Conditioned Clifford-Steerable Kernels

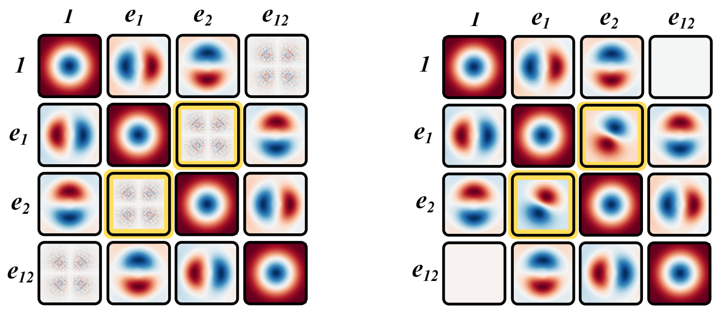
Anonymous Author(s)
Affiliation
Address
email

Abstract

Clifford-Steerable CNNs (CSCNNs) provide a unified framework that allows incorporating equivariance to arbitrary pseudo-Euclidean groups, including $\mathrm{E}(n)$ and Poincaré-equivariance on Minkowski spacetime. In this work, we analyze the shortcomings of the approach. We demonstrate that the kernel basis used in CSCNNs is not complete. Furthermore, we suggest to restore missing degrees of freedom by using an extra information obtained directly from data at virtually no cost. Our approach significantly and consistently outperforms baseline methods on PDE forecasting tasks, specifically fluid dynamics and relativistic electrodynamics.

1 Introduction

Physical systems are often associated with symmetries that govern their evolution. Therefore, it is desirable to respect those symmetries in the modelling process to faithfully capture the dynamics. *Equivariant Neural Networks* have proven to be strong candidates for learning such dynamics due to their ability to produce outputs that transform consistently under these symmetries. As part of the equivariant family, *Steerable Convolutional Neural Networks* ensure equivariance by enforcing a constraint over their convolution kernels. The steerability constraint is group-specific and has been solved and implemented for several common symmetry groups, such as for $\mathrm{O}(2)$ (Weiler and Cesa, 2019), $\mathrm{O}(3)$ (Geiger et al., 2022), and for general compact groups $\mathrm{E}(n)$ as demonstrated by Lang and Weiler (2020); Cesa et al. (2022).



(a) Default Clifford-steerable Kernel (b) Conditioned Clifford-steerable Kernel

Figure 1: Comparison of default vs. conditioned kernels for $\mathrm{O}(2)$. The missing angular frequency 2 basis kernels can be observed as the sparse scatters (which are just noise) in the antidiagonal of subfigure 1a. In comparison, subfigure 1b shows how our implementation recovers these missing kernels. Due to the kernel's input-dependent activation, the typical four-lobed quadrupole patterns may not be visible, despite being present.

Clifford Steerable Convolutional Neural Networks (CSCNNs), introduced by Zhdanov et al. (2024), extend steerability to a more general class of symmetries - pseudo-Euclidean group $\mathrm{E}(p, q)$, which covers, for instance, $\mathrm{E}(n)$-equivariance and Poincaré-equivariance on Minkowski spacetime. By using implicit parameterization of the kernels, CSCNNs are able to process feature fields defined on pseudo-Euclidean spaces, allowing CNNs to learn system dynamics in both non-relativistic (e.g. fluid dynamics) and relativistic (e.g. electromagnetism) scenarios. However, as pointed out by the authors, the original implementation yields incomplete kernel basis, which in turn limits the expressivity of the model as it is unable to represent certain degrees of freedom and therefore capture corresponding interactions (see the missing anti-diagonals in Figure 1).

The authors suggested that this degree of freedom may be recovered; however, this has yet to be proven or experimentally validated. We present *Conditioned Clifford-Steerable Kernels* which augment the kernel with an independent representation constructed from the input field, recovering the missing degrees of freedom, thereby achieving a strictly more expressive parameterization. The contributions of this paper are the following:

- We show that the original parametrization of CS-CNN kernels is missing kernels of angular frequency 2.
- We prove analytically that the missing degrees of freedom can be recovered by extending the input vector of the kernel by a global representation of the feature field, at little to no cost.
- We propose an efficient implementation and evaluate it on multiple PDE simulations, outperforming other convolutional baselines.

This paper is organised as follows: Section 2 provides a brief introduction to Steerable CNNs and the Clifford Algebra. Section 4 uncovers the main limitation of Clifford-Steerable CNN kernels, and introduces Conditioned Clifford-Steerable Kernels as a solution. After deriving their improved properties, the new kernels are empirically validated against the original implementation in Section 5.

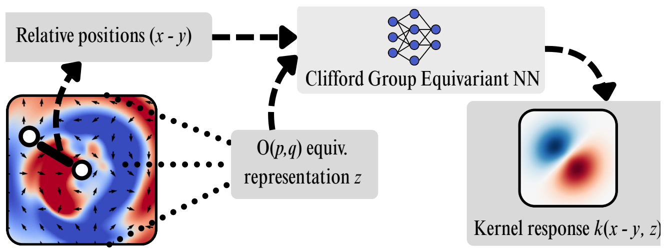

Relative positions $(x - y)$
Clifford Group Equivariant NN
$\mathrm{O}(p,q)$ equiv. representation z
Kernel response $k(x - y, z)$

Figure 2: Illustration of Conditioned Clifford-Steerable Kernels. Similar to CSCNNs, the parameterization is implicit, with the only difference that the kernel network is augmented with an equivariant representation of the input feature field.

