中国人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

Clifford Neural Layers for PDE Modeling

Johannes Brandstetter[1], Rianne van den Berg[1], Max Welling[1], Jayesh K. Gupta[2]

[1]Microsoft Research Amsterdam

[2]Microsoft Autonomous Systems and Robotics Research

Angxiao Yue

2023-5-31

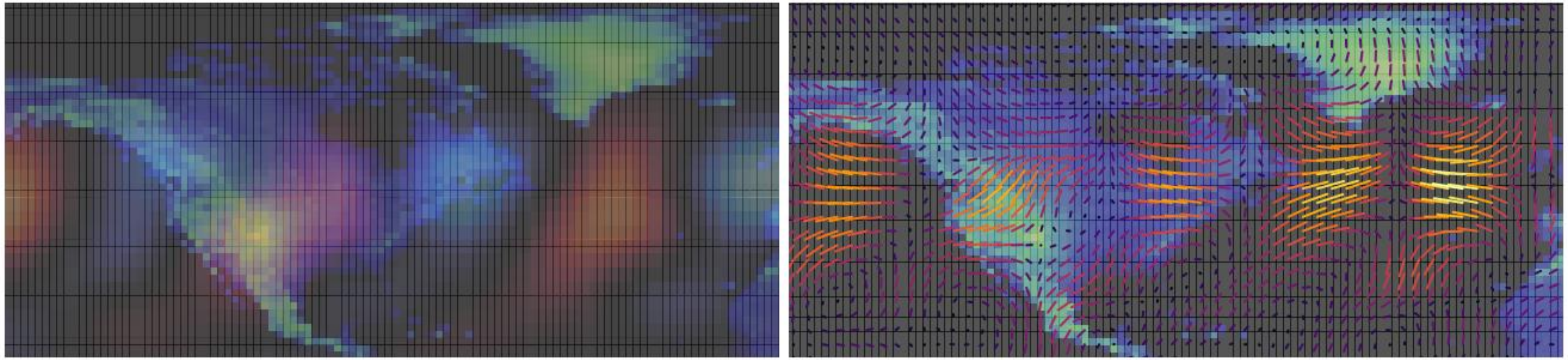
Outline

- Introduction
- Background: Clifford Algebras
- Method: Clifford Neural Layers
- Experiment
- Conclusion

- Introduction
- Background: Clifford Algebras
- Method: Clifford Neural Layers
- Experiment
- Conclusion

Introduction

- Most scientific phenomena are described by the evolution and interaction of physical quantities over space and time. Fluid mechanics, Electromagnetism
- The underlying equations are famously described in various forms of the Navier-Stokes equations and Maxwell's equations. Analytically intractable, Time-consuming
- Deep learning's success at overcoming the curse of dimensionality in many fields has led to a surge of interest in scientific applications. Current approaches treat vector field components the same as scalar fields

