# STAResNet: a Network in Spacetime Algebra to solve Maxwell's PDEs *

**Alberto Pepe, Joan Lasenby**
Signal Processing and Communications Lab
University of Cambridge
Cambridge, UK
{ap2219, jl221}@cam.ac.uk

**Sven Buchholz**
Department of Computer Science and Media
Technische Hochschule Brandenburg
University of Applied Sciences
Brandenburg, Germany
sven.buchholz@th-brandenburg.de

**Abstract**

We introduce STAResNet, a ResNet architecture in Spacetime Algebra (STA) to solve Maxwell's partial differential equations (PDEs). Recently, networks in Geometric Algebra (GA) have been demonstrated to be an asset for truly geometric machine learning. In [1], GA networks have been employed for the first time to solve partial differential equations (PDEs), demonstrating an increased accuracy over real-valued networks. In this work we solve Maxwell's PDEs both in GA and STA employing the same ResNet architecture and dataset, to discuss the impact that the choice of the right algebra has on the accuracy of GA networks. Our study on STAResNet shows how the correct geometric embedding in Clifford Networks allows for a mean square error (MSE) between ground truth (GT) and estimated fields up to 2.6 times lower as opposed to Clifford ResNet with 6 times fewer trainable parameters, consistently lower MSE and higher correlation regardless of the sampling period of the dataset, the presence of obstacles with either seen or unseen configurations, the number of channels in the ResNet architecture, the number of rollout steps and whether the field is in 2D or in 3D space. This demonstrates how choosing the right algebra in Clifford networks is a crucial factor for more compact, accurate, descriptive and better generalising pipelines.

***K*eywords** geometric machine learning · spacetime algebra · Clifford algebra · partial differential equations

## 1 Introduction

Geometric Algebra Networks, also known as Clifford Algebra Networks, leverage the mathematical framework of Geometric Algebra to represent and manipulate data. Geometric Algebra is a powerful, high-dimensional algebraic system that extends the traditional linear algebra, enabling the compact and intuitive representation of geometric transformations, rotations, and reflections [2, 3, 4]. In GA networks, data and operations are expressed in terms of multivectors, which can capture complex geometric relationships more naturally than traditional tensor or matrix representations.

Early proposals for neural networks working in Geometric Algebra (GA) can be found in the literature from the end of the last century [5, 6, 7]. However, it is only in the past few years that the need for an effective and intuitive approach to geometrical problems in learning has sparked renewed interest in the field. Today, several architectures in GA exist, capable of handling convolutions and Fourier transforms [1, 8], performing rotations and rigid body motions [9, 10], and preserving end-to-end equivariance [11, 12, 13].

In recent years, the use of machine learning, particularly neural networks, to solve partial differential equations (PDEs) has gained significant traction [14, 15, 16, 17]. State-of-the-art approaches include physics-informed neural networks (PINNs) [18, 19, 20], Fourier neural operators [21, 22], and deep Ritz methods [23, 24]. These methods aim to approximate solutions to PDEs by embedding the physical laws described by the PDEs directly into the training process

* *Citation*: **Authors. Title. Pages.... DOI:000000/11111.**

of neural networks. For example, PINNs incorporate the residuals of the PDEs into the loss function, ensuring that the neural network's output satisfies the underlying physical laws.

When solving PDEs, GA networks offer several unique advantages. One of the primary benefits is their ability to preserve the geometric meaning of data. By embedding data in the correct algebraic framework, GA networks can handle geometric transformations and physical laws more naturally. This is particularly advantageous for problems involving rotations, reflections, and other geometric transformations, which are common in physics and engineering. Moreover, GA networks can represent complex multi-dimensional relationships compactly and efficiently, making them well-suited for high-dimensional PDEs. The ability to handle convolutions and Fourier transforms within the GA framework also enables GA networks to process and analyze spatial and frequency domain information effectively, which is crucial for many PDE problems.

The question we address here is: how does the choice of the algebra in which data is embedded affect the accuracy of the PDE solution via a GA network? We focus on Maxwell's PDEs, which describe the fundamental behavior of electric and magnetic fields, and compare their solutions using two approaches: one in $n$D GA and one in $(n+1)$D Spacetime Algebra (STA).

By comparing these two approaches, we aim to understand the impact of the algebraic framework on the accuracy and efficiency of the solutions. The choice of algebra can significantly influence the complexity of the problem formulation and the performance of the neural network, potentially leading to more accurate and efficient PDE solvers.

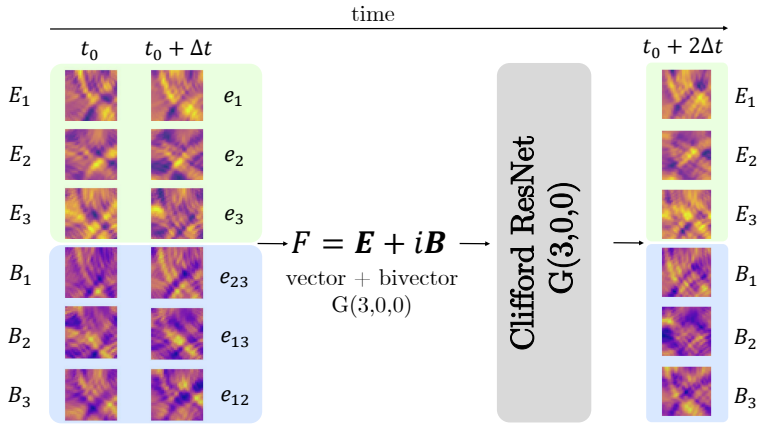



Figure 1: 3D GA approach: solving Maxwell's PDEs through Clifford ResNet, a ResNet-inspired network working in G(3,0,0) in the 3D case and G(2,0,0) in the 2D case.

## 2 Problem Definition

Maxwell's equations describe the behaviour of electric and magnetic fields in classical electromagnetism. They read as follows:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} \tag{1}$$

known as **Gauss's Law for Electricity** and states that the divergence of the electric field ($\mathbf{E}$) is equal to the charge density ($\rho$) divided by the permittivity of free space ($\varepsilon_0$).

$$\nabla \cdot \mathbf{B} = 0 \tag{2}$$

representing **Gauss's Law for Magnetism** and indicating that the divergence of the magnetic field ($\mathbf{B}$) is zero, implying the absence of magnetic monopoles.

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0 \tag{3}$$

i.e. **Faraday's Law of Induction**, stating that the curl of the electric field ($\mathbf{E}$) is equal to the negative rate of change of the magnetic field ($\mathbf{B}$) with respect to time.

$$\nabla \times \mathbf{B} - \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} = \mu_0 \mathbf{J} \tag{4}$$

which is the **Ampère's Law**, stating that the curl of the magnetic field ($\mathbf{B}$) is equal to the sum of the current density ($\mathbf{J}$) and the product of the permeability of free space ($\mu_0$) and the rate of change of the electric field ($\mathbf{E}$) with respect to time.

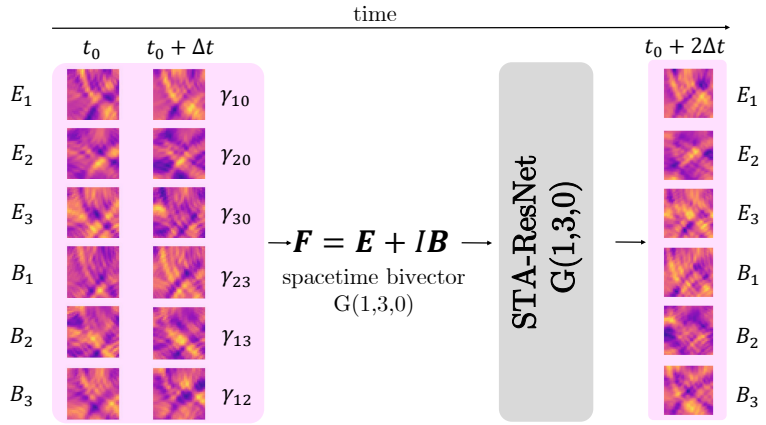



Figure 2: STA approach: solving Maxwell's PDEs through STAResNet, our ResNet-inspired network working in G(1,3,0) in the 3D case and G(1,2,0) in the 2D case.