2 Theoretical Background

2.1 Pseudo-Euclidean Spaces

Pseudo-Euclidean spaces generalize Euclidean space by allowing an indefinite metric, so the squared length of a nonzero vector can be positive, zero, or negative. Every finite dimensional pseudo-Euclidean space is isometric to the *standard* pseudo-Euclidean space, to which we restrict our attention (O'Neill, 1983).

Definition 2.1 (Standard pseudo-Euclidean vector spaces)**.** Let $e_1, \ldots, e_{p+q}$ be the canonical basis of the real vector space $\mathbb{R}^{p+q}$. Define an *inner product of signature* (p, q)-that is, a non-degenerate symmetric bilinear form with p positive and q negative eigenvalues-by $\eta^{p,q}(v_1, v_2) := v_1^\top \Delta^{p,q} v_2$ where $v_1, v_2 \in \mathbb{R}^{p+q}$, and $\Delta^{p,q} := \mathrm{diag}(\underbrace{1, \ldots, 1}_{p \text{ times}}, \underbrace{-1, \ldots, -1}_{q \text{ times}})$.

The resulting inner-product space $\mathbb{R}^{p,q} := \left(\mathbb{R}^{p+q},\ \eta^{p,q}\right)$ is the *standard pseudo-Euclidean vector space of signature* (p, q). This is the space where our feature vectors are defined, and also the space the symmetry groups we are interested in are acting on.

2.2 Clifford algebra and multivector fields

In previous works Ruhe et al. (2023); Zhdanov et al. (2024), it has been demonstrated that Clifford algebra $\mathrm{Cl}(\mathbb{R}^{p,q})$ is a representation space of the pseudo-orthogonal group $\mathrm{O}(p,q)$. In simpler words, it means that elements of the algebra - multivectors - can be used to represent actions of the group. Below we provide the formal definition. Let V be a vector space over a field $\mathbb{F}$, equipped with a bilinear form: $\eta : V \times V \to \mathbb{R}, \quad (v_1, v_2) \mapsto \eta(v_1, v_2)$. The *Clifford Algebra* generated by V is the unitary, associative, non-commutative algebra such that

$$v_i \bullet v_i = \eta(v_i, v_i) \cdot \mathbf{1}_{Cl(V,\eta)}, \quad \forall v_i \in V \subset \mathrm{Cl}(V, \eta) \tag{1}$$

Where $\bullet := \mathrm{Cl}(V,\eta) \times \mathrm{Cl}(V,\eta) \to \mathrm{Cl}(V,\eta)$ is the *geometric product*, an $\mathrm{O}(p,q)$-equivariant operation enabling us to multiply elements of the algebra while respecting their geometric properties. Such elements are *multivectors*, which can be constructed by linearly combining products of $v_1, ...v_n \in V$ the following way: $x = \sum_{i \in I} c_i \cdot v_{i,1} \bullet \cdots \bullet v_{i,l_i}$ where index set I is finite, $v_{i,k} \in V$ and $c_i \in \mathbb{R}$. We can decompose the algebra into vector subspaces $\mathrm{Cl}^{(n)}(V,\eta), \quad n = 0, \ldots, d$, which we refer to as *grades*. One grade $n = |A|$ is composed by choosing n out of the maximum d basis vectors of the underlying vector space V and has dimensionality $\dim \mathrm{Cl}^{(d)}(V,\eta) = \binom{d}{n}$. Grades $n = 0$ and $n = 1$ consist of scalars and vectors, respectively. For $n \geq 2$, one encounters bivectors, trivectors and higher-grade multivectors, which geometrically represent oriented points, areas and their higher-dimensional analogues. For a more in depth introduction to the Clifford Algebra, we refer the reader to Appendix D of Zhdanov et al. (2024).

2.3 Clifford Steerable CNNs

Clifford-Steerable CNNs process multivector fields, which are functions $f : \mathbb{R}^{p,q} \longrightarrow \mathrm{Cl}(\mathbb{R}^{p,q})^c$ assigning a feature $f(x)$ to each point $x \in \mathbb{R}^{p,q}$ in feature vector space $\mathrm{Cl}(\mathbb{R}^{p,q})^c$ where c denotes the feature channel dimension. They are required to commute with the pseudo-Euclidean symmetry group $\mathrm{E}(p,q) = (\mathbb{R}^{p,q}, +) \rtimes \mathrm{O}(p,q)$, which is the semi-direct product of the the *translation group* $(\mathbb{R}^{p,q}, +)$ and the *pseudo-orthogonal group* $\mathrm{O}(p,q) \;=\; \left\{ g \in \mathrm{GL}(\mathbb{R}^{p,q}) \mid g^\top \Delta^{p,q} g = \Delta^{p,q} \right\}$. Convolutions are by construction rotation equivariant, thus the task of constructing an $\mathrm{E}(p,q)$ equivariant CNN reduces to an $\mathrm{O}(p,q)$ equivariance requirement. Previous work by Weiler et al. (2023) showed that to achieve this, the convolution kernels $K : \mathbb{R}^{p,q} \longrightarrow \mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}\big)$ need to obey the $\mathrm{O}(p,q)$-*steerability* constraint $K(gx) = \rho_{\text{out}}(g)\, K(x)\, \rho_{\text{in}}(g)^{-1}, \ \forall g \in \mathrm{O}(p,q)\,,\ x \in \mathbb{R}^{p,q}$.