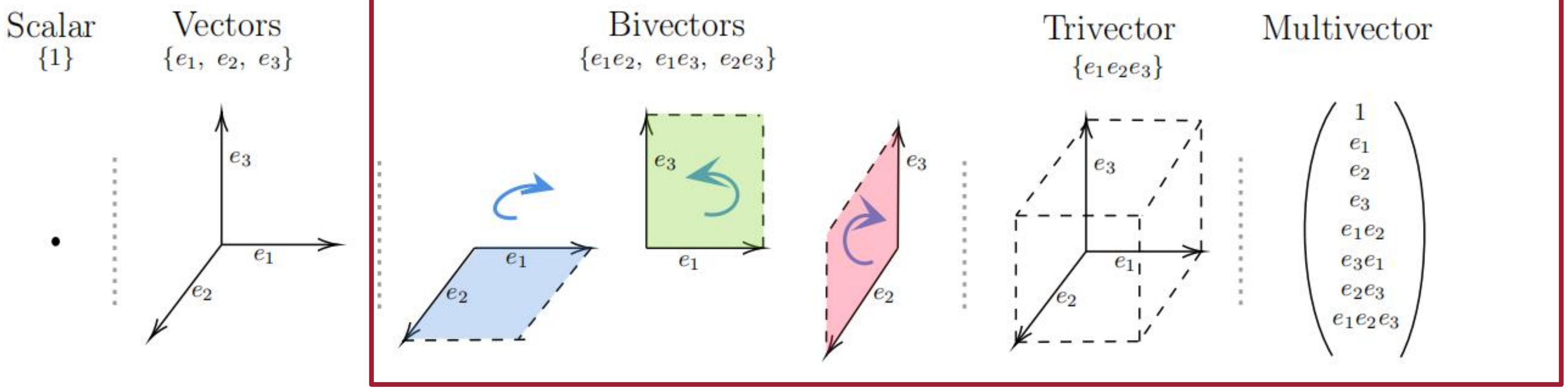


(a) Scalar pressure field (b) Vector wind velocity field

Introduction

- This work builds **Clifford** Neural PDE surrogates which model the relation between different **fields** (e.g. wind and pressure field) and **field components** (e.g. � - and � - components of the wind velocities).
  - Propose Clifford convolution and Clifford Fourier transforms over multivector fields.

    Scalar, Vector → Multivector
  - Clifford neural layers consistently improve the generalization capabilities of the tested architectures.



- Introduction
- Background: Clifford Algebras
- Method: Clifford Neural Layers
- Experiment
- Conclusion

Background: Clifford Algebras

■ Clifford algebra $Cl_{p,q}(\mathbb{R})$

- $Cl_{p,q}(\mathbb{R})$ is generated by $p+q$ orthonormal basis elements $e_1,\ e_2,\ \ldots,\ e_{p+q}$ such that the following quadratic relations hold:

$$
\begin{aligned}
e_i^2 &= +1 \quad \text{for} \quad 1 \le i \le p, \\
e_j^2 &= -1 \quad \text{for} \quad p < j \le p+q, \\
e_i e_j &= -e_j e_i \quad \text{for} \quad i \ne j.
\end{aligned}
$$

- The pair $(p,\ q)$ is called the signature and defines a Clifford algebra $Cl_{p,q}(\mathbb{R})$, together with the basis elements that span the vector space G^{p+q} of $Cl_{p,q}(\mathbb{R})$.
- $Cl_{0,0}(\mathbb{R})$ is spanned by the basis element $\{1\}$. Isomorphic to $\mathbb{R}$.
- $Cl_{0,1}(\mathbb{R})$ is spanned by $\{1,\ e_1\}$. Isomorphic to $\mathbb{C}$.
- $Cl_{0,2}(\mathbb{R})$ is spanned by $\{1,\ e_1, e_2, e_1e_2\}$. Isomorphic to quaternions $\mathbb{H}$.

Background: Clifford Algebras

■ **Grade, Pseudoscalar, Dual, Geometric product**

- The grade of a Clifford algebra basis element is the dimension of the subspace it represents. For example, the basis elements $\{1,\ e_1, e_2, e_1e_2\}$ of $Cl_{0,2}(\mathbb{R})$ have the grades $\{0,\ 1,\ 1,\ 2\}$.
- The grade subspace of smallest dimension is M_0, the subspace of all scalars. Elements of M_1 are called vectors, elements of M_2 are bivectors, and so on. The vector space G^{p+q} of a Clifford algebra $Cl_{p,q}(\mathbb{R})$ can be written as the direct sum of all of these subspaces:

$$G^{p+q} = M_0 \oplus M_1 \oplus \ldots M_{p+q}$$

- The basis element with the highest grade is called the pseudoscalar i_{p+q}, which in $Cl_{0,2}(\mathbb{R})$ is $i_2 = e_1e_2$.

Background: Clifford Algebras

■ Grade, Pseudoscalar, Dual, Geometric product

➢ The geometric product is the bilinear operation on multivectors in Clifford algebras. For arbitrary multivectors $a,\ b,\ c \in G^{p+q}$, and scalar λ:

$ab \in G^{p+q}$	closure,
$(ab)c = a(bc)$	associativity,
$\lambda a = a\lambda$	commutative scalar multiplication,
$a(b+c) = ab + ac$	distributive over addition.
$\boldsymbol{ab \neq ba}$	**non-commutative**

➢ The dual a^* of a multivector a is defined as

$$a^* = a i_{p+q}$$

where i_{p+q} represents the respective pseudoscalar of the Clifford algebra. This definition allows us to relate different multivectors to each other. For quaternion, a dual of scalar is bivector $e_1 e_2$ or $\boldsymbol{k}$.

