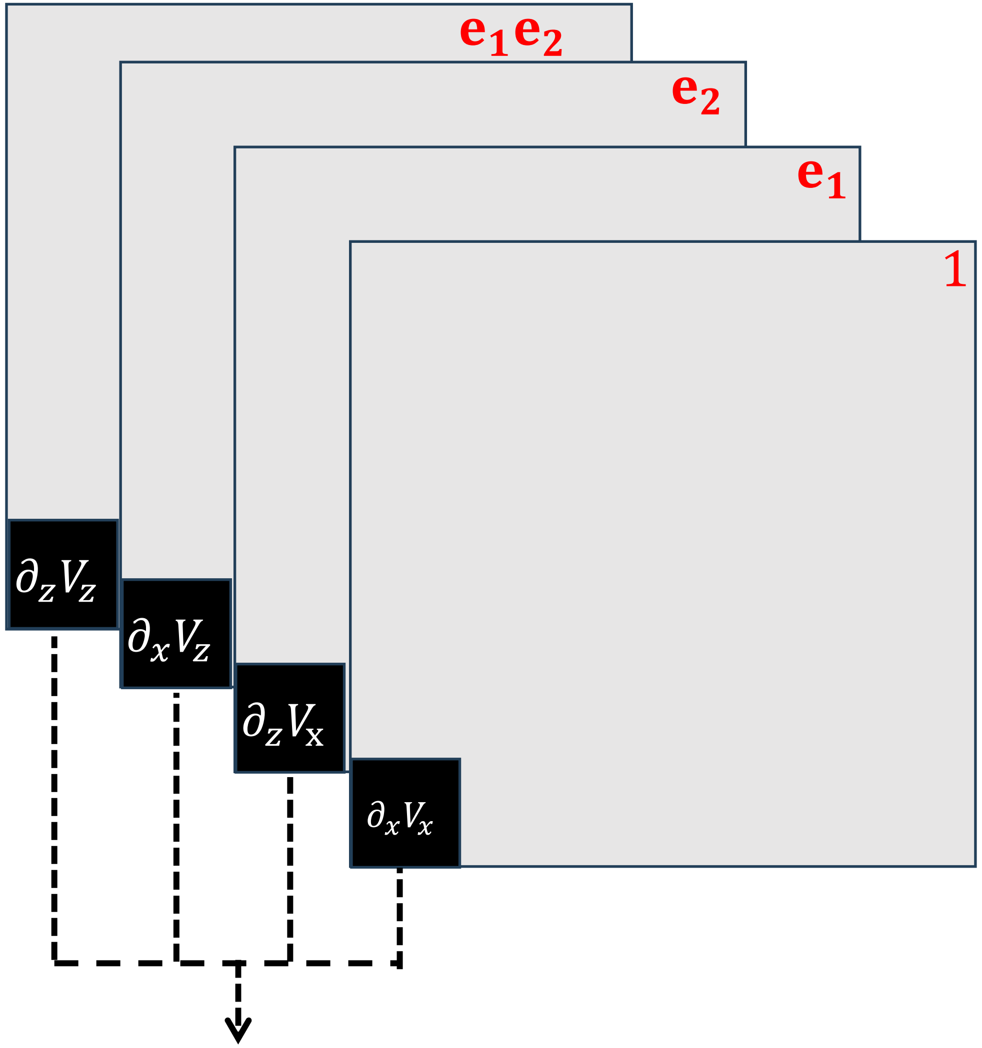
$$\mathbb{H}^{1\times 1\times N_z\times N_x\times 4}$$



$$h_l(1, z, x, 4) = a + b\,\mathbf{e_1} + c\,\mathbf{e_2} + d\,\mathbf{e_1}\,\mathbf{e_2}$$

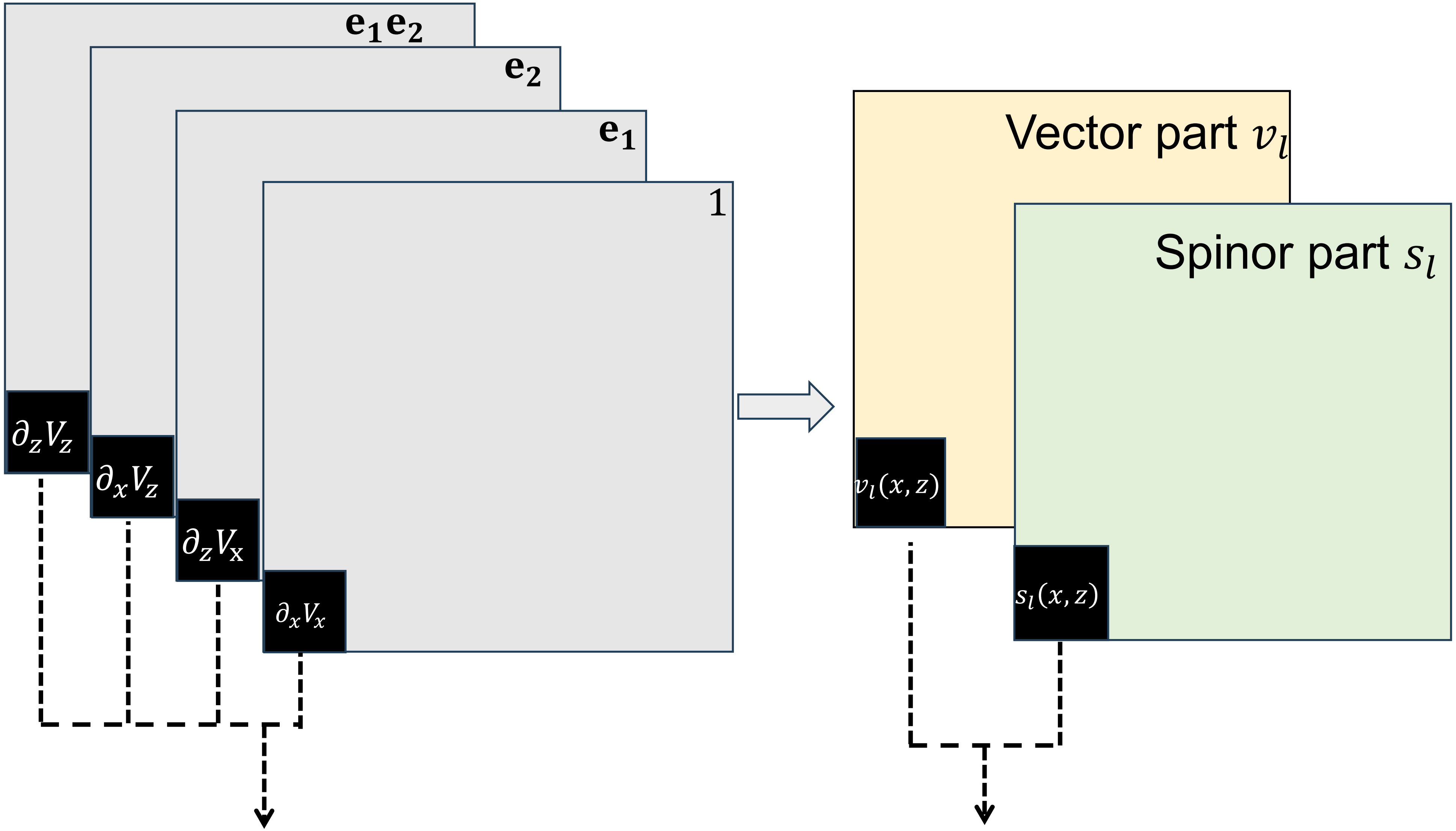
Partial derivative velocity wavefields viewed as Quaternions:

$$\mathbb{H}^{1\times 1\times N_z\times N_x\times 4}$$



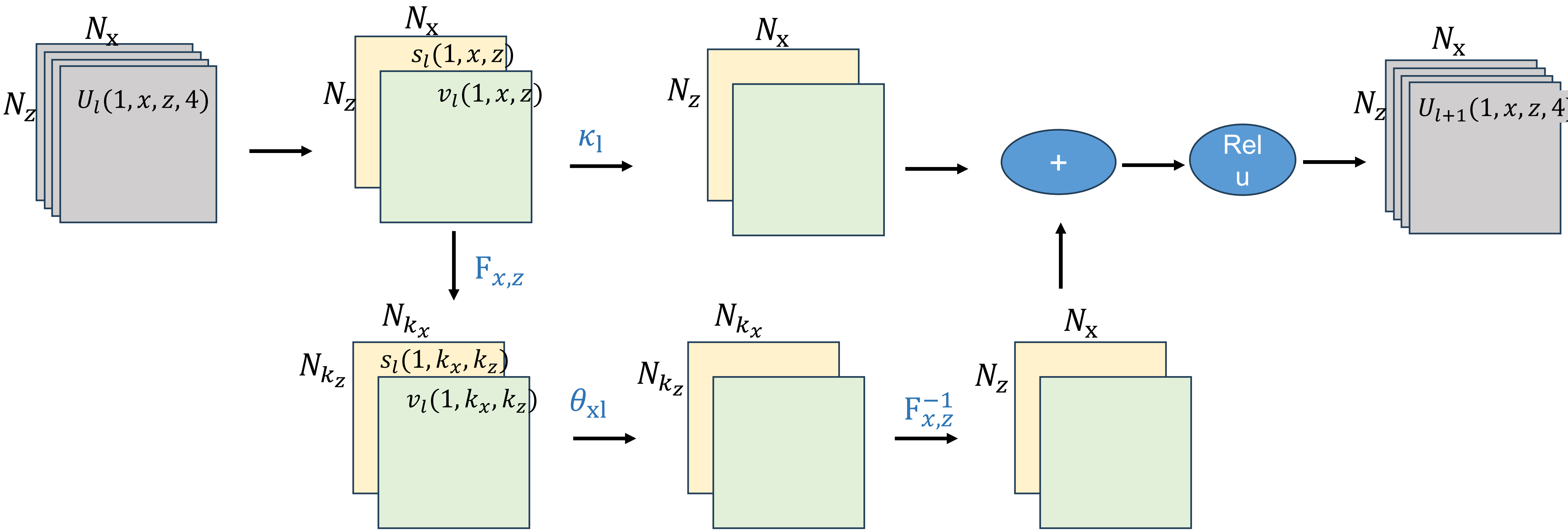
$$h_l(1,z,x,4) = \partial_x V_x + \partial_z V_x\,\mathbf{e_1} + \partial_x V_z\,\mathbf{e_2} + \partial_z V_z\,\mathbf{e_1}\,\mathbf{e_2}$$

2. Clifford(Quaternion) Fourier neural operators



$$U_l(1,x,z,4) = \partial_x V_x * 1 + \partial_z V_x \, \mathbf{e_1} + \partial_x V_z \, \mathbf{e_2} + \partial_z V_z \, \mathbf{e_1} \, \mathbf{e_2} = s_l(x,z) + \mathbf{e_1} \, v_l(x,z)$$

2. Clifford(Quaternion) Fourier neural operators



Part three Numerical training results

Examples of the training model

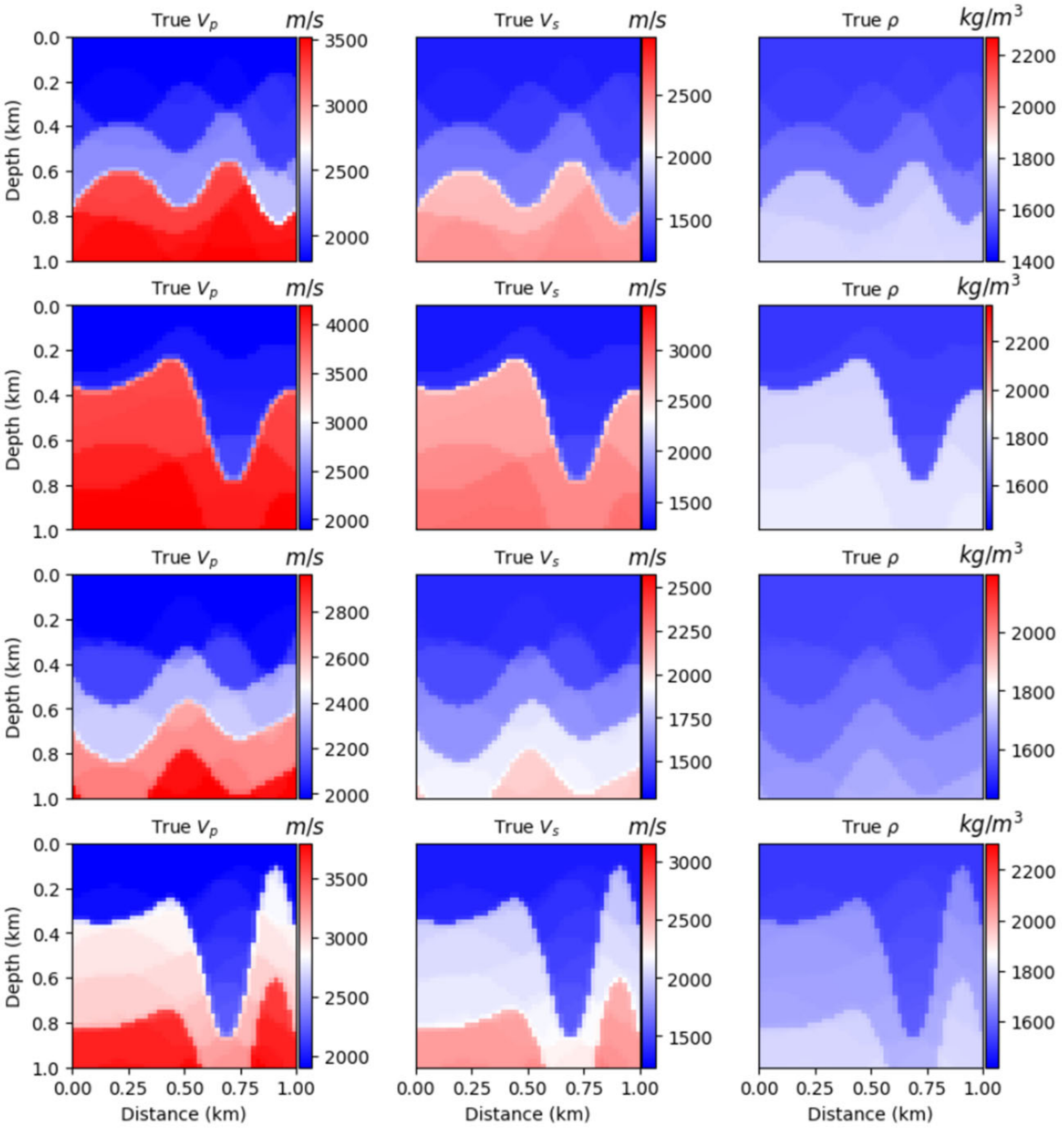
Obtained from the OpenFWI dataset,
Chengyuan Deng, et al. 2021