Several approaches have been proposed to solve this constraint, for example (Lang and Weiler, 2020; Cesa et al., 2022) analytically derived steerable kernel bases. Kernels can also be implemented as channel permutations in the matrix dimensions, such as in (Cohen and Welling, 2016; Bekkers et al., 2018). Both methods however are only suitable for compact groups, leaving out $\mathrm{O}(p,q)$. Numerical approaches have been proposed by (De Haan et al., 2020; Shutty and Wierzynski, 2023) based on Lie-algebra irreps, however only for connected subgroups of $\mathrm{O}(p,q)$. (Zhdanov et al., 2023) solved the steerability constraint *implicitly* by parametrising the kernel using an equivariant neural network. This method was adapted to work on Clifford algebras, by using *Clifford Group Equivariant neural networks* (Ruhe et al., 2023) to parametrise the kernels of *Clifford Steerable CNNs* (Zhdanov et al., 2024).

The implicit kernel K serving as the convolution kernel of *Clifford Steerable CNN* is a composition of two operators: $K \;=\; H \circ \mathcal{K}$, where $\mathcal{K}$ is the parametrising *Kernel Network*, and H denotes a *kernel head*, which transforms the kernel network output to be used as a convolution kernel on multivector fields.

Definition 2.2 (Kernel network)**.** The Kernel network $\mathcal{K} : \mathbb{R}^{p,q} \;\longrightarrow\; \mathrm{Cl}(\mathbb{R}^{p,q})^{\,c_{\mathrm{out}} \times c_{\mathrm{in}}}$ is an $\mathrm{O}(p,q)$-equivariant *Clifford-group equivariant neural network* (CGENN) (Ruhe et al., 2023). Let $k = 0, \ldots d$ denote the multivector grades and $w_{mn}^{(k)} \in \mathbb{R}$ weights. A CGENN is built only from operations that are $\mathrm{O}(p,q)$-equivariant by construction:

1. *Grade-wise linear maps* $L_m^{(k)}(x_1, \ldots, x_{c_{\mathrm{in}}}) \;:=\; \sum_{n=1}^{c_{\mathrm{in}}} w_{mn}^{k} \cdot x_n^{(k)}$, which act independently inside every irreducible sub-representation (grade) of the respective Clifford metric space $\mathrm{Cl}(\mathbb{R}^{p,q})^{(k)}$;

2. *Weighted geometric-product layers* $P^{(k)}(x_1, x_2) := \sum_{m=0}^{d} \sum_{n=0}^{d} w_{mn}^{k} \cdot \left(x_1^{(m)} \bullet x_2^{(n)}\right)^{(k)}$

Additionally, the network contains component-wise nonlinearities and norm-based normalisations that depend only on the quadratic form and therefore remain invariant under the group action. Because every CGENN layer is a multivector *polynomial*, the universal result that "all multivector polynomials are $\mathrm{O}(p, q)$-equivariant" applies (Ruhe et al., 2023), which guarantees that their composition in $\mathcal{K}$ satisfies $\mathcal{K}(gv) = \rho_{\mathrm{Cl}(\mathbb{R}^{p,q})}(g)\, \mathcal{K}(v)$ for all $g \in \mathrm{O}(p, q)$.

Definition 2.3 (Kernel head)**.** The kernel head: $H : \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}} \longrightarrow \mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}\big)$ is a grade-projected, partially evaluated geometric product that turns each multivector component $\mathrm{k} = [\mathrm{k}_{ij}]$ into an $\mathbb{R}$-linear map between channel stacks:

$$\big[H(\mathrm{k})\, f\big]_i^{(k)} = \sum_{j \in [c_{\mathrm{in}}]} \sum_{m,n=0}^{d} w_{mn,ij}^{(k)} \left(\mathrm{k}_{ij}^{(m)} \bullet f_j^{(n)}\right)^{(k)}.$$

The scalars $w_{mn,ij}^{(k)}$ (shared over space) merely re-weight already equivariant terms; hence H itself is $\mathrm{O}(p, q)$-equivariant. A proof can be found in Appendix E.1 of Zhdanov et al. (2024). Since both $\mathcal{K}$ and H are $\mathrm{O}(p, q)$-equivariant, their composition $K = H \circ \mathcal{K}$ obeys the steerability constraint. Convolving with K therefore yields an operator that is exactly $\mathrm{E}(p, q)$-equivariant (Theorem 3.4 in Zhdanov et al. (2024)).

3 Missing basis of regular CSCNN kernels

This section takes a representation theoretic investigation of the original CSCNN kernel implementation by Zhdanov et al. (2024), uncovering the missing kernel basis. To overcome this limitation, we introduce *Conditioned-Clifford Steerable Kernels*, that recover the missing kernels without breaking equivariance, allowing for the full parametrization of the convolution kernels. Note that here we restrict our attention to $\mathrm{O}(2)$ acting on $\mathbb{R}^2$, however, the same decomposition can be used to generalise the analysis to $\mathrm{O}(n)$ and $\mathrm{O}(p, q)$.

To understand what is meant by angular frequency 2 kernels, it is worth investigating the kernel functions in the frequency domain. We can do this, as it was shown by Weiler and Cesa (2019); Weiler et al. (2018), the irreducible representations of $\mathrm{O}(2)$ are one- or two–dimensional circular harmonic modes. More precisely, every kernel obeying the steerability constraint admits a polar decomposition

$$K(r, \varphi) = \sum_{m \in \mathbb{Z}} R_m(r)\, Y_m(\varphi), \quad Y_m(\varphi) = e^{\mathrm{i}m\varphi}.$$

Where *m* is the grade (irrep) of the multivector output and where the angular factors Y_m are fixed and only the radial parts are freely learned R_m. As such, the kernel can be fully expressed by the different angular frequency bases and their learned radial part. Depending on the field types a convolution operator is mapping between, its kernel is required to contain specific angular frequency bases for the mapping to be fully complete.