Background: Clifford Algebras

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

$$
\begin{aligned}
\boldsymbol{ab} &= a_0b_0 - a_1b_1 - a_2b_2 - a_{12}b_{12} \\
&\quad + (a_0b_1 + a_1b_0 + a_2b_{12} - a_{12}b_2)\, e_1 \\
&\quad + (a_0b_2 - a_1b_{12} + a_2b_0 + a_{12}b_1)\, e_2 \\
&\quad + (a_0b_{12} + a_1b_2 - a_2b_1 + a_{12}b_0)\, e_1e_2
\end{aligned}
$$

Hamilton Product

  - The dual pairs of the base vectors are $1 \leftrightarrow e_1e_2,\ \ e_1 \leftrightarrow e_2$. We rewrite a

$$\boldsymbol{a} = a_0 + a_1e_1 + a_2e_2 + a_{12}e_{12}\ ,$$

commutes with i_2

$$\boldsymbol{a} = 1\underbrace{(a_0 + a_{12}i_2)}_{\text{spinor part}} + e_1\underbrace{(a_1 + a_2i_2)}_{\text{vector part}}\ ,$$

anti-commutes with i_2

- Introduction
- Background: Clifford Algebras
- Method: Clifford Neural Layers
- Experiment
- Conclusion

Method: Clifford Neural Layers

■ Clifford CNN layers

- Regular convolutional neural network (CNN) layers take as input feature maps $f\colon \mathbb{Z}^2 \longrightarrow \mathbb{R}^{c_{in}}$ and convolve them with a set of c_{out} filters $\{w^i\}_{i=1}^{c_{out}}$ with $w^i\colon \mathbb{Z}^2 \longrightarrow \mathbb{R}^{c_{in}}$:

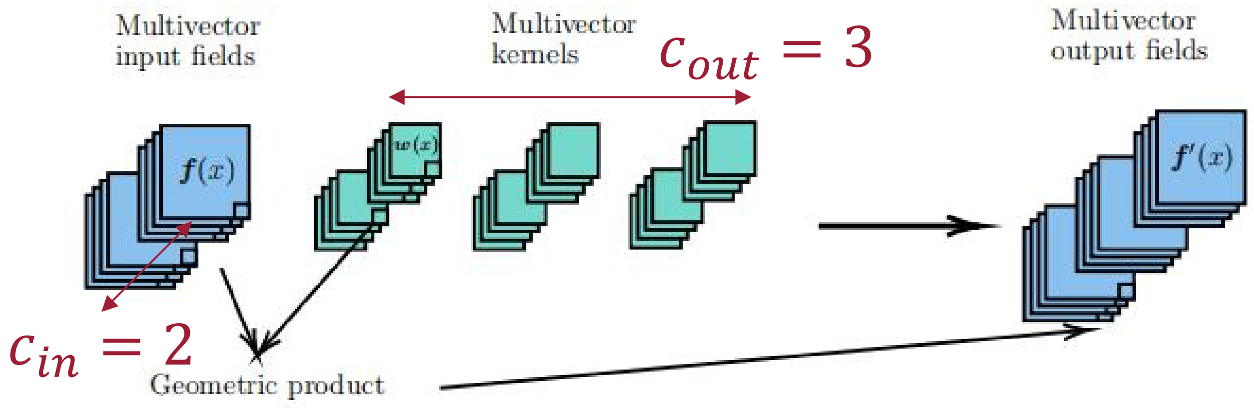
$$[f \star w_i](x) = \sum_{y \in \mathbb{Z}^2} \langle f(y), w^i(y-x) \rangle = \sum_{y \in \mathbb{Z}^2} \sum_{j=1}^{c_{\text{in}}} f^j(y) w^{i,j}(y-x) \ .$$

- $f^j(y)w^{i,j}(y-x)$ is replaced by the geometric product of multivector $\boldsymbol{f}^j(y)\boldsymbol{w}^{i,j}(y-x)$. Feature maps $\boldsymbol{f}\colon \mathbb{Z}^2 \longrightarrow (G^2)^{c_{in}}$ and filters $\{\boldsymbol{w}^i\}_{i=1}^{c_{out}}$ with $\boldsymbol{w}^i\colon \mathbb{Z}^2 \longrightarrow (G^2)^{c_{in}}$.

$$[\boldsymbol{f} \star \boldsymbol{w}^i](x) = \sum_{y \in \mathbb{Z}^2} \sum_{j=1}^{c_{\text{in}}} \underbrace{\boldsymbol{f}^j(y)\boldsymbol{w}^{i,j}(y-x)}_{\boldsymbol{f}^j \boldsymbol{w}^{i,j} \ : \ G^2 \times G^2 \to G^2} \ .$$