100 Vp models are used for training (Grid size 50×50)
Vs and density are obtained by scaling the Vp model.

10 models are used for validation.

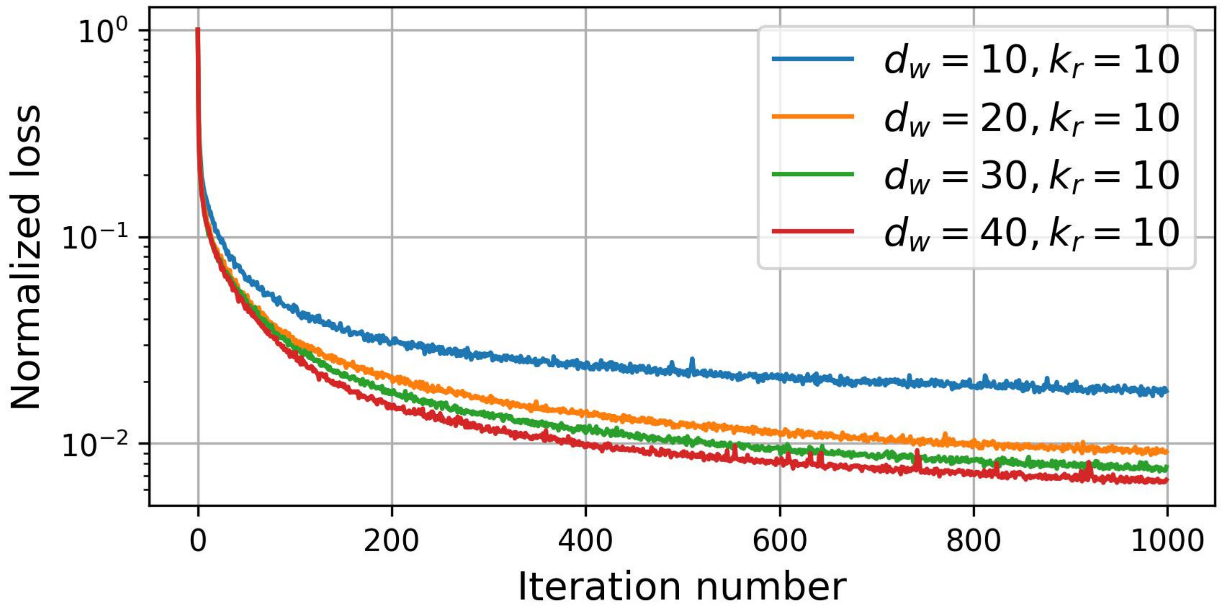
Labeled data (wavefields) are obtained with the finite difference method.
(10 order accuracy in space, 2 order accuracy in time)

The Clifford neural networks are developed by a Microsoft team.
Johannes, et al ICLR 2023