3.1 Clebsch–Gordan requirements for kernels

To identify what angular frequencies are required for convolutions, we use the *Clebsch–Gordan* decomposition of its tensor products, which was adapted by Lang and Weiler (2020) for kernel irreps. When a kernel maps one field type (irrep j) to another field type (irrep l), the tensor product $D^{(j)} \otimes D^{(l)}$ must supply the trivial representation so that contraction with an input feature can yield an output feature. For multivector grades $\mathrm{Cl}(\mathbb{R}^{2,0})^{(k)}$ the Clebsch-Gordan decomposition $D^{(j)} \otimes D^{(l)} \cong \bigoplus_{m=|j-l|}^{j+l} D^{(m)}$ is shown in Table 1

Examining $j = l = 1$ (so the vector field $\rightarrow$ vector field mapping) yields the set of required harmonic blocks $m \in \big\{|1-1|,\ 1+1\big\} = \big\{0,\ 2\big\}$. Which means that if we want our mapping from vector $\rightarrow$ vector to be complete, we need bases with angular frequencies $m \in \{0, 2\}$ to be represented in the kernel. The scalar part $m = 0$ can be recovered easily: it already "lives" in grades 0 and 2. However,

Table 1: Clifford field types by grade and $O(2)$ irreducible representations

grade	field type	irrep	angular frequency
$\mathrm{Cl}(\mathbb{R}^{2,0})^{(0)}$	scalar	$D^{(0)}$	$m = 0$ (even)
$\mathrm{Cl}(\mathbb{R}^{2,0})^{(1)}$	vector	$D^{(1)}$	$m = 1$
$\mathrm{Cl}(\mathbb{R}^{2,0})^{(2)}$	pseudoscalar	$D^{(0)}_{\text{odd}}$	$m = 0$ (odd)

the quadrupole part with angular frequency $m = 2$ can appear only through the symmetric traceless component of a product of two *independent* vectors (Zou and Zheng, 2003).

The implicit kernel of a Clifford-Steerable CNN, $K = H \circ \mathcal{K}$, receives a *single* grade-1 vector per edge, namely the displacement $v := x - y \in \mathbb{R}^2 \cong \mathrm{Cl}(\mathbb{R}^2)^{(1)}$ (Zhdanov et al., 2023, 2024). Inside of kernel network $\mathcal{K}$, the transformations applied to this vector (as detailed in Definition 2.2) only do the following:

1. *Grade-wise linear maps* reshuffle channels but never change the spatial irrep: a vector stays a vector ($m = 1$).
2. *Geometric products* take two copies of the same vector. Their product immediately projects (due to the Clifford rules in Equation (1)) into grade 0 or grade 2, i.e. frequency $m = 0$.

Because both operands derive from the same direction, no symmetric-traceless rank-2 object-and hence no $m = 2$ harmonic-can be formed at any depth inside a single kernel network. This is not changed by the kernel head H either, since its operation is also a (partially evaluated) geometric product. This results in the incomplete vector$\to$ vector mapping of the convolutional layers, and the original CSCNN architecture. To complete the parametrisation, the interaction of independent vectors are needed.

4 Conditioned Clifford-Steerable Kernels

4.1 Conditioned Steerable Convolution

We follow the derivation of a steerable convolution from Weiler et al. (2023) (Theorem 4.3.1). Let us start with a convolutional operator

$$I_k\left[F\right](x) := \int_{\mathbb{R}^d} dy\, K(x-y)F(y) \tag{2}$$

that is parameterized by a kernel $K : \mathbb{R} \to \mathbb{R}^{c_{\text{out}} \times c_{\text{in}}}$. For the convolutional layer to be G-equivariant, the kernel must be G-steerable, that is,

$$K(gx) = \rho_{\text{out}}(g)K(x)\rho_{\text{in}}(g)^{-1} \qquad \forall x \in \mathbb{R}^d, g \in G. \tag{3}$$

where ρ_{in} and ρ_{out} are representations of input and output feature fields, respectively.

Let us now introduce an equivariant function $T : L^2(\mathbb{R}^d, \mathbb{R}^c) \to \mathbb{R}^c$ which computes a vector representation of a feature field. We will use the function to augment the kernel function, yielding the following quasi-linear transformation

$$I_k^T\left[F\right](x) := \int_{\mathbb{R}^d} dy\, K(x-y,\, T[F])F(y). \tag{4}$$

Since the conditioning function T is equivariant, we have by definition $T[g.F] = \rho(g)T[F]$.

Lemma 1. The quasi-linear map 4 is G-equivariant if and only if

$$K(gx, \rho_{\text{in}}(g)T[F]) = \rho_{\text{Hom}}(g)\left(K(x, T[f_{\text{in}}])\right)$$

We provide the proof of the lemma in the Appendix B.1.

4.2 Conditioned Clifford-Steerable Convolution

The quasi-linear map provides a framework to tackle the issue of missing degrees of freedom in Clifford-Steerable convolutions. As discussed, the kernel basis of the latter is incomplete, as it relies on implicit kernels that are unable to generate high-frequency components from a positional vector alone. Furthermore, the key idea of our approach is to condition the implicit kernel on relative position, as well as a field representation, whose interaction together will yield the otherwise missing high-frequency components of the kernel basis.