Method: Clifford Neural Layers

■ Clifford CNN layers



$$\left[\boldsymbol{f} \star \boldsymbol{w}^i\right](x) = \sum_{y \in \mathbb{Z}^2} \sum_{j=1}^{c_{\text{in}}} \underbrace{\boldsymbol{f}^j(y)\boldsymbol{w}^{i,j}(y-x)}_{\boldsymbol{f}^j\boldsymbol{w}^{i,j}\ :\ G^2 \times G^2 \to G^2} .$$

$$\begin{aligned} \boldsymbol{ab} &= a_0b_0 - a_1b_1 - a_2b_2 - a_{12}b_{12} \\ &+ (a_0b_1 + a_1b_0 + a_2b_{12} - a_{12}b_2)\, e_1 \\ &+ (a_0b_2 - a_1b_{12} + a_2b_0 + a_{12}b_1)\, e_2 \\ &+ (a_0b_{12} + a_1b_2 - a_2b_1 + a_{12}b_0)\, e_1e_2 \end{aligned}$$

➢ $\{b_0, b_1, b_2, b_{12}\} \longrightarrow \{w_0^{i,j}, w_1^{i,j}, w_2^{i,j}, w_{12}^{i,j}\}$ correspond to 4 different kernels representing one 2D multivector kernel. $\{a_0, a_1, a_2, a_{12}\} \longrightarrow \{f_0^j, f_1^j, f_2^j, f_{12}^j\}$

Method: Clifford Neural Layers

高瓴人工智能学院
Gaoling School of Artificial Intelligence

Rotational Clifford CNN layers

- Represent the feature map $\boldsymbol{f}^j$ and filter $\boldsymbol{w}^{i,j}$ as quaternions: $\boldsymbol{f}^j = f_0^j + f_1^j\boldsymbol{i} + f_2^j\boldsymbol{j} + f_{12}^j\boldsymbol{k}$ and $\boldsymbol{w}^{i,j} = w_0^{i,j} + w_1^{i,j}\boldsymbol{i} + w_2^{i,j}\boldsymbol{j} + w_{12}^{i,j}\boldsymbol{k}$.
- A quaternion rotation can be realized by a matrix multiplication. $\boldsymbol{w}^{i,j}\boldsymbol{f}^j(\boldsymbol{w}^{i,j})^{-1}$ can be manipulated into a vector-matrix operation $\boldsymbol{R}^{i,j}\boldsymbol{f}^j$, where $\boldsymbol{R}^{i,j}$ is built from the elements of $\boldsymbol{w}^{i,j}$.

$$\left[\boldsymbol{f} \star \boldsymbol{w}_{\mathrm{rot}}^{i}\right](x) = \sum_{y\in\mathbb{Z}^2}\sum_{j=1}^{c_{\mathrm{in}}} \boldsymbol{f}^j(y)\boldsymbol{w}_{\mathrm{rot}}^{i,j}(y-x) = \sum_{y\in\mathbb{Z}^2}\sum_{j=1}^{c_{\mathrm{in}}} \underbrace{\left[\boldsymbol{f}^j(y)\boldsymbol{w}_{\mathrm{rot}}^{i,j}(y-x)\right]_0}_{\text{scalar output}} + \boldsymbol{R}^{i,j}(y-x)\cdot\begin{pmatrix} f_1^j(y) \\ f_2^j(y) \\ f_{12}^j(y)\end{pmatrix}$$

$$\left[\boldsymbol{f}^j(y)\boldsymbol{w}_{\mathrm{rot}}^{i,j}(y-x)\right]_0 = f_0^j w_{\mathrm{rot},0}^{i,j} - f_1^j w_{\mathrm{rot},1}^{i,j} - f_2^j w_{\mathrm{rot},2}^{i,j} - f_{12}^j w_{\mathrm{rot},12}^{i,j}$$