### 2.1 Maxwell's equations in G(3,0,0)

This scenario will be our baseline, and it is equivalent to that proposed in [1]. G(3,0,0) is the three-dimensional Geometric Algebra spanned by the three basis vectors $\{e_1, e_2, e_3\}$. By defining the pseudoscalar $i = e_{123}$ we can introduce the electromagnetic (EM) field multivector $F$:

$$F = \mathbf{E} + i\mathbf{B} = E_1 e_1 + E_2 e_2 + E_3 e_3 + B_1 e_{23} + B_2 e_{13} + B_3 e_{12}. \tag{5}$$

With this object we can pair Eq. 1-4 into a set of two equations only:

$$\nabla \cdot F = \frac{\rho}{\varepsilon_0} \tag{6}$$

$$\nabla \times F + \frac{\partial F}{\partial (it)} = i\mu_0 \mathbf{J}, \tag{7}$$

which, using the geometric product, can be further reduced to:

$$\frac{1}{c}\left(\frac{\partial}{\partial t} + \nabla\right) F = \frac{1}{\epsilon_0}\left(\rho - \frac{\mathbf{J}}{c}\right) \tag{8}$$

In which $c$ is the speed of light. In this scenario, we formulate the PDE solution as a 3D multivector-to-multivector regression problem. The inputs to the network will be a pair of multivectors sampled at two consecutive time instants $\{F_i, F_{i+\Delta t}\}$, while the target will be the multivector after a history of 2 time steps, i.e. $F_{i+2\Delta t}$. This is summarized in Fig. 1.

### 2.2 Maxwell's equations in G(1,3,0)

STA i.e. G(1,3,0), is spanned by four basis vectors $\{\gamma_\mu\}$, with $\gamma_0^2 = -\gamma_k^2 = 1$ and $k = 1, 2, 3$. In STA, the electric and magnetic fields are both bivectors of the form:

$$\mathbf{E} = E_1\sigma_1 + E_2\sigma_2 + E_3\sigma_3 \tag{9}$$

$$\mathbf{B} = B_1\sigma_1 + B_2\sigma_2 + B_3\sigma_3 \tag{10}$$

with $\sigma_k = \gamma_k\gamma_0$. Also in STA, like in G(3,0,0), Maxwell's PDEs can be combined into a single equation:

$$\nabla(\mathbf{E} + I\mathbf{B}) + \frac{\partial}{\partial t}(\mathbf{E} + I\mathbf{B}) = \rho - \mathbf{J}, \tag{11}$$

which, by further defining the Faraday bivector $F = \mathbf{E} + I\mathbf{B}$ and the current $J = (\rho - \mathbf{J})\gamma_0$, reduces to just:

$$\nabla F = J. \tag{12}$$

Eq. 12 is not only an extremely compact rendition of Maxwell's equations, especially if compared to Eq.1-4, but it has also the advantage of being expressed in term of a geometric product, which is invertible. This means that working in STA offers an easier way of computing the propagation of the EM field in a conducting medium [25, 26]. We wish to show that a neural network working in the same mathematical space performs better as opposed to working in G(3,0,0), by keeping data and architectures identical.

In this scenario, we formulate the PDEs solution as a spacetime bivector-to-bivector regression problem. The spacetime bivector has form:

$$\mathbf{F} = \mathbf{E} + I\mathbf{B} = E_1\gamma_{10} + E_2\gamma_{20} + E_3\gamma_{30} + B_1\gamma_{13} + B_2\gamma_{13} + B_3\gamma_{12}. \tag{13}$$

The inputs to the network will be a pair of Faraday bivectors sampled at two consecutive time instants $\{\mathbf{F}_i, \mathbf{F}_{i+\Delta t}\}$, while thevtarget will be the bivector after a history of 2 time steps, i.e. $\mathbf{F}_{i+2\Delta t}$. The STA approach via STAResNet is summarized in Fig. 2. Note how the difference with respect to the approach in 2.2 is exclusively the mathematical framework the network works in.

## 3 Approach

### 3.1 Architecture

We study 2D and 3D Maxwell's PDEs. In the 2D case, we compared the 2D Clifford ResNet and the 2D STAResNet: 2D Clifford ResNet, that handles multivectors $F$ in G(2,0,0), has 20 blocks of 2D convolutions paired with a ReLU activation function. Each convolutional layer has $C = 32$ channels, excepts the first one with 2 channels and the last one with 1 channel, corresponding to the number of input and output time steps, respectively.

2D STAResNet, that handles bivectors $\mathbf{F}$ in G(1,2,0), shares the same structure with the 2D Clifford ResNet, except for the embedding in a different mathematical space and the number of channels, reduced to $C = 24$. This has been done to make sure both networks have the same number of parameters for a fair comparison. Both networks have slightly below 1M parameters.

In the 3D case, the 3D Clifford ResNet sits in G(3,0,0) while the 3D STAResNet sits in G(1,3,0). The architectures are identical to their 2D counterparts, with the only exceptions that 2D convolutions are replaced with 3D convolutions and the number of channels are reduced from 32 to 11 in Clifford ResNet and from 24 to 8 in STAResNet, for a total of approximately 600,000 parameters for both networks.

### 3.2 Training details

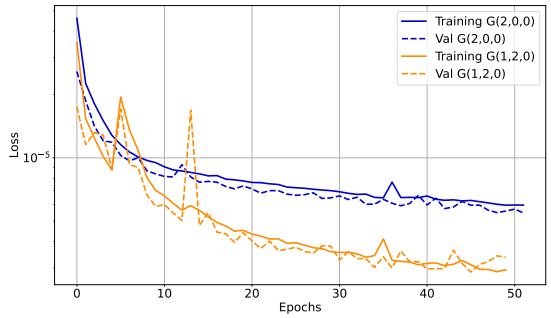
Both networks have been trained for 50 epochs with a batch size of 32 in 2D and of 2 in the 3D case. We chose the Adam optimiser with learning of $10^{-3}$. The objective function be minimised is the mean squared error (MSE) between ground truth fields $\mathbf{E}_{i+2\Delta t}, \mathbf{B}_{i+2\Delta t}$ and estimated fields $\hat{\mathbf{E}}_{i+2\Delta t}, \hat{\mathbf{B}}_{i+2\Delta t}$ as our objective function, which in 2D is defined as

$$\mathcal{L}_{2D} = \frac{1}{MN}\sum_{m=0}^{M}\sum_{n=0}^{N}(E_{xmn,i+2\Delta t} - \hat{E}_{xmn,i+2\Delta t})^2 + (E_{ymn,i+2\Delta t} - \hat{E}_{ymn,i+2\Delta t})^2 + (B_{zmn,i+2\Delta t} - \hat{B}_{zmn,i+2\Delta t})^2, \tag{14}$$

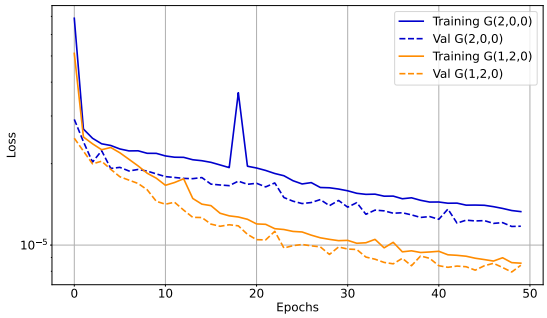
in which $m, n$ indicate the spatial location of the field within the 2D surface. In the 3D case, we have that

$$\mathcal{L}_{3D} = \frac{1}{LMN}\sum_{l=0}^{L}\sum_{m=0}^{M}\sum_{n=0}^{N}(E_{jlmn,i+2\Delta t} - \hat{E}_{jlmn,i+2\Delta t})^2 + (B_{jlmn,i+2\Delta t} - \hat{B}_{jlmn,i+2\Delta t})^2, \tag{15}$$

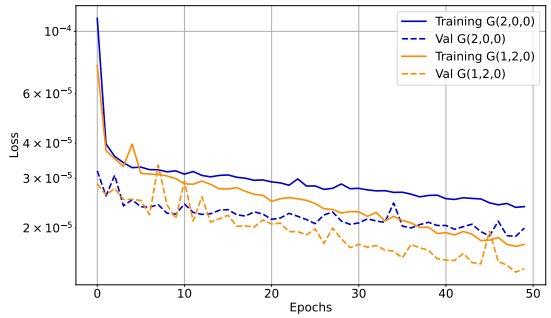
in which $l, m, n$ indicate the spatial location of the field within the 3D volume and $j = \{x, y, z\}$. We compare results obtained with the weights of the two pipelines yielding minimum MSE error at validation stage. The code has been implemented via tensorflow 2.13.1 and accelerated with CUDA 12.2 on an NVIDIA GeForce RTX 4090 GPU. Code can be found here.