As discussed in Section 2.3, in the framework of Clifford-Steerable CNNs, the kernel K is defined as a composition $K = H \circ \mathcal{K}$. The kernel head H transforms the output of the kernel network $\mathcal{K}$ to matrix-valued kernels and is agnostic to the form of $\mathcal{K}$. Furthermore, we only need to adapt the kernel network to be compatible with conditioned convolution. Specifically, let us define conditioned kernel network $\mathcal{K}_T : \mathbb{R}^{p,q}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\text{in}}} \to \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\text{out}} \times c_{\text{in}}}$ which now additionally takes the multivector representation of the input field generated by the conditioning function $T : L^2(\mathbb{R}^d, \mathrm{Cl}(\mathbb{R}^{p,q})^c) \to \mathrm{Cl}(\mathbb{R}^{p,q})^c$. Since we use CGENNs to implement $\mathcal{K}_T$, the function is $\mathrm{O}(p,q)$-equivariant by definition. Having the conditioned kernel network, we define Conditioned-Steerable kernels as $K = H \circ \mathcal{K}_T$.

Lemma 2 (Equivariance of conditioned Clifford-steerable kernels). Every conditioned Clifford-steerable kernel $K = H \circ \mathcal{K}_T$ is $\mathrm{O}(p,q)$-steerable w.r.t. the standard action $\rho(g) = g$ and ρ_{Hom}:

$$K(gv, \rho_{\text{in}}(g)T[F]) = \rho_{\text{Hom}}(g)\left(K(v, T[f_{\text{in}}])\right) \qquad \forall g \in O(p,q),\ v \in \mathbb{R}^{p,q}.$$

We provide the proof in Appendix B.2.

Equipped with conditioned kernels, we are now able to tackle the issue with missing degrees of freedom in the original framework of CSCNNs:

Corollary 4.1 (Completeness of the kernel basis)**.** Let the output of $T[f_{\text{in}}]$ have non-trivial grade 1 component. Then the kernel basis of conditioned Clifford-Steerable CNNs is complete.

5 Experimental Results

To test the efficacy of our new kernel implementation, we compared its performance to the original CSCNN model and several strong baselines in three PDE learning tasks: the 2-dimensional ($\mathbb{R}^2$) *Navier-Stokes (NS2)* equations, the 3-dimensional ($\mathbb{R}^3$) *Maxwell (MW3)* equations, and the relativistic 2-dimensional ($\mathbb{R}^{1,2}$) *Maxwell (MW2)* equations.

5.1 Notes on implementation

For the simplicity, we choose the conditioning function T to be the global mean pooling operation. The operation is equivariant, as proven in Weiler et al. (2023). Compared to the original CSCNNs, our model's only additional computational cost comes from computing the condition. However, since this step uses highly optimized operations (like mean and max pooling), the resulting overhead is minimal. The implementation is done in JAX (Bradbury et al., 2018). More details can be found in Appendix A.1.

5.2 Experiment Setup

The goal of each experiment is to empirically learn the dynamics of the systems from numerical simulations, and predict future states based on previous timesteps (Gupta and Brandstetter, 2022). For the non-relativistic tasks, variables describing the states in the 4 previous timesteps are taken as input, with each timestep serving as an additional feature channel, and the goal is to predict the subsequent state. Although improper, this setup allows us to compare our results to prior models which otherwise would not be able to handle a full spacetime. For Navier-Stokes, the input state is described by a velocity vector for the vector part, and the pressure field for the scalar part, with the bivector part padded with zeros to keep the multivector structure consistent. For the MW3 task, the inputs are vector electric fields and bivector magnetic fields, with the rest of the blades padded. For the relativistic Maxwell task, time forms its own spacetime dimension of size 32, thus the task becomes predicting the next 32 timesteps based on the previous 32. The input is an electromagnetic field, which forms the bivector part of a multivector. This setup properly incorporates the time dimension into spacetime.

5.3 Model Architectures

Our model is built upon the ResNet architecture, where we substitute the standard convolutional layers with our novel Conditioned Clifford-Steerable convolutions. We compare Conditioned CSCNNs against architecturally similar networks: the standard ResNet, the Clifford ResNet (Brandstetter et al., 2022), the O(n)-steerable CNN (Weiler and Cesa, 2019; Weiler et al., 2023) and the original Clifford-Steerable CNN (Zhdanov et al., 2024). Additionally, we evaluate the prominent PDE learner Fourier Neural Operator (FNO) (Li et al., 2021) and its $D_4 < O(2)$ equivariant variation, G-FNO (Helwig et al., 2023). All models were scaled to match the parameter count of the basic ResNet architecture.

5.4 Results

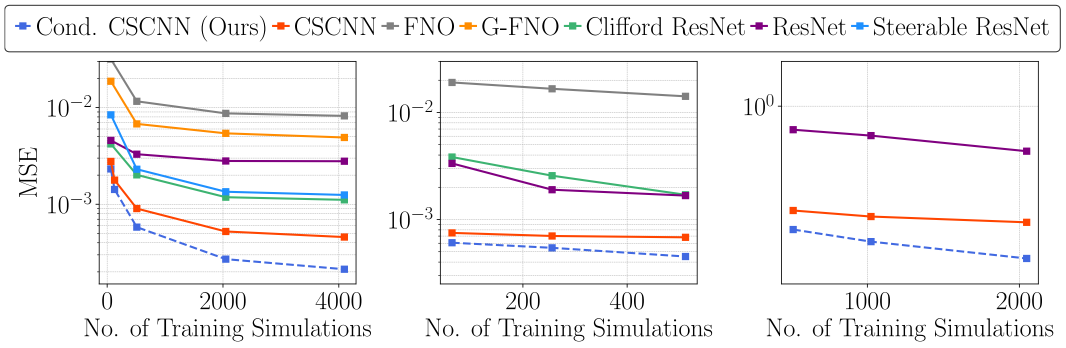


Figure 3: Mean squared errors for (1) Navier-Stokes $\mathbb{R}^2$, (2) Maxwell $\mathbb{R}^3$, and (3) relativistic Maxwell $\mathbb{R}^{1,2}$ simulation tasks as a function of the simulations included in the training dataset. Conditioned-CSCNNs outperform all baselines, with the gap widening as data increases.