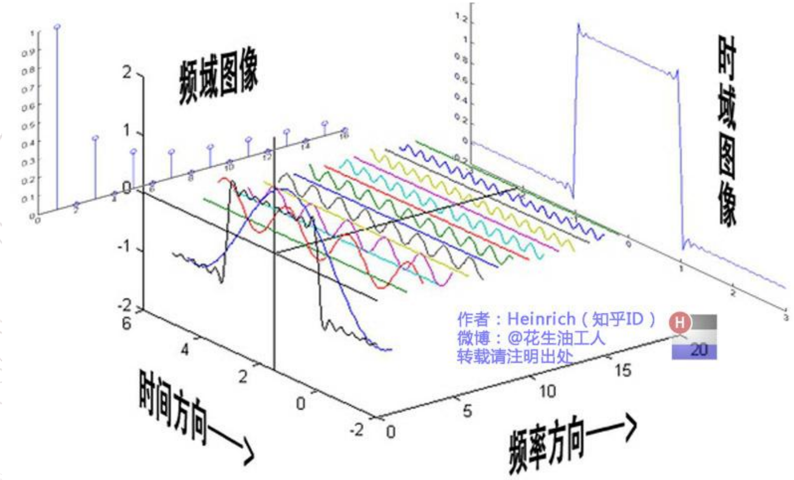
$Re(q_1 \otimes q_2) = s_1 s_2 - < \boldsymbol{v}_1, \boldsymbol{v}_2 >$

Method: Clifford Neural Layers

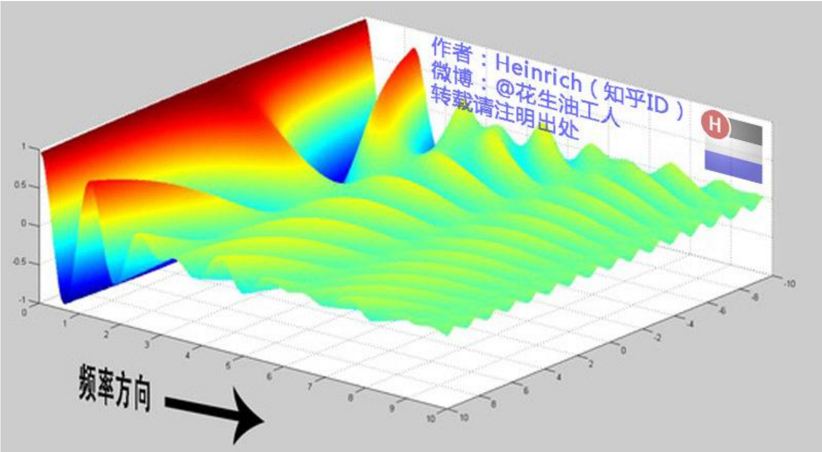
■ Fourier Neural Operator (FNO)

➢ In arbitrary dimension n, the discrete Fourier transform of an n-dimensional complex signal $f(x) = f(x_1, x_2, \ldots, x_n)$: $\mathbb{R}^n \longrightarrow \mathbb{C}$

$$\hat{f}(\xi_1, \ldots, \xi_n) = \mathcal{F}\{f\}(\xi_1, \ldots, \xi_n) = \sum_{m_1=0}^{M_1} \ldots \sum_{m_n=0}^{M_n} f(m_1, \ldots, m_n) \cdot e^{-2\pi i \cdot \left(\frac{m_1 \xi_1}{M_1} + \ldots + \frac{m_n \xi_n}{M_n}\right)},$$



Fourier Series



Fourier Transformation

Method: Clifford Neural Layers

■ Fourier Neural Operator (FNO)

➢ The 2-dimensional Clifford Fourier transform and the respective inverse transform for multivector valued functions $\boldsymbol{f}(\boldsymbol{x})$: $\mathbb{R}^2 \longrightarrow G^2$ and vectors $x,\ \xi \in \mathbb{R}^2$

$$\hat{\boldsymbol{f}}(\xi) = \mathcal{F}\{\boldsymbol{f}\}(\xi) = \frac{1}{2\pi}\int_{\mathbb{R}_2} \boldsymbol{f}(x)e^{-2\pi i_2 \langle x,\xi\rangle}\, dx \ , \ \forall \xi \in \mathbb{R}^2 \ ,$$