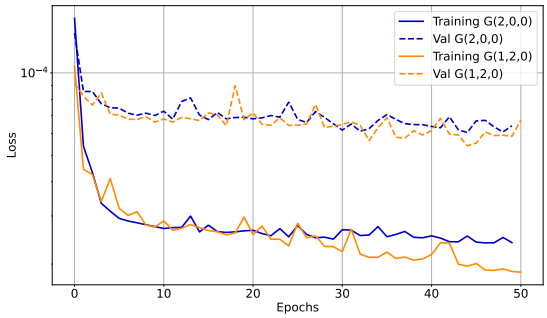

(a) $\Delta t = 25$s




(b) $\Delta t = 50$s

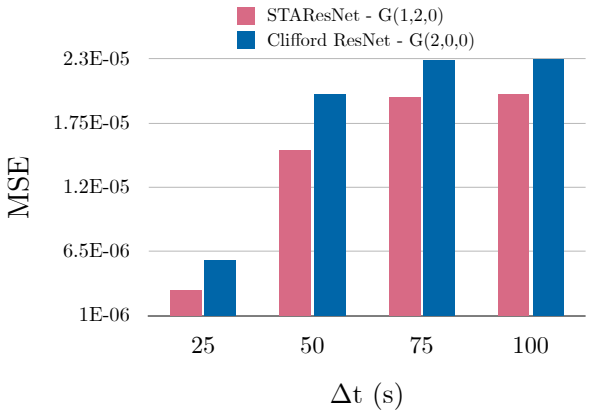



(c) $\Delta t = 75$s

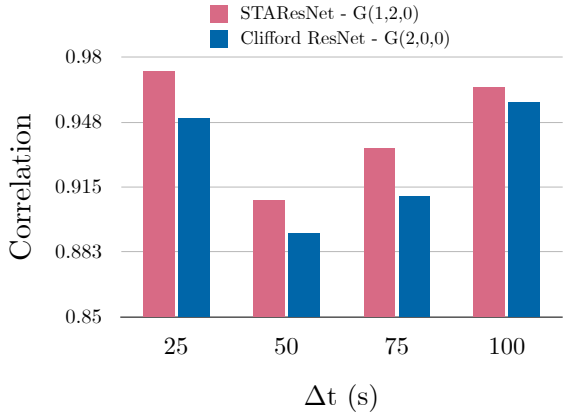



(d) $\Delta t = 100$s

Figure 3: Training and validation losses versus number of epochs for 2D Maxwell's PDEs.




(a) MSE (↓) versus $\Delta t$.




(b) Correlation (↑) versus $\Delta t$.

Figure 4: Test metrics over test set for varying $\Delta t$.

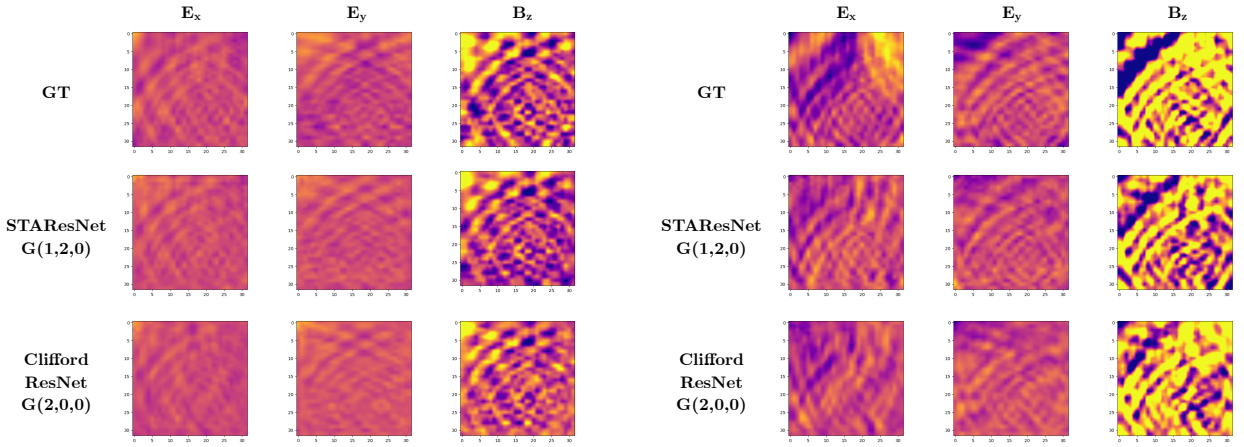
# 4 Experiments in 2D

## 4.1 Impact of sampling period

The first scenario we look at is the solution of Maxwell PDEs for varying sampling time $\Delta t$ of the EM field. The datasets have been generated with a Finite-Difference Time-Domain (FDTD) solver following the specifications of [1]. We consider a surface with spatial resolution of $32 \times 32$ and step size $\Delta x = \Delta y = 5 \cdot 10^{-7}$m, with EM field sampled with varying sampling period $\Delta t = \{25, 50, 75, 100\}$s. The light is propagated from 6 point sources randomly placed in the $xy$ plane. The wavelength of the emitted light is $\lambda = 10^{-5}m$. Each light source emits light with a random phase and random amplitude.

The training set includes 30000 frames, divided into 32 sequences of 12 samples of the EM field. For each sequence we employ the first two samples, i.e. the first two time steps, as input to the networks and the third as target. The validation and test sets are structured similary, and they include 3000 and 2400 frames, respectively.

The training and validation losses profiles for the four $\Delta t$ cases are shown in Fig. 3. STAResNet consistently achieves lower loss both at training and validation stage compared to its 2D GA counterpart.




(a) $\Delta t = 25$s. (b) $\Delta t = 50$s.

Figure 5: Estimated EM field for varying $\Delta t$.

We measure two quantities at testing stage, the MSE as defined in Eq. 14 and the correlation index, defined in 2D as

$$r_{2D} = \frac{1}{MN} \sum_{m=0}^{M} \sum_{n=0}^{N} T_{mn} \hat{T}_{mn}, \tag{16}$$

where $T = [E_{x,i+2\Delta t}, E_{y,i+2\Delta t}, B_{z,i+2\Delta t}]$. Results are summarised in Fig. 4. As expected, the MSE increases as more time passes between successive frames. Correlation increases as for large $\Delta t$ the fields tend to dissipate, hence presenting less busy patterns that result in higher correlation despite a worse estimation of the fields. STAResNet consistently performs better regardless of the sampling period $\Delta t$.

Examples of the estimated electric and magnetic fields for varying $\Delta t$ are given in Fig. 18a. In all the cases presented, STAResNet does a better job at estimating the interference patterns arising from multiple sources as opposed to the Clifford ResNet. Additionally, STA has the advantage of unifying the fields component into a single object, the Faraday bivector, which it is built from the GT and estimated fields, respectively, and we visualise its square magnitude defined as

$$\mathbf{F}^2 = (E_x \gamma_{10} + E_y \gamma_{20} + B_z \gamma_{12})^2 \tag{17}$$

Examples for each tested $\Delta t$ are shown in Fig. 6 As expected, as $\Delta t$ increases the PDE solutions become less accurate, but in each of the four cases STAResNet yields magnitude plots with more similar patterns to GT as opposed to Clifford ResNet working in 2D GA.

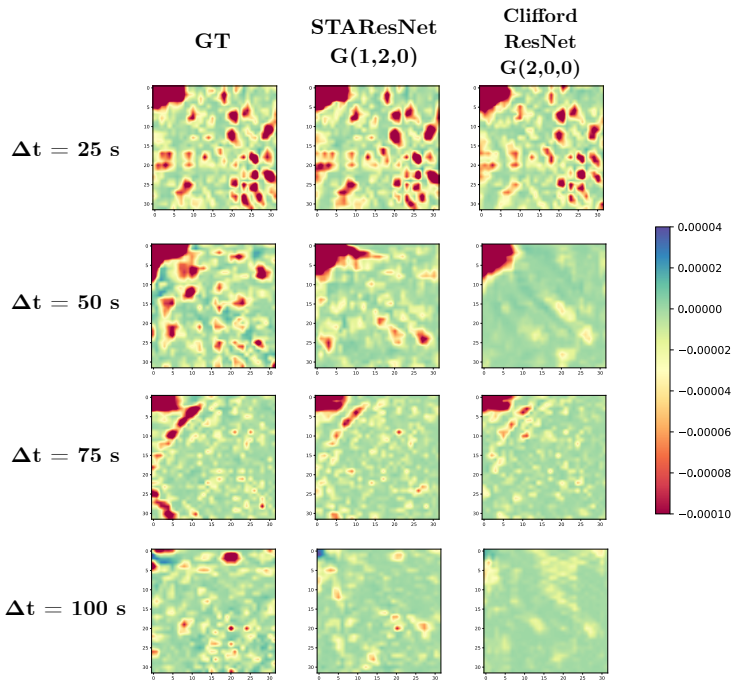



Figure 6: Magnitude of the Faraday bivector for varying $\Delta t$.