Figure 3 shows the MSE on the test set. On all tasks, our *Conditioned-CSCNN* outperforms every baseline, including the original CSCNN implementation. Its advantage becomes more pronounced as more simulations are included in the training set, which is an experimental proof of the added model complexity from the recovered angular frequency 2 kernels. Indeed, as shown in Figure 4, the highest MSE improvement against the baseline CSCNN comes from the vector component of the loss, suggesting complete *vector*$\to$ *vector* mappings of the convolution layers. Note that we can only test this for the NS experiment, where we have ground truth data for the vector output.

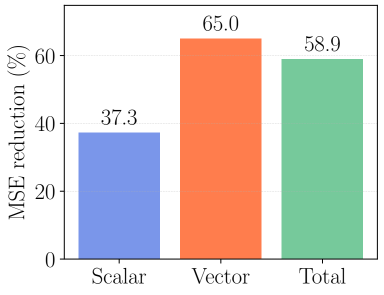


Figure 4: Per-grade MSE reduction of our model vs. baseline CSCNN for the Navier–Stokes experiment with $n = 2048$ simulations in the training dataset. We see the largest improvement in the vector component.

To test whether our model is still equivariant, we calculated relative equivariance error between the output of the transformed input vs the transformed output of the original input: $\mathrm{err}\big(f; g, x\big) =$

Table 2: Relative equivariance errors for Clifford-Steerable convolutions

Kernel	Relative error (mean)	Relative error (max)
Default CS Convolution	2.4×10^{-7}	1.1×10^{-3}
Conditioned CS Convolution	3.4×10^{-7}	5.9×10^{-4}

$\frac{|f(gx) - gf(x)|}{|f(gx) + gf(x)|}$. Table 2 shows the $\mathrm{O}(2)$ relative errors for the original and our conditioned convolutions. While the error is slightly higher for the conditioned convolutions, it is still equivariant up to numerical artifacts.

6 Conclusion

We introduce Conditioned Clifford-Steerable CNNs, a generalization of Clifford-Steerable CNNs that employs quasi-linear convolutions. By conditioning the convolution kernel on representations of the input feature field, our method recovers the degrees of freedom that were absent in the original framework. Conditioned Clifford-Steerable CNNs can therefore express complete mappings between multivector fields on (pseudo-)Euclidean spaces while maintaining consistent transformation properties under the general Euclidean group $\mathrm{E}(p, q)$. This capability makes our approach particularly well-suited for learning the physical dynamics of both relativistic and non-relativistic systems. We demonstrate this advantage across various PDE learning tasks, where Conditioned CSCNNs consistently outperform strong baselines, including the original CSCNN implementation.

Limitations Our work has two primary limitations. First, we analytically derive the missing basis kernels only for the special case of $\mathrm{O}(2, 0)$, leaving the proof for the general case of $\mathrm{O}(p, q)$ as future work. However, our empirical results demonstrate that addressing this limitation yields dramatically improved performance across all tasks, indicating a more expressive underlying transformation. Second, our implementation has significantly slower runtimes than baseline CSCNNs for higher-dimensional groups, which we attribute to suboptimal optimization of conditioned CNNs. We anticipate that more efficient implementations—such as those utilizing Triton—could resolve these computational bottlenecks.

Impact Statement Our approach has broad applications in equivariant physical modeling, including neural PDE surrogates, weather forecasting, and fluid dynamics simulation. By incorporating known symmetries into deep learning models, one can improve data efficiency and robustness to domain shifts, which is important in real-world applications. Additionally, we also hope to apply the framework to robotics, where exploiting symmetries can dramatically improve sample efficiency when training a physical robotic system Wang et al. (2022).

References

Bekkers, E. J., Lafarge, M. W., Veta, M., Eppenhof, K. A., Pluim, J. P., and Duits, R. (2018). Roto-translation covariant convolutional networks for medical image analysis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pages 440–448. Springer.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

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

Cesa, G., Lang, L., and Weiler, M. (2022). A program to build e (n)-equivariant steerable cnns. In *International conference on learning representations*.

Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR.

De Haan, P., Weiler, M., Cohen, T., and Welling, M. (2020). Gauge equivariant mesh cnns: Anisotropic convolutions on geometric graphs. *arXiv preprint arXiv:2003.05425*.

Filipovich, M. J. and Hughes, S. (2022). Pycharge: An open-source python package for self-consistent electrodynamics simulations of lorentz oscillators and moving point charges. *Computer Physics Communications*, 274:108291.

Geiger, M., Smidt, T., Alby, M., Miller, B. K., Boomsma, W., Dice, B., Lapchevskyi, K., Weiler, M., Tyszkiewicz, M., Batzner, S., et al. (2022). Euclidean neural networks: e3nn. *Preprint at https://doi. org/10.48550/arXiv*, 2207.

Gupta, J. K. and Brandstetter, J. (2022). Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*.

Helwig, J., Zhang, X., Fu, C., Kurtin, J., Wojtowytsch, S., and Ji, S. (2023). Group equivariant fourier neural operators for partial differential equations. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Holl, P., Koltun, V., and Thuerey, N. (2020). Learning to control pdes with differentiable physics. *CoRR*, abs/2001.07457.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Lang, L. and Weiler, M. (2020). A wigner-eckart theorem for group equivariant convolution kernels. *arXiv preprint arXiv:2010.10952*.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*.