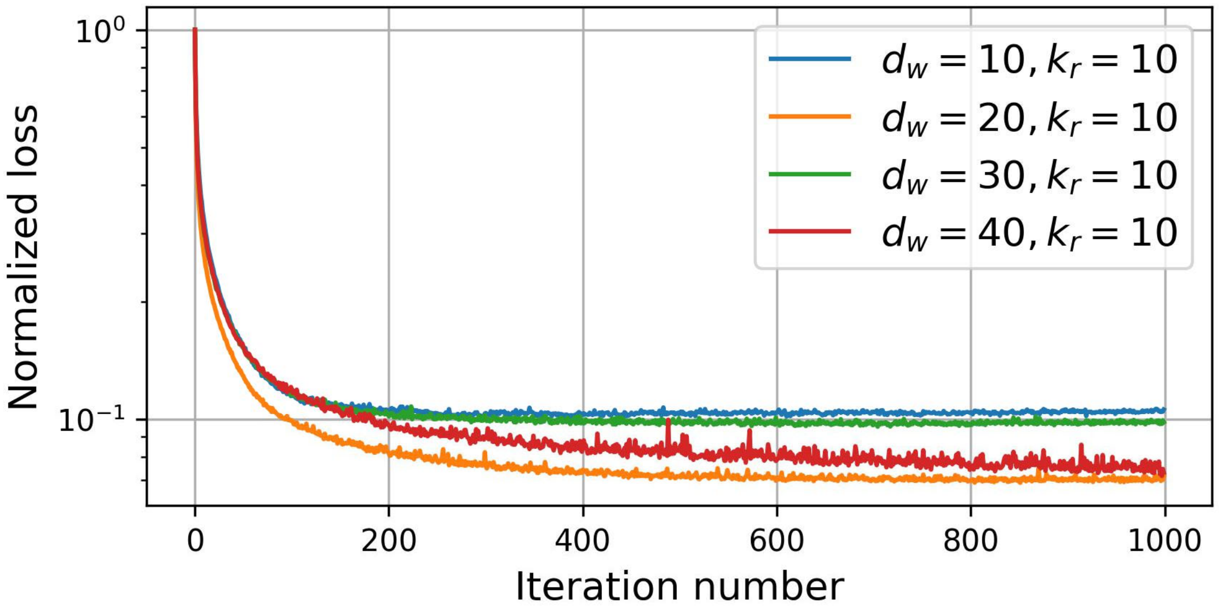


Training loss and validation loss with different d_w

Training loss

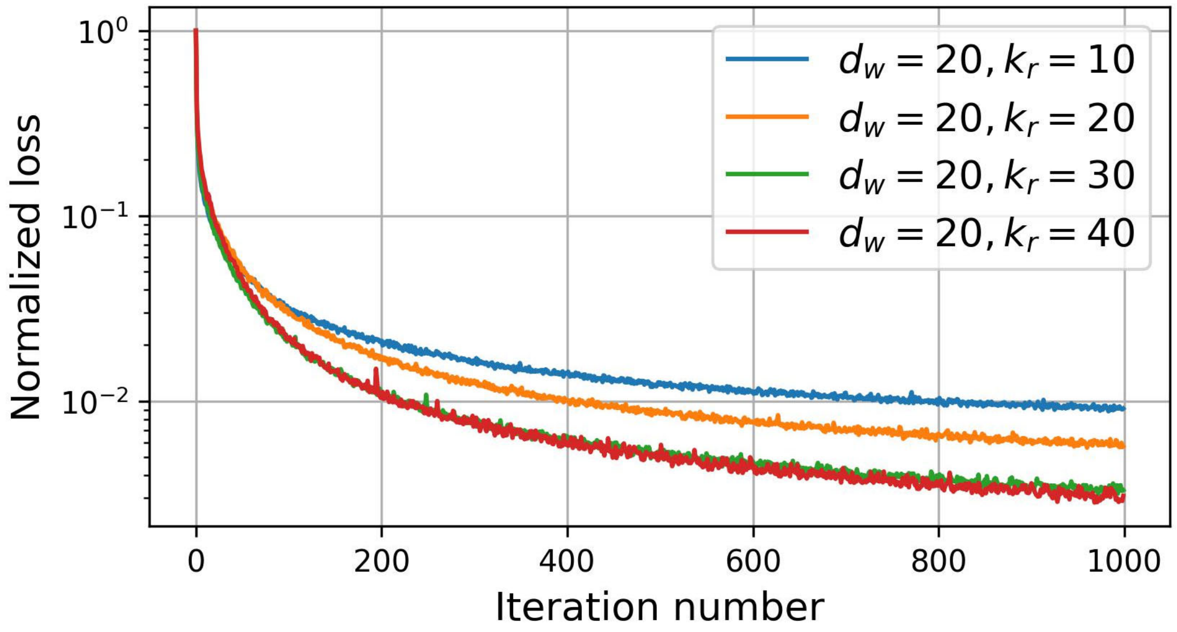


Validation loss

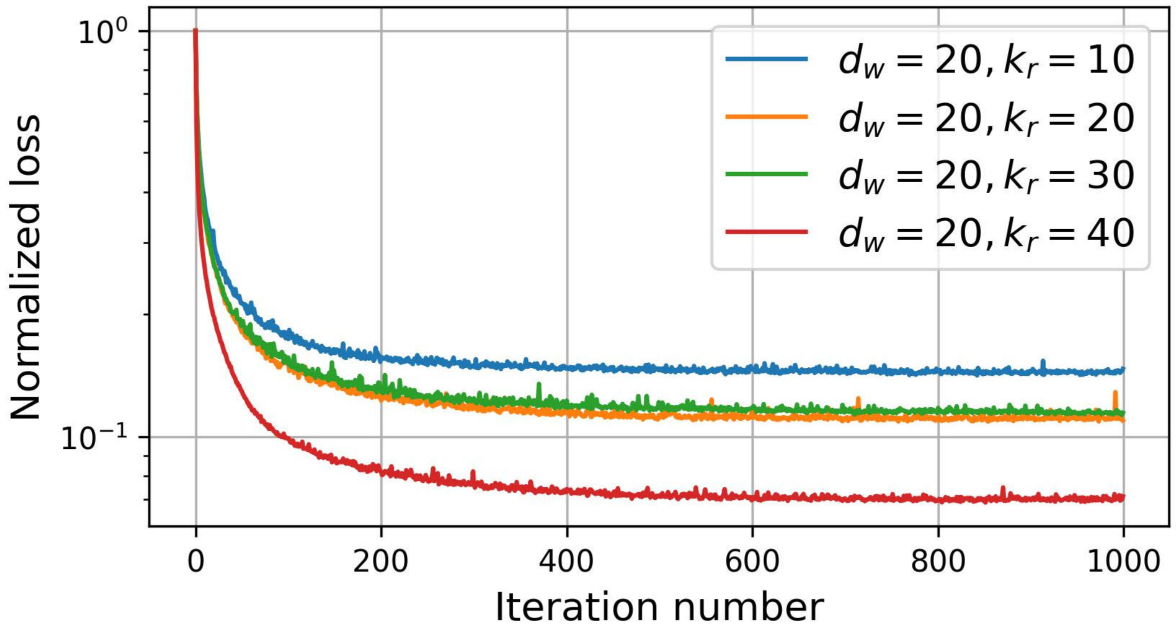