### 4.2 Impact of obstacles

We make the PDE solution more challenging by increasing the surface size to $48 \times 48$, and adding a single obstacle of varying dimension and fixed permittivity $\epsilon = 1.7^2$. The step size has been kept to $\Delta x = \Delta y = 5 \cdot 10^{-7}$m and the sampling period to $\Delta t = 25$s. Light sources follow the same specifications of 4.1.

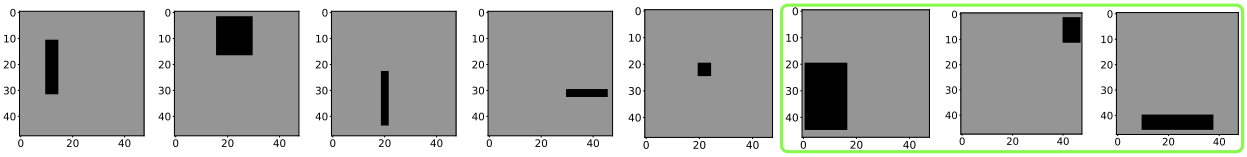


Figure 7: The 5 different obstacle configurations. The 3 unseen geometries are highlighted.

The training set includes 30000 frames, divided into 32 sequences of 12 samples of the EM field, with five different obstacle configurations appearing with the same probability (i.e. 6000 frames per configuration). The validation includes 12800 sequences with the same five obstacle configurations (i.e. 2560 frames per configuration) but with different light sources with respect to the training set. We employed two different test sets, with 12800 sequences each. The first test set presents obstacles in the same position of the training set but with different light sources. The second test is generated with three obstacles in different locations to those in the training set, to assess the network's ability to generalise on previously unseen geometries (see Fig. 7). To make the comparison even more challenging, we reduced the number of channels of STAResNet from 24 to 23, for a total of 866326 trainable parameters agains 928580 parameters of Clifford ResNet.

Training and validation losses for Clifford ResNet and STAResNet are shown in Fig. 8. The trend is the same to that shown in Fig. 3: also in presence of obstacles with fixed sampling period, STAResNet attains a lower error. Moreover, STAResNet achieves lower MSE and higher correlation coefficient as compared to 2D Clifford ResNet regardless if testing on seen or unseen obstacle configuration (see Fig. 9). This is clearly visible in Fig. 10: estimated fields with STAResNet preserve more structure than those estimated with Clifford ResNet.

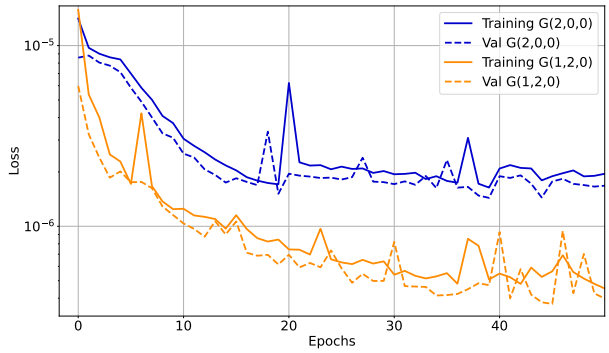
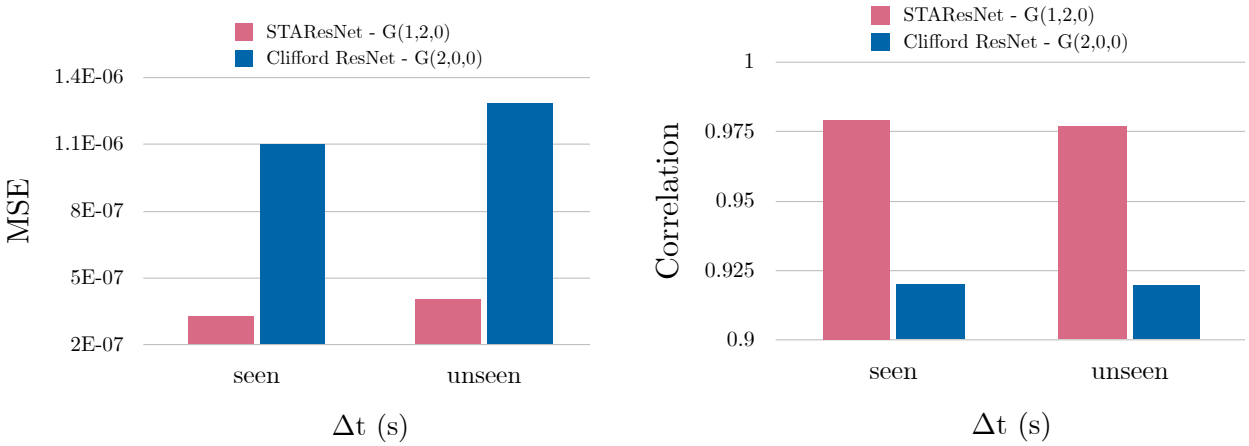


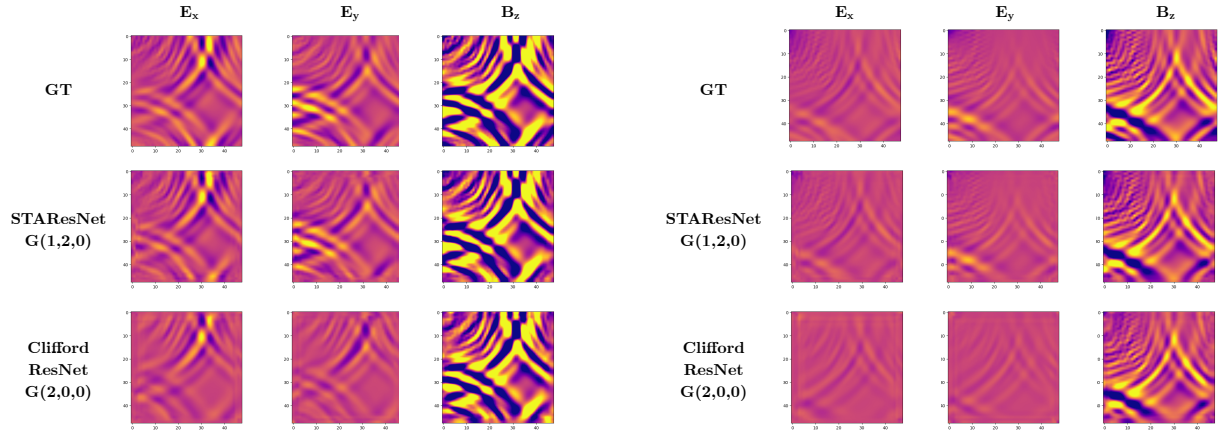

Figure 8: Training and validation losses versus number of epochs for the PDE in presence of obstacles.




(a) MSE (↓).

(b) Correlation (↑).

Figure 9: Test metrics over the two test sets with seen and unseen obstacle configurations.




(a) Estimated EM field for configuration 2.

(b) Estimated EM field for configuration 5.

Figure 10: Estimated EM fields with obstacles.

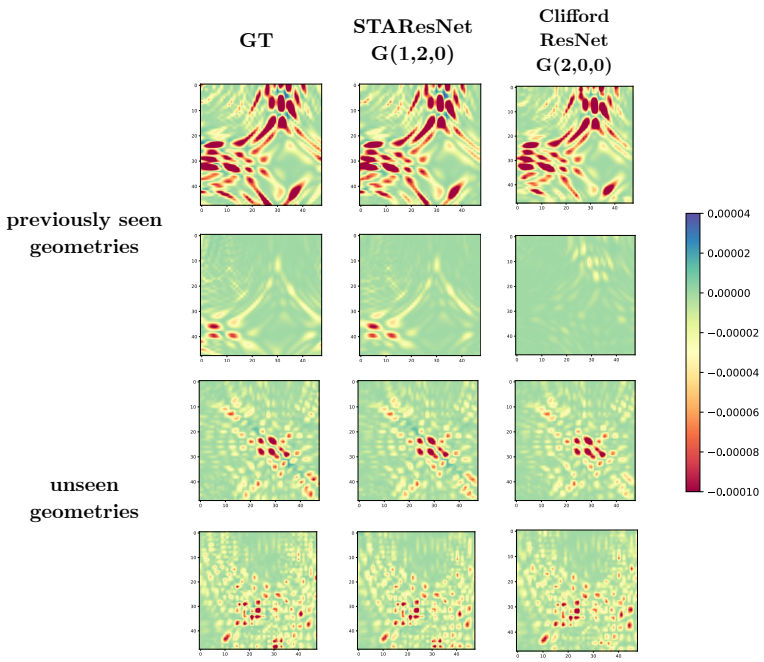



Figure 11: Magnitude of the Faraday bivector in presence of obstacles.