O'Neill, B. (1983). Semi-riemannian geometry with applications to relativity. *Pure and Applied Mathematics/Academic Press, Inc.*

Ruhe, D., Brandstetter, J., and Forré, P. (2023). Clifford group equivariant neural networks. *Advances in Neural Information Processing Systems*, 36:62922–62990.

Shutty, N. and Wierzynski, C. (2023). Computing representations for lie algebraic networks. In *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, pages 1–21. PMLR.

Wang, D., Jia, M., Zhu, X., Walters, R., and Platt, R. W. (2022). On-robot learning with equivariant models. In *Conference on Robot Learning*.

Wang, R., Walters, R., and Yu, R. (2021). Incorporating symmetry into deep dynamics models for improved generalization. In *International Conference on Learning Representations*.

Weiler, M. and Cesa, G. (2019). General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32.

Weiler, M., Forré, P., Verlinde, E., and Welling, M. (2023). Equivariant and coordinate independent convolutional networks. *A Gauge Field Theory of Neural Networks*, page 110.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. (2018). 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *CoRR*, abs/1807.02547.

Zhdanov, M., Hoffmann, N., and Cesa, G. (2023). Implicit convolutional kernels for steerable cnns. *Advances in Neural Information Processing Systems*, 36:17395–17407.

Zhdanov, M., Ruhe, D., Weiler, M., Lucic, A., Brandstetter, J., and Forré, P. (2024). Clifford-steerable convolutional neural networks. *arXiv preprint arXiv:2402.14730*.

Zou, D. and Zheng, Q.-s. (2003). Maxwell's multipole representation of traceless symmetric tensors and its application to functions of high-order tensors. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 459:527–538.

A Appendix

A.1 Implementation details

The basic ResNet architecture that was used for constructing the Conditioned-CSCNN, the CSCNN, the Clifford ResNet and the $\mathrm{O}(n)$-steerable ResNet were based on the setup of Wang et al. (2021); Brandstetter et al. (2022); Gupta and Brandstetter (2022). They consist of 8 residual blocks with 7×7 and $7 \times 7 \times 7$ sized kernels for the 2D and 3D experiments respectively. We used two embedding and two output layers. For constructing the FNO (Li et al., 2021) and G-FNO (Helwig et al., 2023), we followed the original implementations. The models have approximately 7M parameters for the Navier Stokes and 1.5M for Maxwell's experiments.

A.1.1 Conditioned CS kernel implementation

In constructing the Conditioned Clifford-Steerable Kernels, we built on the architecture described in Appendix A of Zhdanov et al. (2024). We form the conditioning vector by a *masked spatial mean* computed for each channel c and blade/grade k. On a grid $\Omega \subset \mathbb{R}^d$ and with the indicator χ_{B_r} of the largest centered ball B_r (the circular/spherical mask, which we explain below), the pooled stack is

$$\left(T_{\text{pool}}[f_{\text{in}}]\right)_c^{(k)} := \frac{1}{|\Omega|} \sum_{x \in \Omega} \chi_{B_r}(x)\, f_{\text{in},c}^{(k)}(x),$$

The resulting conditioning multivector stack is then concatenated with the relative position vector to form the input to the Kernel Network.

Circular mask : In practice, the multivector feature fields are often discretised as square shaped arrays $(c, X_1 \dots X_d, 2^d)$. Thus, an operation defined solely on this finite grid will break equivariance towards the corners of the domain, as shown in Figure 5. To overcome this, we apply circular masking (set grid values outside of the circle/sphere to 0) before the pooling operation, making it $\mathrm{O}(n)$-equivariant in the continuum.

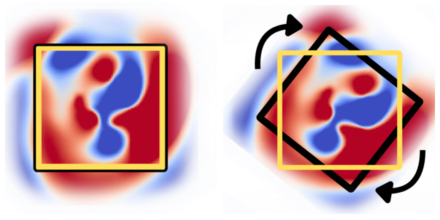


Figure 5: Illustration of how the receptive field of the finite, discretised kernel changes under rotations. The square shape causes any operation defined on it to break equivariance towards the corners.

A.1.2 Training details

We adopted the optimised hyperparameters from Zhdanov et al. (2024) for all of our models and experiments. We used Adam optimizer (Kingma and Ba, 2017) with cosine learning rate scheduler. Each model was trained to convergence. The models were trained on one node of Snellius, the Dutch national computing cluster with 4 NVIDIA A100 GPUs.

A.1.3 Datasets

2D Navier Stokes equations The ground truth simulations are taken from Gupta and Brandstetter (2022) and are based on ΦFlow by Holl et al. (2020). From the corresponding validation and test partitions, we randomly sampled 1024 trajectories. The simulations were generated on a 128×128 pixel grid with uniform spatial spacing $\Delta x = \Delta y = 0.25\,\mathrm{m}$ and a time step of $\Delta t = 1.5\,\mathrm{s}$.

3D Maxwell's equations Within the non-relativistic $\mathrm{Cl}(\mathbb{R}^{3,0})$ setting, we represent the electric field E as a vector field and the magnetic field B as a bivector field. The dataset, drawn from Brandstetter et al. (2022), consists of 3D Maxwell simulations discretized on a $32 \times 32 \times 32$ voxel grid with uniform spacing $\Delta x = \Delta y = \Delta z = 5 \times 10^{-7}\,\mathrm{m}$ and time step $\Delta t = 50\,\mathrm{s}$. The validation and test splits together contain 128 simulations.