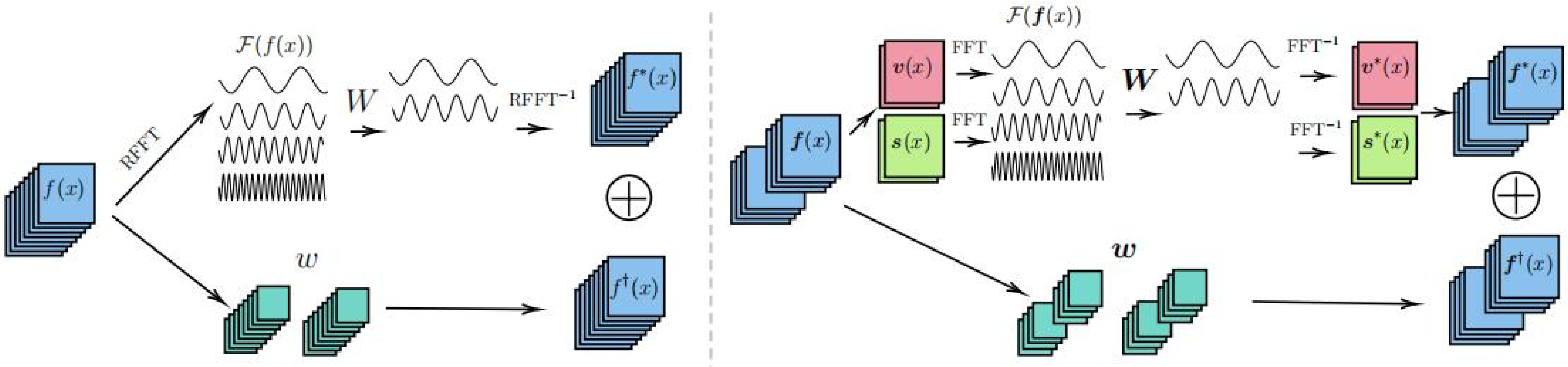
$$\boldsymbol{f}(x) = \mathcal{F}^{-1}\{\mathcal{F}\{\boldsymbol{f}\}\}(x) = \frac{1}{2\pi}\int_{\mathbb{R}_2} \hat{\boldsymbol{f}}(\xi)e^{2\pi i_2 \langle x,\xi\rangle}\, d\xi \ , \ \forall x \in \mathbb{R}^2 \ ,$$

f_0 and f_1 are two standard Fourier transform

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

Method: Clifford Neural Layers

- Fourier Neural Operator (FNO)



(a) FNO layer

(b) Clifford FNO layer

- Introduction
- Background: Clifford Algebras
- Method: Clifford Neural Layers
- **Experiment**
- Conclusion

Experiment

Experiment setting

- Change every convolution and every Fourier transform to their respective Clifford operation to get ResNet and Fourier Neural Operator (FNO) based architectures by their respective Clifford counterparts.
- Evaluate different training set sizes, and report losses for scalar and vector fields.
  - The one-step loss is the mean-squared error at the next timestep summed over fields.
  - The rollout loss is the mean-squared error after applying the neural PDE surrogate 5 times, summing over fields and time dimension