### 4.3 Impact of number of parameters

STAResNet works in a mathematical space with one additional dimension compared to Clifford ResNet in 2D GA. This means that the tensor embedded in G(1,2,0) in STAResNet will have its last dimension of size $2^3 = 8$, as opposed to $2^2 = 4$ in 2D Clifford ResNet. This impacts the number of trainable parameters of the two pipelines, meaning that for the same number of channels or for the same size of the convolutional filters, STAResNet will generally have a larger number of trainable parameters.

To verify that STAResNet is consistently superior to a 2D GA approach and provide a fair comparison, we train both networks with a varying number of hidden channels, ranging between 15 and 40. This yields pipelines with different number of trainable parameters, from $2 \times 10^5$ up to $2 \times 10^6$. Results are summarized in Fig. 12.

The first thing it can be noticed is that the error decreases for a larger number of parameters in both networks, as expected. The test MSE error with 2D Clifford ResNet, however, never reaches values below $1 \times 10^{-6}$. Even with 40 channels and above 2 million trainable parameters, the 2D Clifford ResNet cannot catch up with STAResNet, that with just 15 channels and 370,000 parameters yields a test error of below $5.5 \times 10^{-7}$. This means that our 3D STAResNet, with $16.4\%$ the number of parameters of 2D Clifford ResNet, can estimate EM fields twice as accurately. This proves that the improvement offered by STAResNet comes from the mathematical space it works in rather than the fact that it is generally a larger network.

The second thing that can be seen in Fig. 12 is the robustness to previously unseen data. The gap in the error between seen and unseen obstable configurations with STAResNet is visibly smaller than the gap with Clifford ResNet, which is less capable of generalisation when presented with test data which were not included in the training set. This is true regardless of the number of parameters of the networks.

### 4.4 Impact of rollout

Lastly, we quantify the rollout error, i.e. we quantify the discrepancy between the predicted and actual EM fields over multiple time steps by employing the estimated fields at $i + 2\Delta t$ at step $m = 1$ in output of the two architectures as new inputs in a sliding window fashion to estimate the next snapshot of the field for a total of $m = 10$ rollout steps. Since each trajectory contains 12 samples, we measure the rollout error for $m$ ranging between 1 and 10. Results are summarized in Fig. 13, in which we measure MSE and structural similarity index (SSIM) between GT and predicted fields. Since the error rapidly propagates between successive time steps, we train Clifford ResNet and STAResNet on datasets captured for small $\Delta t$, i.e. $\Delta t = \{5, 8, 10, 15\}s$.

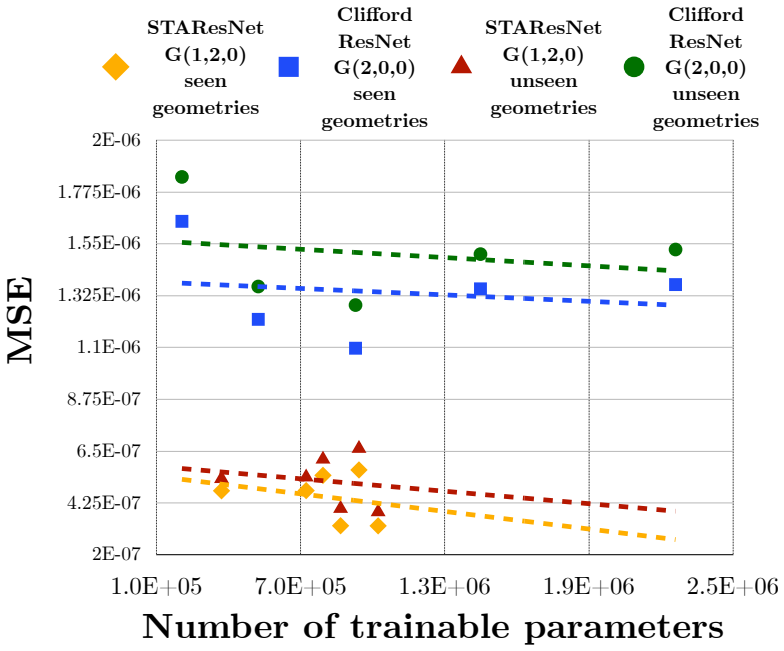
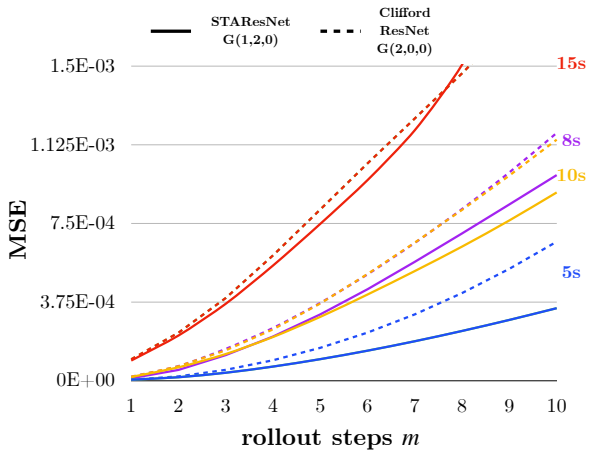
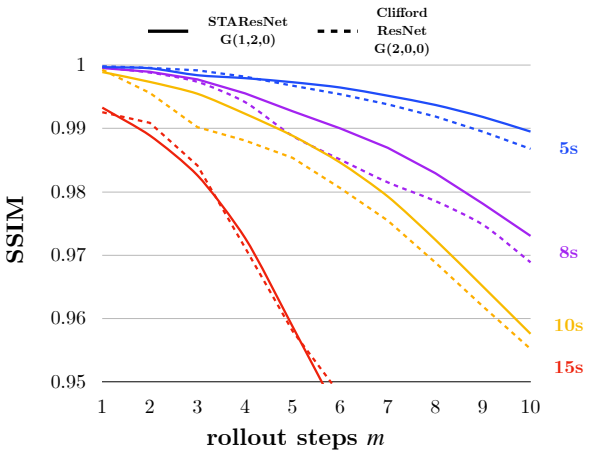



Figure 12: Test error over the estimated EM fields in presence of seen and unseen obstacle geometries versus number of trainable parameters.

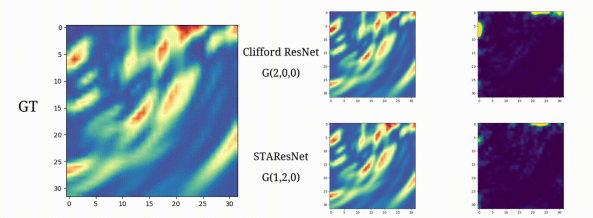



(a) MSE (↓) versus rollout steps.

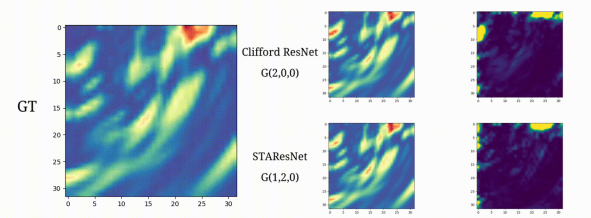



(b) SSIM (↑) versus rollout steps.

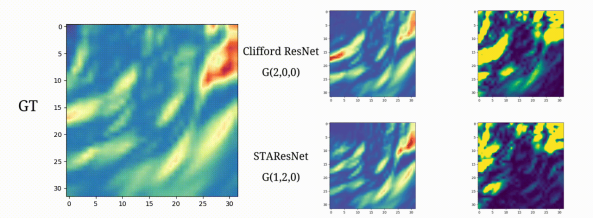
Figure 13: Test metrics over test set versus rollout steps $m$ for 2D EM fields.