Training loss and validation loss with different k_r

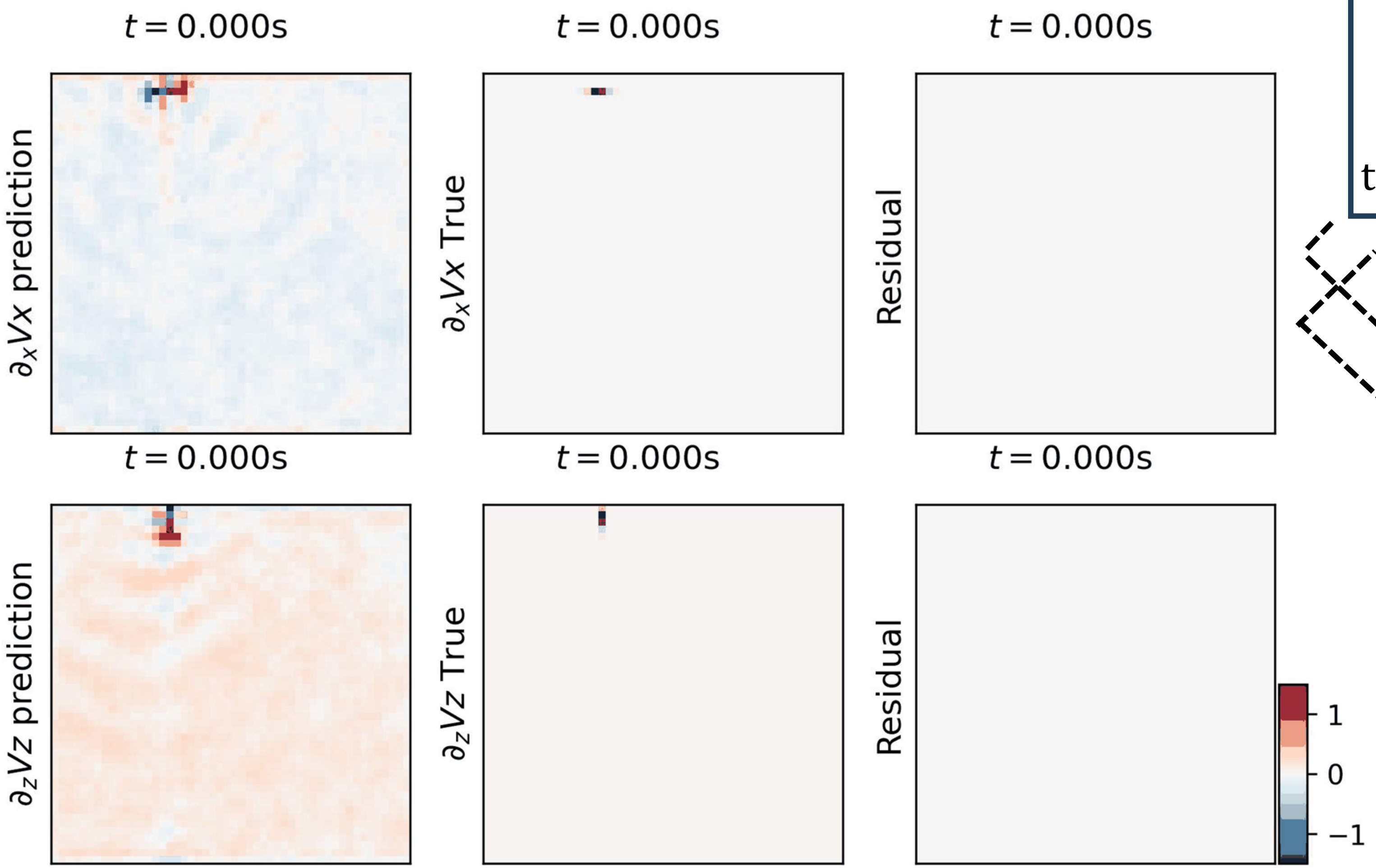
Training loss

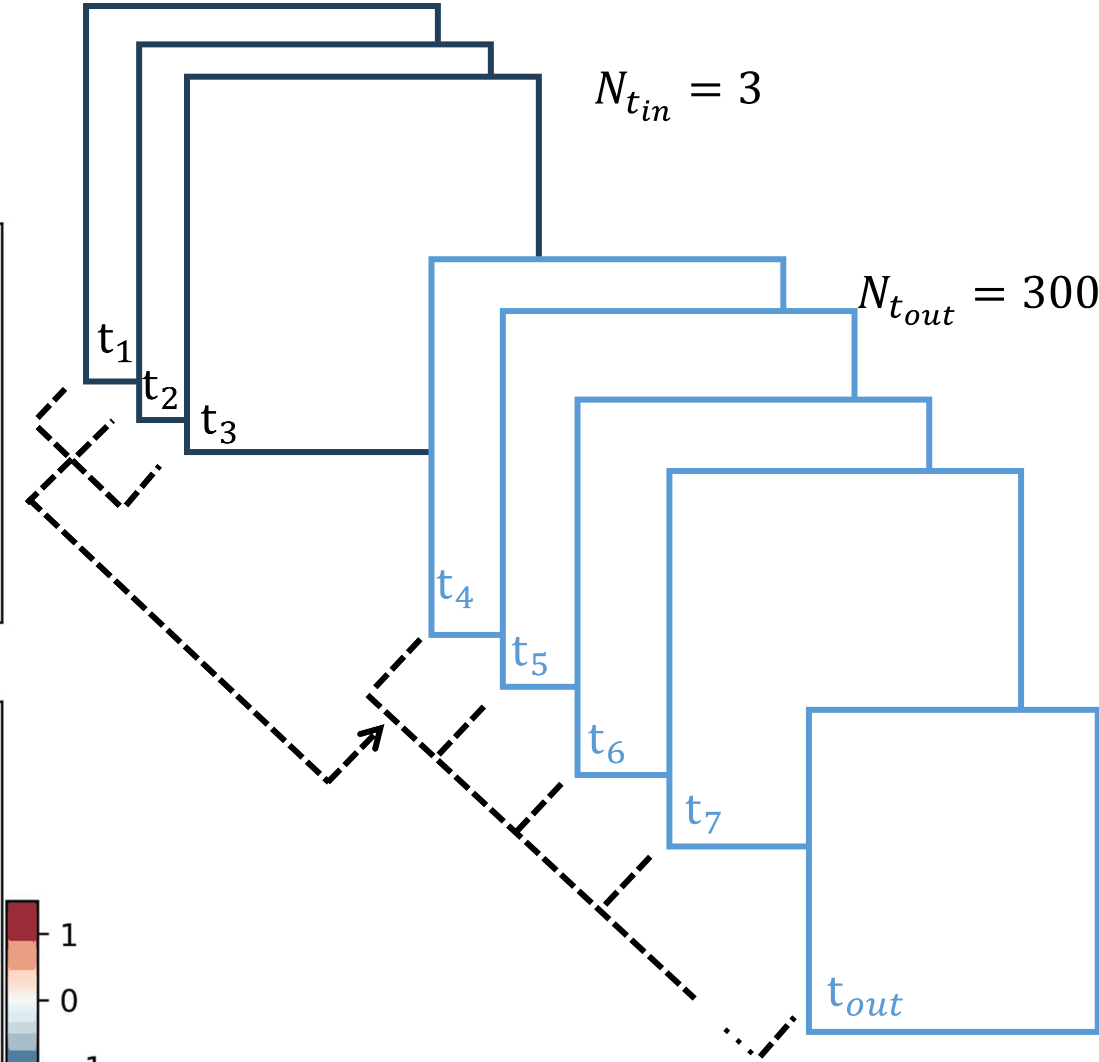


Validation loss

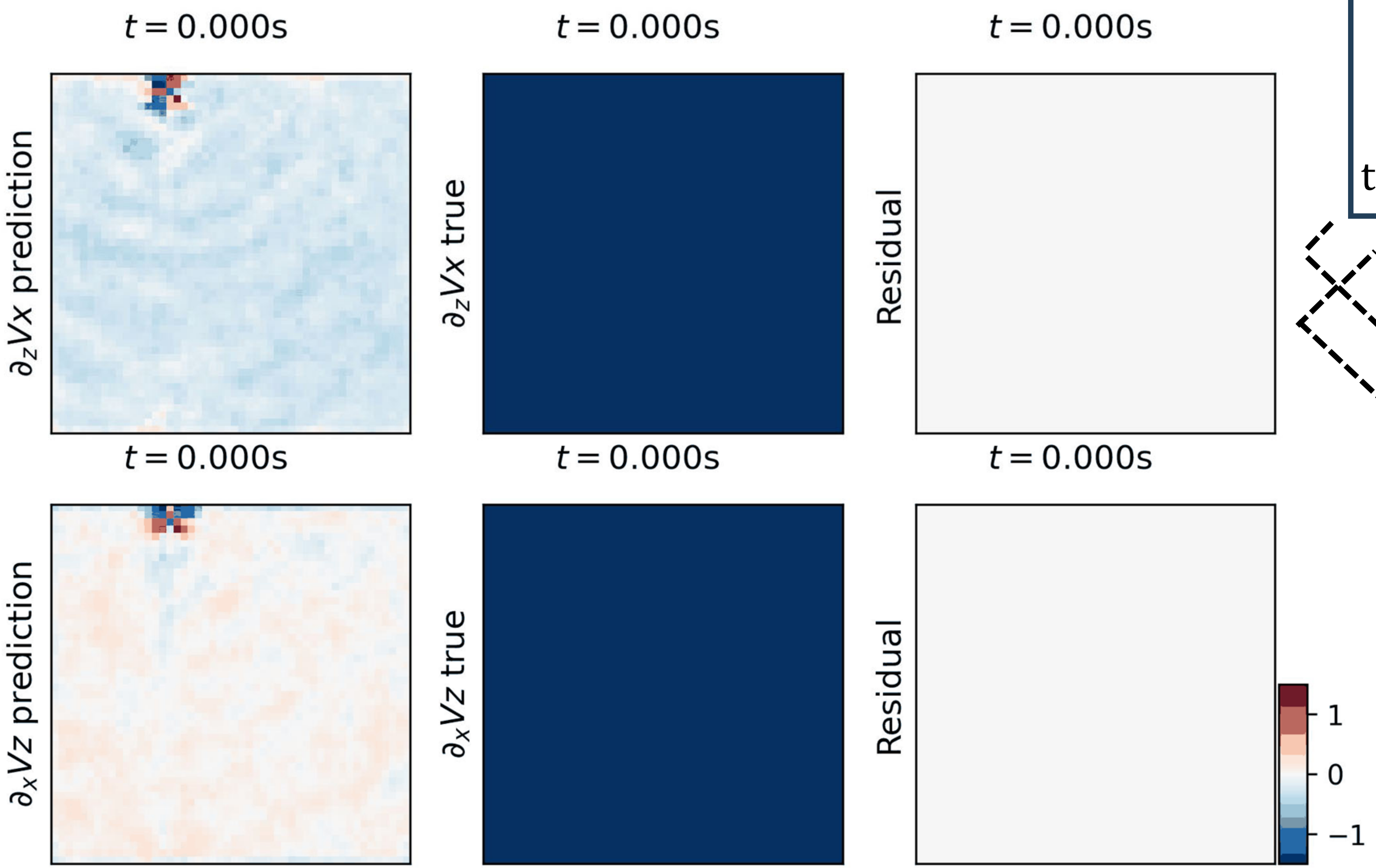


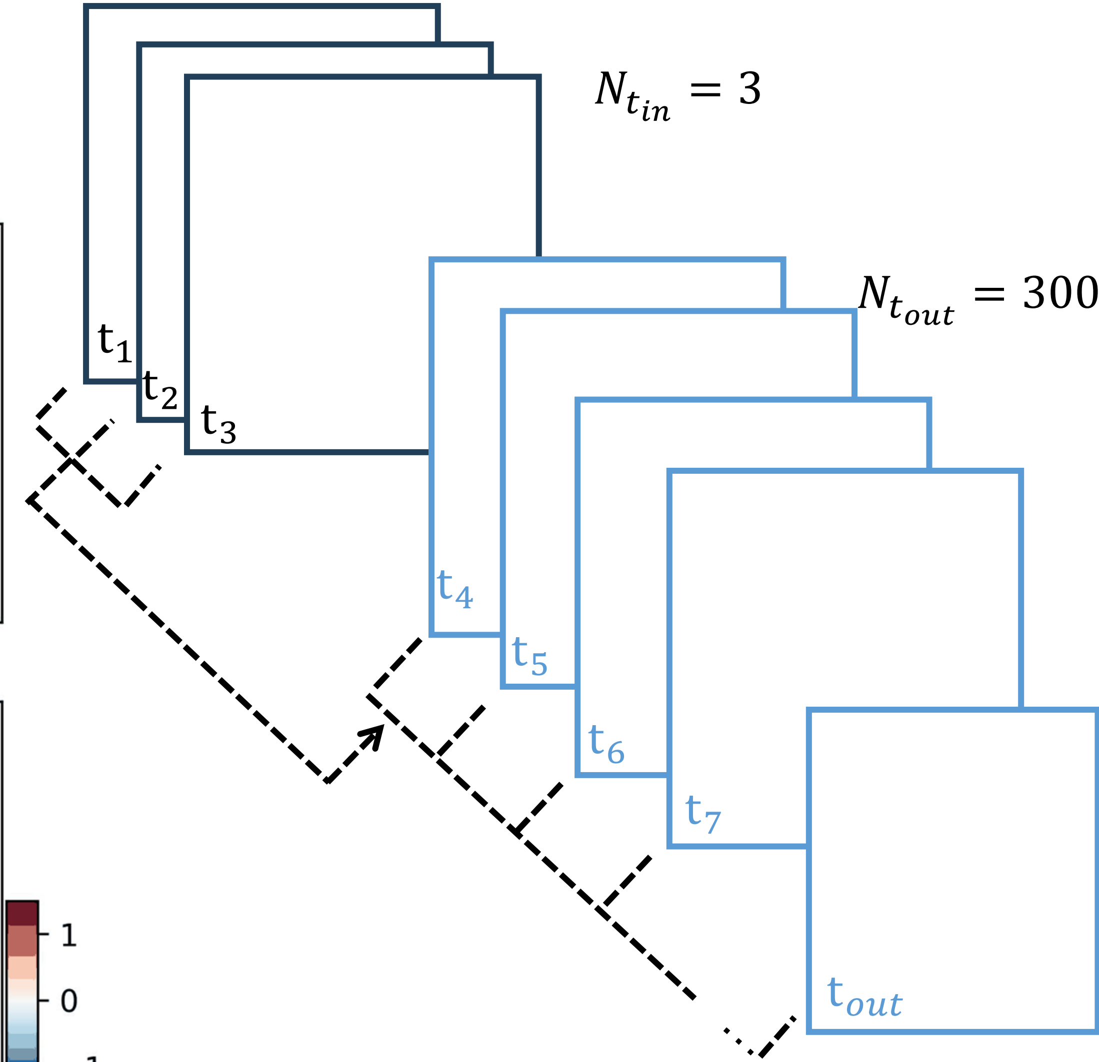