Experiment

■ Navier-Stokes in 2D

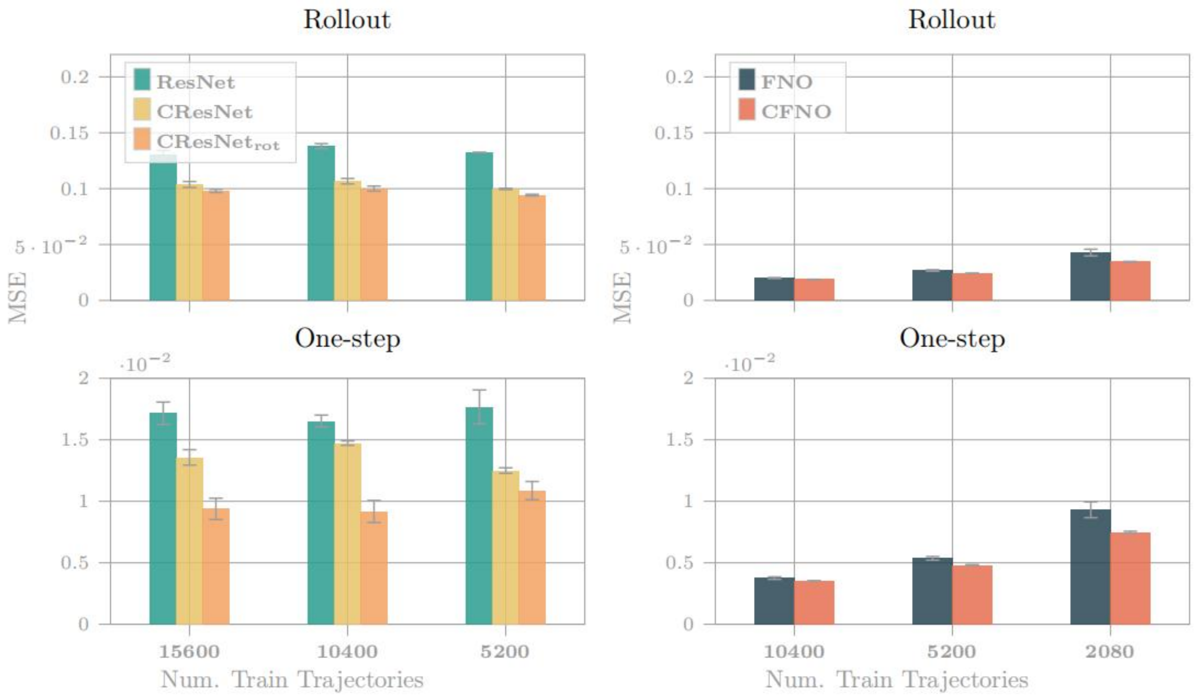


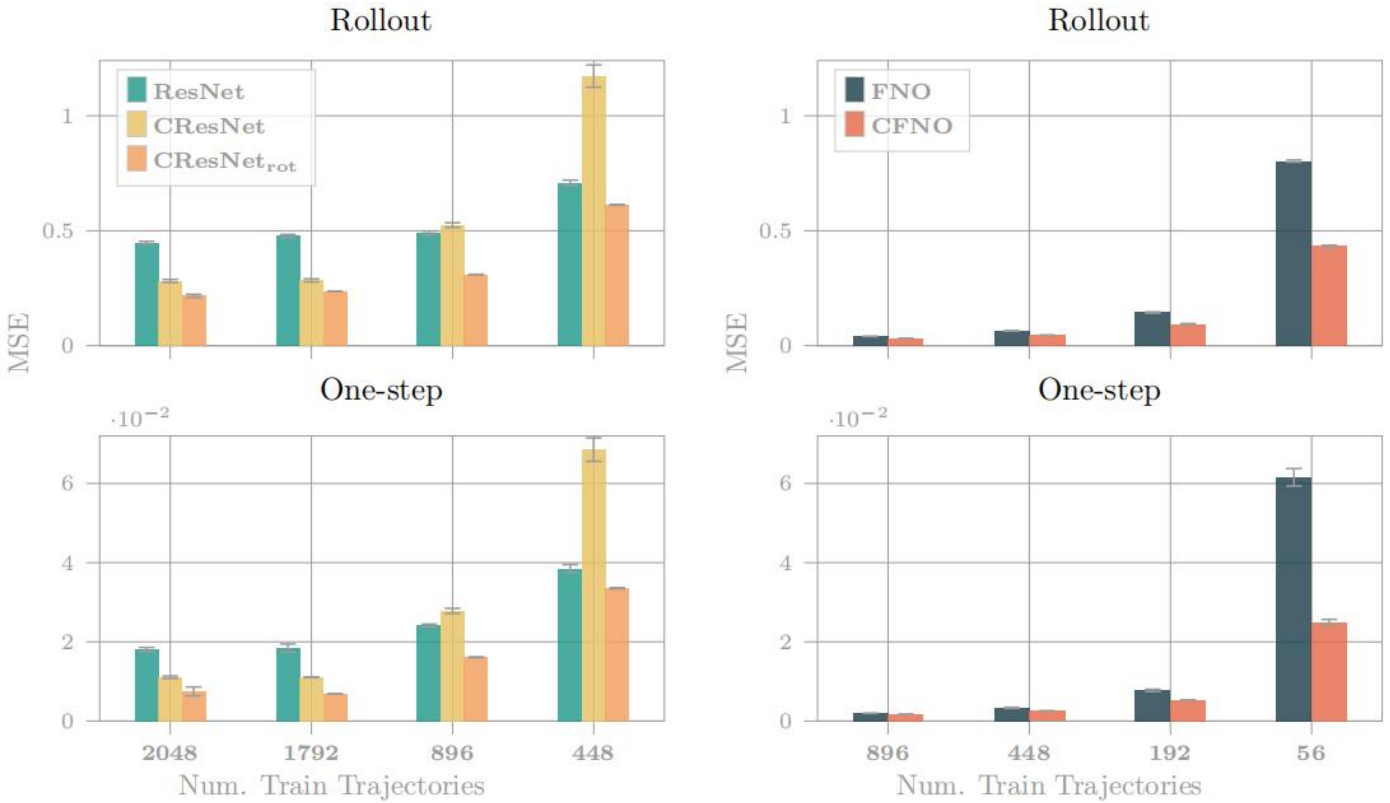
Figure 7: Results on ResNet based (left) and Fourier based (right) architectures on the Navier-Stokes experiments. One-step and rollout losses are reported.