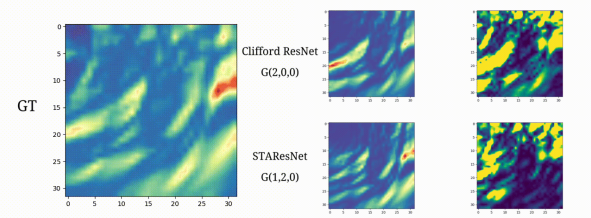

(a) $m = 1$




(b) $m = 2$




(c) $m = 8$

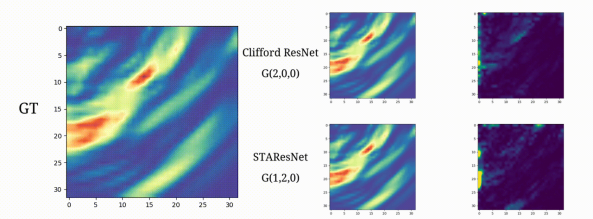



(d) $m = 10$

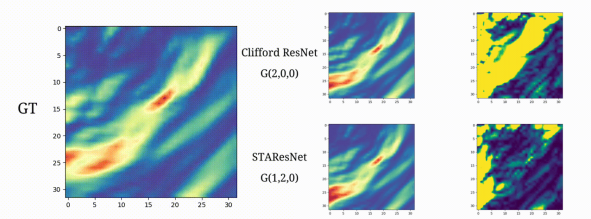
Figure 14: 2D GT $\mathbf{F}$, estimated $\hat{\mathbf{F}}$ and difference $|\mathbf{F} - \hat{\mathbf{F}}|$ for Clifford ResNet and STAResNet at different rollout steps. $\Delta t = 5$s.

From Fig. 13 it is possible to notice that STAResNet approach yields consistently lower MSE and higher SSIMover Clifford ResNet, and that the gap between the two becomes more visible as the number of rollout steps increases. It can be argued that, as $\Delta t$ increases, the gap between the two models tends to narrow. This happens because for large $\Delta t$ there is little continuity between successive time steps, meaning that the error is already significant for small $m$, and for large $m$ both estimates with STAResNet and Clifford ResNet deviate so much with respect to GT to the comparison meaningless.

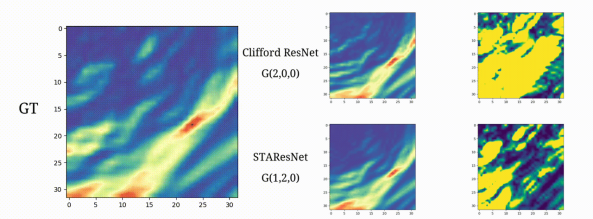
Examples of the evolution of $\mathbf{F}$ as a function of time are given in Figs. 15-14 for two different sequences. In $viridis$ colormap is reported the absolute difference between $\mathbf{F}$ and $\hat{\mathbf{F}}$ clipped between $[0, 0.02]$. It can be seen how errors in the estimates obtained via STAResNet are visibly more localised and their magnitude more bounded as opposed to those obtained via Clifford ResNet.



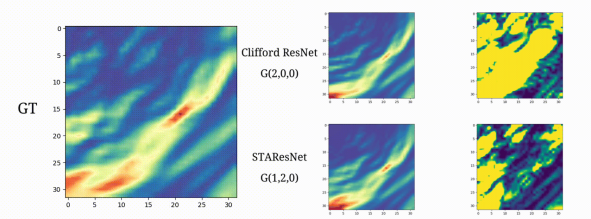

(a) $m = 1$




(b) $m = 5$




(c) $m = 8$




(d) $m = 10$

Figure 15: 2D GT $\mathbf{F}$, estimated $\hat{\mathbf{F}}$ and difference $|\mathbf{F} - \hat{\mathbf{F}}|$ for Clifford ResNet and STAResNet at different rollout steps. $\Delta t = 8$s.

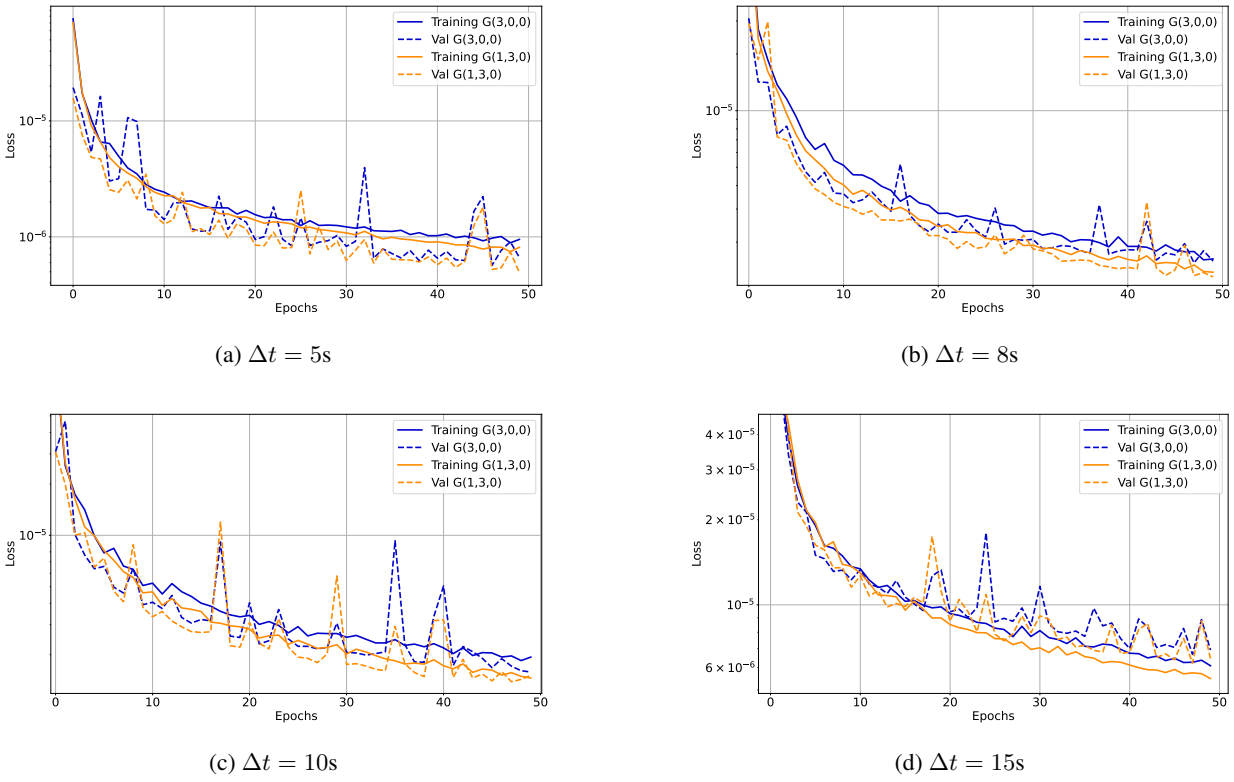
## 5 Experiments in 3D

In the 3D case we consider a volume with spatial resolution of $28 \times 28 \times 28$ and step size $\Delta x = \Delta y = \Delta z = 5 \cdot 10^{-7}$m, with EM field sampled with varying sampling period $\Delta t = \{5, 8, 10, 15\}$s. The light is propagated from 6 point sources randomly placed in each of the the $xy$, $yz$ and $xz$ planes, for a total of 18 planar sources. The wavelength of the emitted light is $\lambda = 10^{-5}m$. Each light source emits light with a random phase and random amplitude. Training, validation and test sets are structured similarly to their 2D counterparts.

We measure the MSE error between estimated and GT fields as described in Eq. 15, and visualise the magnitude of the 4D spacetime bivector:

$$\mathbf{F}^2 = (E_1\gamma_{10} + E_2\gamma_{20} + E_3\gamma_{30} + B_1\gamma_{13} + B_2\gamma_{13} + B_3\gamma_{12})^2 \tag{18}$$

Training and validation losses for the two architectures at different sampling periods are shown in Fig. 16. Our STA approach achieves lower validation loss in all four cases, similarly to Fig. 3. We believe that the smaller gap between the loss profiles of STAResNet and CliffordResNet is to be attributed to the small $\Delta t$ and smaller domain rather than the fact that we are working in 3D over 2D.




(a) $\Delta t = 5$s

(b) $\Delta t = 8$s

(c) $\Delta t = 10$s

(d) $\Delta t = 15$s

Figure 16: Training and validation losses versus number of epochs for 3D Maxwell's PDEs.

A plot of the MSE as a function of the number of rollout steps $m$ for successive time instants is given in Fig. 17. For each of the four sampling period and regardless of the number of rollout steps, STAResNet consistently outperforms Clifford ResNet. Notice how this holds true also for $\Delta t = 15$s, while in the 2D case, in Fig. 13, errors explode for $\Delta t \geq 15$s.