2D relativistic Maxwell's equations We generate a dataset for Maxwell's equations in 2+1D spacetime ($\mathbb{R}^{1,2}$) utilizing the `PyCharge` simulation package by Filipovich and Hughes (2022). The simulations model the dynamics of electromagnetic fields emitted by oscillating and orbiting point charges moving at relativistic speeds. The spacetime grid is discretized with a resolution of 128×128 points, corresponding to a spatial extent of $50\,\mathrm{nm}$ and a temporal duration of $3.77 \cdot 10^{-14}$ s.

Each simulation is initialized with a unique configuration of charge sources, governed by the following randomly sampled parameters: **Source Composition:** A combination of 2 to 4 oscillating charges and 1 to 2 orbiting charges, with integer magnitudes sampled uniformly from the range $[-3e, 3e]$. **Initial Conditions:** Sources are placed uniformly on the grid with a predefined minimum separation. Each is assigned a random linear velocity and either oscillates in a random direction or orbits with a random radius. **Relativistic Constraint:** Oscillation/rotation frequencies and velocities are sampled such that the total particle velocity does not exceed $0.85c$, a necessary constraint to ensure the stability of the PyCharge solver.

To handle the wide dynamic range of the resulting field strengths, we apply a normalization scheme. The generated field bivectors are divided by their Minkowski norm and then multiplied by the logarithm of that norm. Although Minkowski norms can be zero or negative, we found they were consistently positive in our generated data. Finally, we filter numerical artifacts by removing any outlier simulations that exhibit a standard deviation greater than 20. The curated dataset is split into 2048 training, 256 validation, and 256 test simulations.

B Proofs

B.1 Proof of Lemma 1

Proof. We seek to proof the G-equivariance of the quasi-linear convolution

$$I_k^T\left[F\right](x) := \int_{\mathbb{R}^d} dy\, K(x-y,\, T[F])F(y). \tag{5}$$

where the conditioning function T is G-equivariant. We will use the notation of Weiler et al. (2023), where the action of a group G on a feature vector field of type ρ is written as

$$\left[tg \rhd_\rho F\right](x) = \rho(g)F\left((tg)^{-1}x\right). \tag{6}$$

The G equivariance of the convolutional layer then formally means that

$$I_k^T\left[g \rhd_{\rho_{\text{in}}} F\right] = g \rhd_{\rho_{\text{out}}} I_k^T\left[F\right], \qquad \forall g \in G \tag{7}$$

where ρ_{in} and ρ_{out} stand for the type of input and output feature fields, respectively.

By expanding the left-hand side of the equation above, we obtain

$$I_k^T\left[g \rhd_{\rho_{\text{in}}} F\right] = \int_{\mathbb{R}^d} dy\, K(x-y,\, T\left[g \rhd_\rho F\right])\left[g \rhd_\rho F\right](y) \tag{8}$$

$$\overset{T\text{ equiv.}}{=} \int_{\mathbb{R}^d} dy\, K(x-y,\, \rho_{\text{in}}(g)T\left[F\right])\rho_{\text{in}}F\left(g^{-1}y\right) \tag{9}$$

$$\overset{g^{-1}y \to y}{=} \int_{\mathbb{R}^d} dy\, K(x-gy,\, \rho_{\text{in}}(g)T\left[F\right])\rho_{\text{in}}F\left(y\right) \tag{10}$$

The right-hand side then yields

$$g \rhd_{\rho_{\text{out}}} I_k^T\left[F\right] = \int_{\mathbb{R}^d} dy\, \rho_{\text{out}}(g)K\left(g^{-1}x-y,\, T\left[F\right]\right)F(y) \tag{11}$$

These expressions agree for any $g \in G$ and any feature map $F \in L^2(\mathbb{R}^d, \mathbb{R}^{c_{\text{in}}})$ if and only if

$$K(x-gy,\, \rho_{\text{in}}(g)T\left[F\right])\rho_{\text{in}} = \rho_{\text{out}}(g)K\left(g^{-1}x-y,\, T\left[F\right]\right). \tag{12}$$

After substitution $g^{-1}x \to x$ and $x - y = x$, we obtain the constraint:

$$K(gx,\ \rho_{\text{in}}(g)T\,[F]) = \rho_{\text{out}}(g)K\,(x,\ T\,[F])\,\rho_{\text{in}}(g)^{-1} \equiv \rho_{\text{Hom}}(g)\,(K\,(x,\ T\,[F])) \tag{13}$$

□

B.2 Proof of Lemma 2

Proof. $\mathcal{K}_T$ is $O(p,q)$-equivariant by definition, and the kernel head H is $O(p,q)$-equivariant by Proposition 3.2 in Zhdanov et al. (2024). The $O(p,q)$-equivariance of their composition follows from

$$\begin{aligned} K\left(gv, \rho_{Cl}^{c_{\text{in}}}(g)\,T[f_{\text{in}}]\right) =& H\left(\mathcal{K}_{\mathcal{T}}\left(gv, \rho_{Cl}^{c_{\text{in}}}(g)\,T[f_{\text{in}}]\right)\right) & (14)\\ \overset{\mathcal{K}_T\text{ equiv.}}{=}& H\left(\rho_{Cl}^{c_{\text{out}}\times c_{\text{in}}}(g)\mathcal{K}_T(v, T[f_{\text{in}}])\right) & (15)\\ =&\rho_{\text{Hom}}(g)H(\mathcal{K}_T(v, T[f_{\text{in}}])) & (16)\\ =&\rho_{\text{Hom}}(g)K(v, T[f_{\text{in}}]) & (17)\end{aligned}$$

□