➢ The incompressible Navier-Stokes equations are built upon momentum and mass conservation of fluids expressed for the velocity flow field v.

$$\frac{\partial v}{\partial t} = -v \cdot \nabla v + \mu \nabla^2 v - \nabla p + f \ , \quad \nabla \cdot v = 0 \ ,$$

Experiment

■ Shallow water equations



- . The shallow water equations describe a thin layer of fluid of constant density in hydrostatic balance, bounded from below by the bottom topography and from above by a free surface.

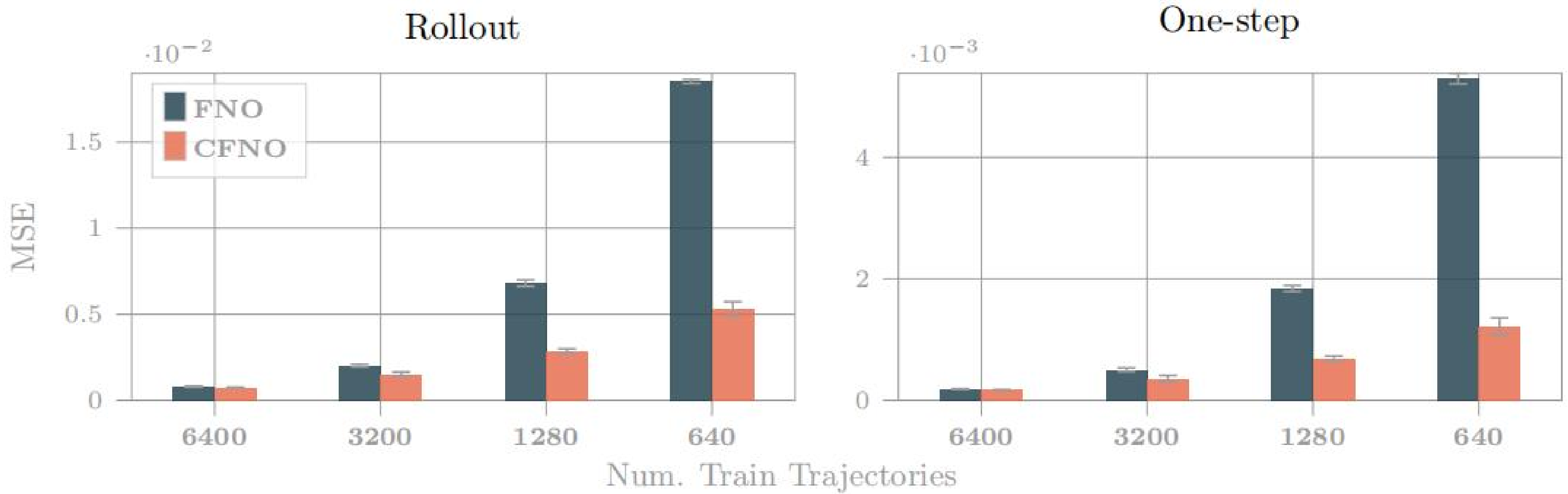
$$\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} + g \frac{\partial \eta}{\partial x} = 0 \,,$$

$$\frac{\partial v_y}{\partial t} + v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} + g \frac{\partial \eta}{\partial y} = 0 \,,$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\Big[(\eta + h) v_x\Big] + \frac{\partial}{\partial y}\Big[(\eta + h) v_y\Big] = 0 \,.$$

Experiment

■ **Maxwell's equations in matter in 3D**



高瓴人工智能学院
Gaoling School of Artificial Intelligence

Conclusion

- This work introduced Clifford neural layers, both for convolution and Fourier operations, that handle the various scalar (e.g. charge density), vector (e.g. electric field), bivector (magnetic field) and higher order fields as proper geometric objects organized as multivectors.
- The multivector viewpoint with Clifford neural layers led to better representation of the relationship between different fields and their individual components, allowing us to significantly outperform equivalent standard architectures for neural PDE surrogates.
- Extensions towards Clifford graph networks and Clifford attention based models, where the geometric product replaces the dot product will be useful to explore.

中国人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

Thank You for listening!

Angxiao Yue

2023-5-31