Examples of the evolution of the magnitude of the 3D Faraday bivector for successive rollout steps are shown in Fig. 18. The absolute difference between GT and estimated 3D Faraday bivectors' magnitudes clipped between $[0, 0.02]$ is shown in $viridis$ color map, meaning they share the same range. The higher the intensity and the opacity of the plot, the higher the error between ground truth. It can be noticed how, even after only two time steps, STAResNet yields output closer to GT as opposed to the 3D Clifford ResNet.

Similarly, slices along the $z$-dimension of the 3D Faraday bivector for successive rollout steps are shown in Fig. 19-20. Slices are taken at the bottom ($z = 0$), middle ($z = 13$) and top ($z = 27$) of the volume. The clipped, absolute

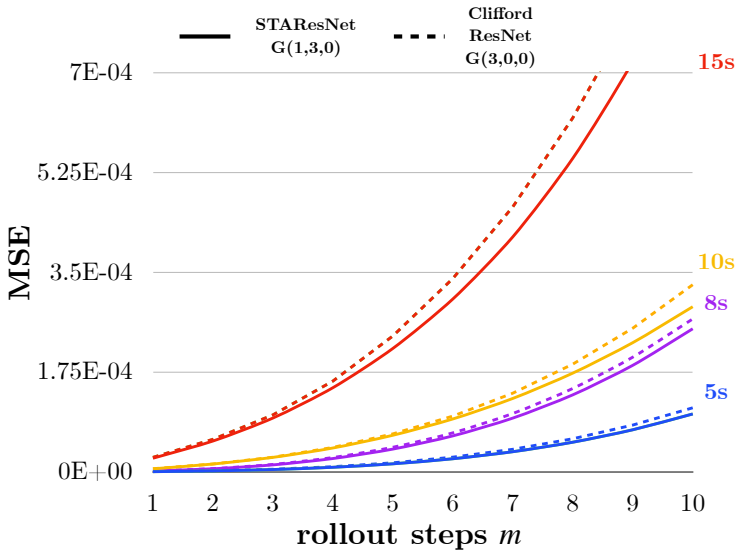
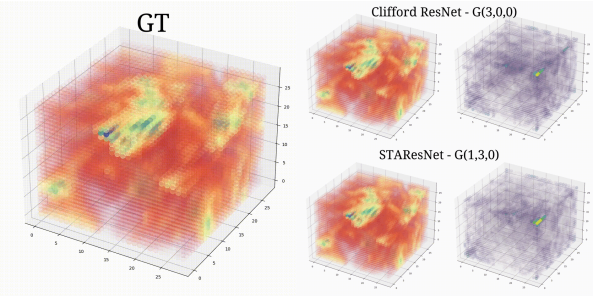
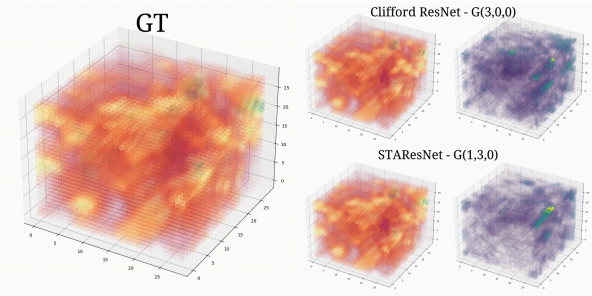


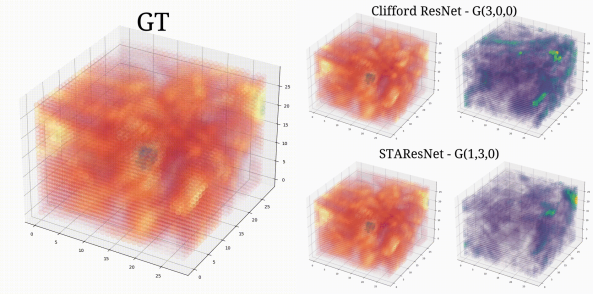

Figure 17: Test metrics over test set versus rollout steps $m$ for 3D EM fields.



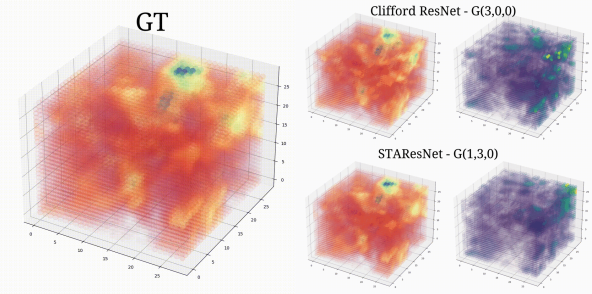

(a) $m = 2$




(b) $m = 5$



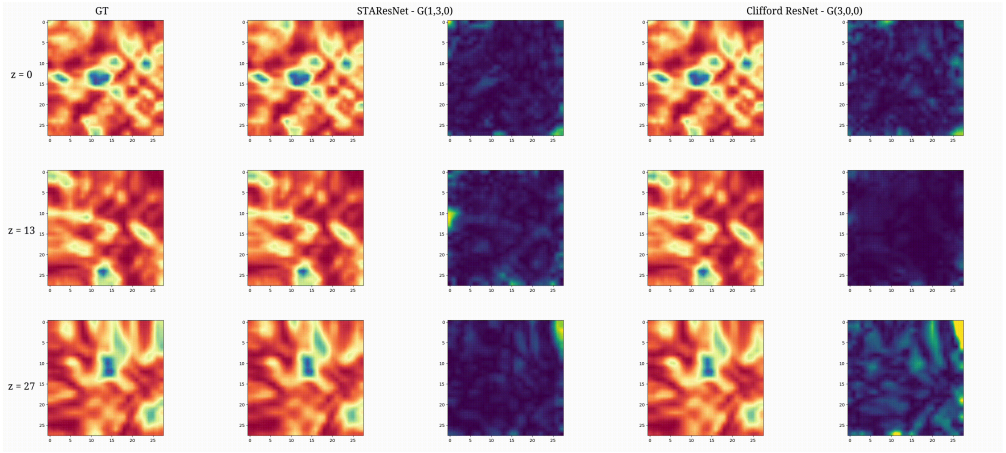

(c) $m = 7$



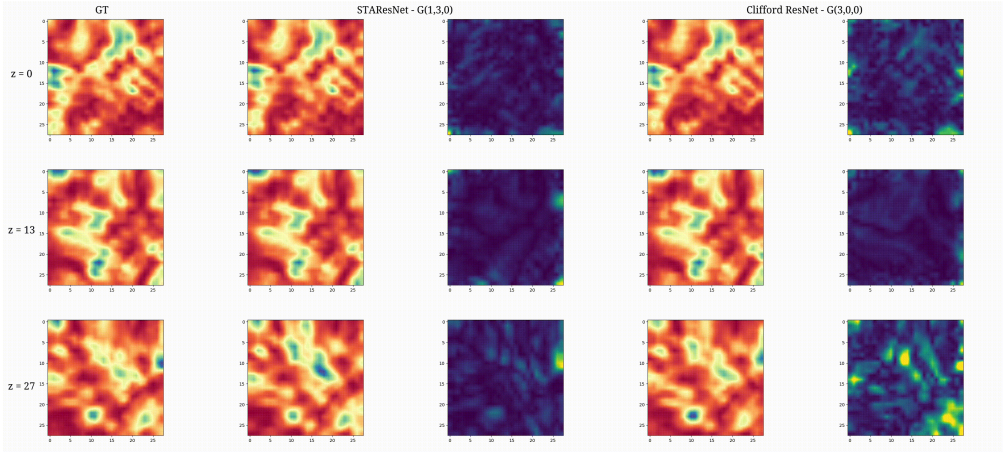

(d) $m = 10$

Figure 18: 3D GT $\mathbf{F}$, estimated $\hat{\mathbf{F}}$ and difference $|\mathbf{F} - \hat{\mathbf{F}}|$ for Clifford ResNet and STAResNet at different rollout steps. $\Delta t = 5$s. Transparency is proportional to intensity.

differences are shown in $viridis$ color map. Clifford ResNet yields significantly larger errors for each of the four rollout steps and across each of the three vertical slices considered.