Rollout training



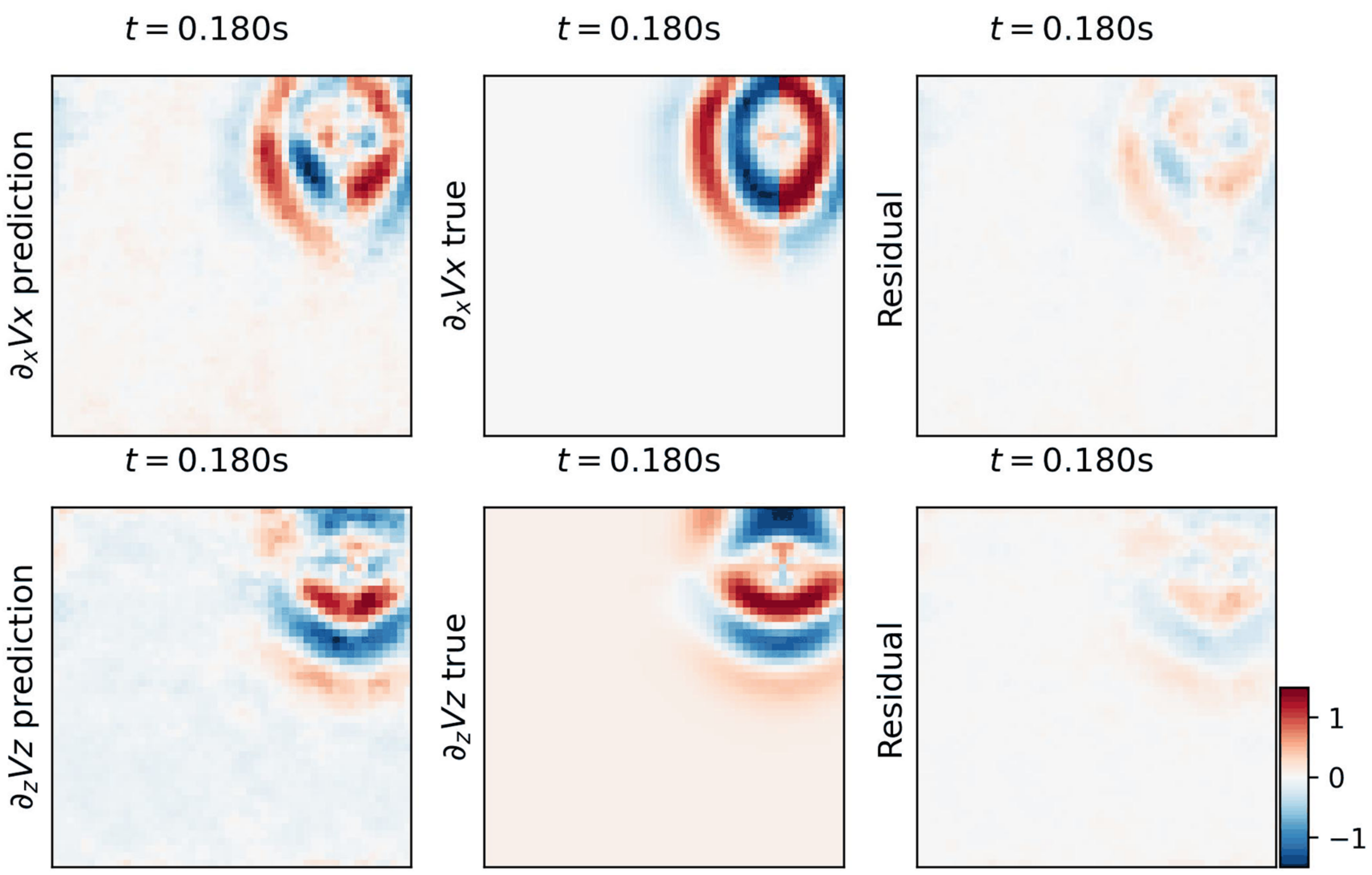


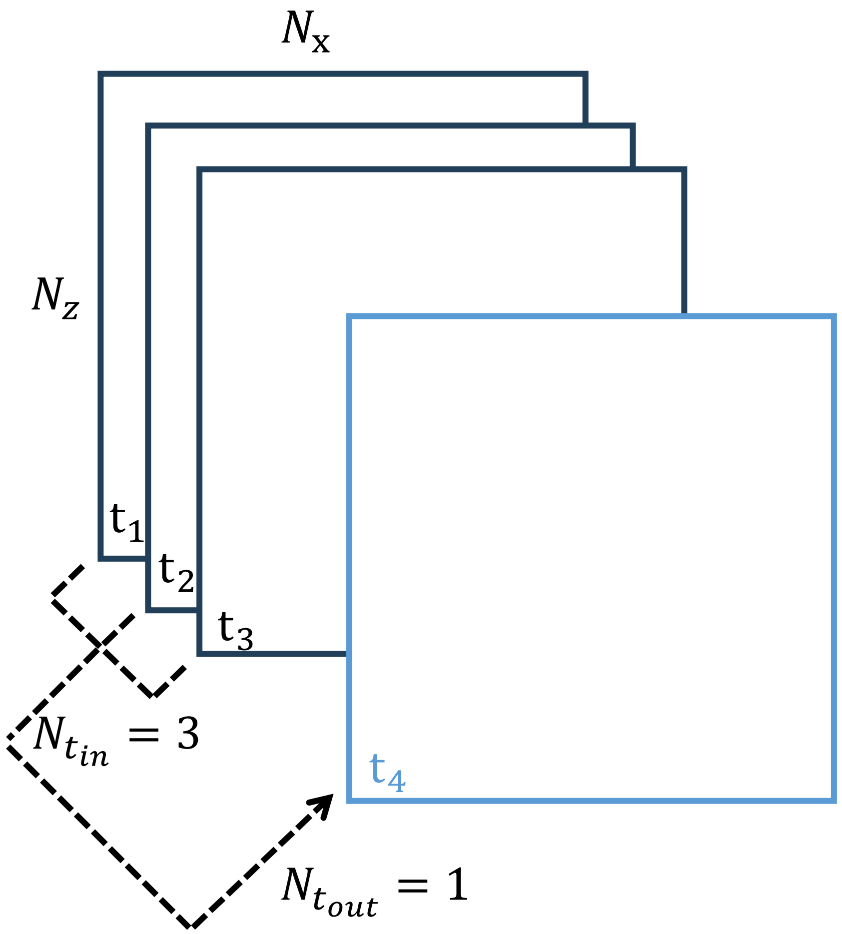
Rollout training



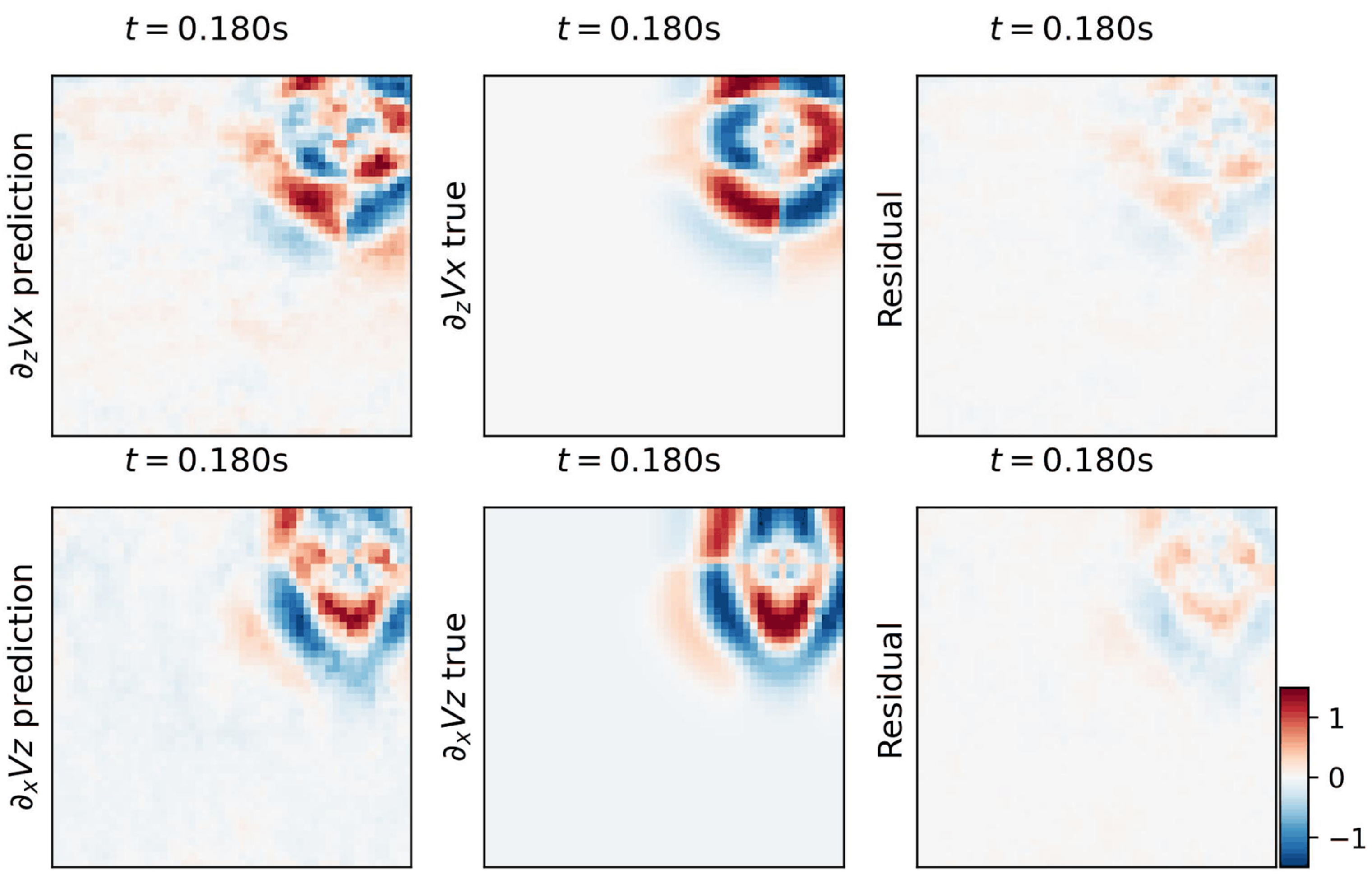


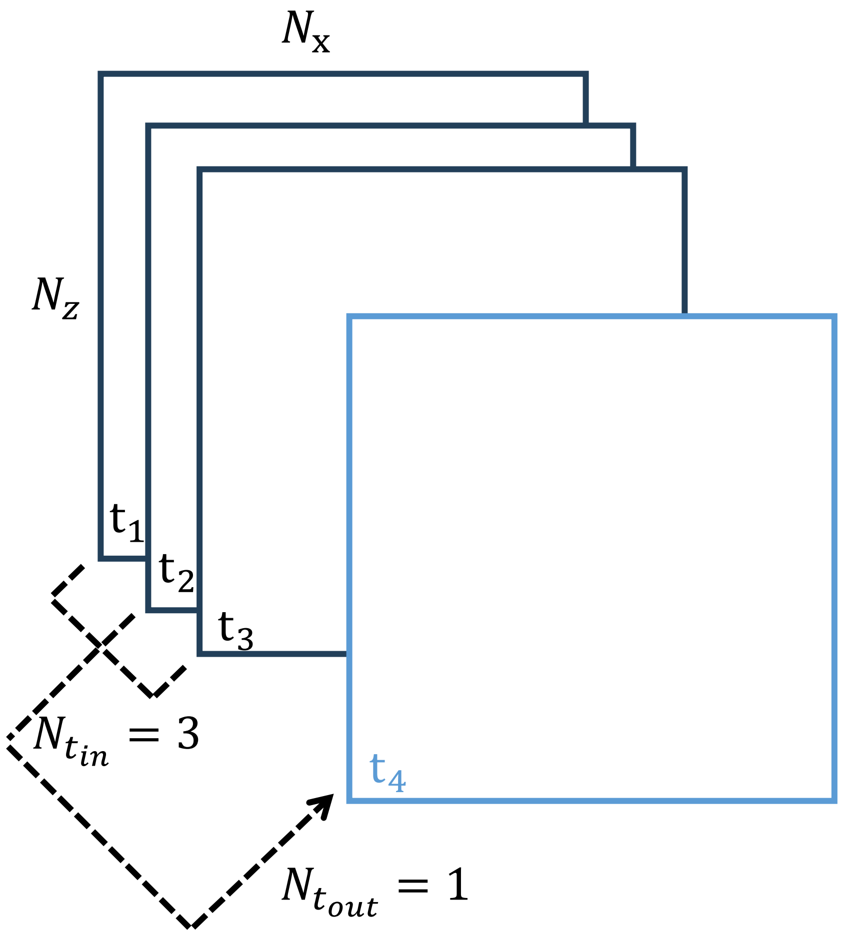
One-step training(Validation)





One-step training(Validation)





Conclusion

- Quaternion numbers are used to represent the multi-dimensional elastic wavefields.

- Clifford (quaternion) neural network is trained to learn to solve the elastic wave equation. The overall predicted results are promising.

Feature study

- Discover the mesh-free feature of the FNO and implement it into inversion.

- **Thanks, all CREWES sponsors and students**