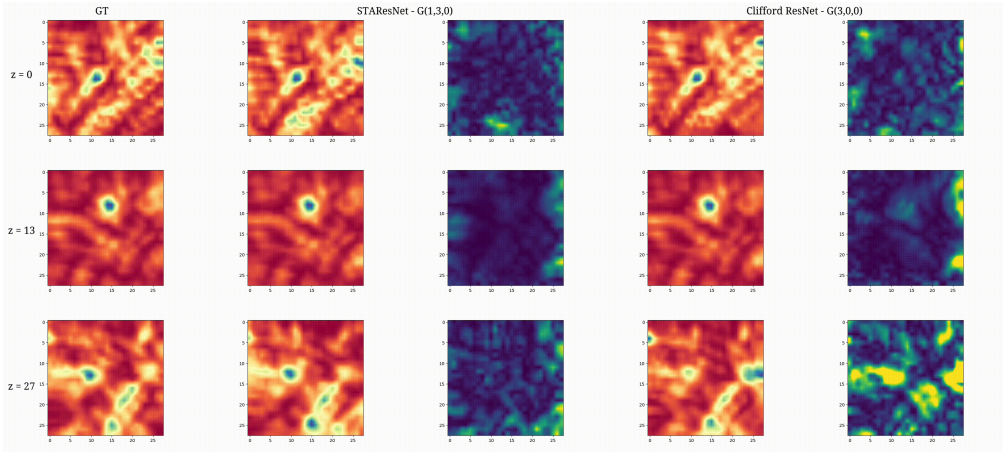



(a) $m = 3$

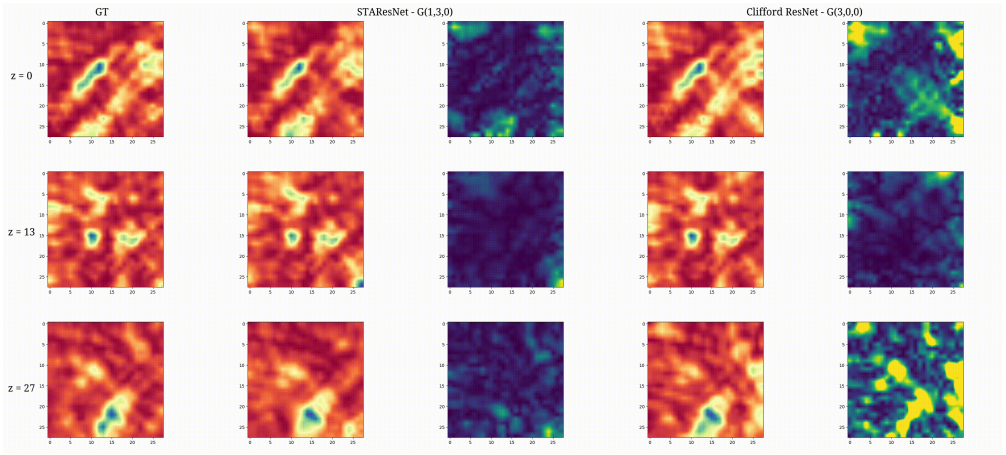



(b) $m = 5$

Figure 19: Slices of the 3D GT $\mathbf{F}$, estimated $\hat{\mathbf{F}}$ and difference $|\mathbf{F} - \hat{\mathbf{F}}|$ for Clifford ResNet and STAResNet at different rollout steps. $\Delta t = 5$s.




(a) $m = 8$




(b) $m = 10$

Figure 20: Slices of the 3D GT $\mathbf{F}$, estimated $\hat{\mathbf{F}}$ and difference $|\mathbf{F} - \hat{\mathbf{F}}|$ for Clifford ResNet and STAResNet at different rollout steps. $\Delta t = 5$s.

## 6 Conclusions & Future Work

We introduced STAResNet, a ResNet-inspired architecture that works with multivectors in STA, to shed light on the importance of choosing the right mathematical space for GA networks. We compared the performance of STAResNet, which is the first example of a network working in spacetime and with STA multivectors, with Clifford ResNet, working in vanilla GA, on the solution of Maxwell's equations.

It had already been demonstrated in the literature that a description of Maxwell's PDEs in STA offers an easier way to solve them numerically. We verified that this holds true also when the solution of Maxwell's PDEs is learnt, as long as the learning happens in STA.

STAResNet outperforms Clifford ResNet in 2 and 3D, at different sampling periods, in presence of obstacles either in a previously seen or unseen configuration, and over multiple time steps into the future. Most notably, STAResNet is able to generalise better over previously unseen data and achieve a lower error at a fraction of the number of trainable parameters as opposed to Clifford ResNet.

This substantial improvement is not surprising as it is grounded in the physics of the problem: working with Faraday bivectors in STA is a more natural and appropriate choice over the modelling of the EM fields as mixed-grade multivectors with vector and bivector components describing electric and magnetic fields, respectively.

We can then conclude that the choice of the right algebra in Clifford networks is a key factor to obtain more descriptive, compact and accurate learning pipelines.

Future works to further corroborate our thesis might include the study of different PDEs that can be expressed in more than one algebra and solving them with Clifford networks in those algebras.

## References


[1] Johannes Brandstetter, Rianne van den Berg, Max Welling, and Jayesh K Gupta. Clifford neural layers for pde modeling. *arXiv preprint arXiv:2209.04934*, 2022.

[2] Chris Doran and Anthony Lasenby. *Geometric algebra for physicists*. Cambridge University Press, 2003.

[3] Joan Lasenby and Leo Dorst. *Guide to geometric algebra in practice*, volume 2. Springer, 2011.

[4] Eckhard Hitzer. Introduction to clifford's geometric algebra. *Journal of the Society of Instrument and Control Engineers*, 51(4):338–350, 2012.

[5] JK Pearson and DL Bisset. Neural networks in the clifford domain. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 3, pages 1465–1469. IEEE, 1994.

[6] Eduardo Bayro-Corrochano and Sven Buchholz. Geometric neural networks. In *Algebraic Frames for the Perception-Action Cycle: International Workshop, AFPAC'97 Kiel, Germany, September 8–9, 1997 Proceedings 1*, pages 379–394. Springer, 1997.

[7] Justin Pearson. Clifford networks. In *Complex-Valued Neural Networks: Theories and Applications*, pages 81–106. World Scientific, 2003.

[8] Alberto Pepe, Sven Buchholz, and Joan Lasenby. GA-reLU: an activation function for geometric algebra networks applied to 2d navier-stokes PDEs. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024.

[9] Alberto Pepe, Joan Lasenby, and Sven Buchholz. Cgaposenet+ gcan: A geometric clifford algebra network for geometry-aware camera pose regression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6593–6603, 2024.

[10] David Ruhe, Jayesh K Gupta, Steven De Keninck, Max Welling, and Johannes Brandstetter. Geometric clifford algebra networks. In *International Conference on Machine Learning*, pages 29306–29337. PMLR, 2023.

[11] David Ruhe, Johannes Brandstetter, and Patrick Forré. Clifford group equivariant neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Alberto Pepe, Sven Buchholz, and Joan Lasenby. Clifford group equivariant neural network layers for protein structure prediction. In *Northern Lights Deep Learning Conference*, pages 205–211. PMLR, 2024.

[13] Johann Brehmer, Pim De Haan, Sönke Behrends, and Taco S Cohen. Geometric algebra transformer. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International conference on machine learning*, pages 3208–3216. PMLR, 2018.

[15] Senwei Liang, Shixiao W Jiang, John Harlim, and Haizhao Yang. Solving pdes on unknown manifolds with machine learning. *Applied and Computational Harmonic Analysis*, page 101652, 2024.

[16] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[17] Lulu Zhang, Tao Luo, Yaoyu Zhang, Zhi-Qin John Xu, Zheng Ma, et al. Mod-net: A machine learning approach via model-operator-data network for solving pdes. *arXiv preprint arXiv:2107.03673*, 2021.

[18] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021.

[19] Zhiping Mao, Ameya D Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.

[20] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021.

[21] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[22] Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *International conference on machine learning*, pages 2806–2823. PMLR, 2023.

[23] Carlos Uriarte, David Pardo, Ignacio Muga, and Judit Muñoz-Matute. A deep double ritz method (d2rm) for solving partial differential equations using neural networks. *Computer Methods in Applied Mechanics and Engineering*, 405:115892, 2023.

[24] Bing Yu et al. The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[25] Anthony Lasenby, Chris Doran, and Elsa Arcaute. Applications of geometric algebra in electromagnetism, quantum theory and gravity. *Clifford Algebras: Applications to Mathematics, Physics, and Engineering*, pages 467–489, 2004.

[26] Justin Dressel, Konstantin Y Bliokh, and Franco Nori. Spacetime algebra as a powerful tool for electromagnetism. *Physics Reports*, 589:1–71, 2015.