# Clifford-Steerable Convolutional Neural Networks

**Maksim Zhdanov** [1] **David Ruhe** [*1 2 3] **Maurice Weiler** [*1] **Ana Lucic** [4] **Johannes Brandstetter** [5 6] **Patrick Forré** [1 2]

## Abstract

We present *Clifford-Steerable Convolutional Neural Networks* (CS-CNNs), a novel class of $\mathrm{E}(p,q)$-equivariant CNNs. CS-CNNs process *multivector fields* on pseudo-Euclidean spaces $\mathbb{R}^{p,q}$. They cover, for instance, $\mathrm{E}(3)$-equivariance on $\mathbb{R}^3$ and Poincaré-equivariance on Minkowski spacetime $\mathbb{R}^{1,3}$. Our approach is based on an implicit parametrization of $\mathrm{O}(p,q)$-steerable kernels via Clifford group equivariant neural networks. We significantly and consistently outperform baseline methods on fluid dynamics as well as relativistic electrodynamics forecasting tasks.

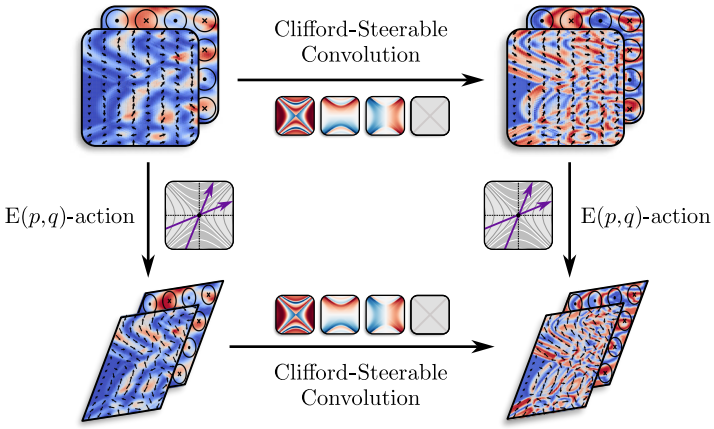



*Figure 1.* CS-CNNs process multivector fields while respecting $\mathrm{E}(p,q)$-equivariance. Shown here is a Lorentz-boost $\mathrm{O}(1,1)$ of electromagnetic data on 1+1-dimensional spacetime $\mathbb{R}^{1,1}$.

## 1. Introduction

Physical systems are often described by *fields* on (pseudo)-Euclidean spaces. Their equations of motion obey various symmetries, such as isometries $\mathrm{E}(3)$ of Euclidean space $\mathbb{R}^3$ or relativistic Poincaré transformations $\mathrm{E}(1,3)$ of Minkowski spacetime $\mathbb{R}^{1,3}$. PDE solvers should respect these symmetries. In the case of deep learning based surrogates, this property is ensured by making the neural networks *equivariant* (commutative) w.r.t. the transformations of interest.

A fairly general class of equivariant CNNs covering arbitrary spaces and field types is described by the theory of *steerable CNNs* (Weiler et al., 2023). The central result there is that equivariance requires a *"$G$-steerability"* constraint on convolution kernels, where $G = \mathrm{O}(n)$ or $\mathrm{O}(p,q)$ for $\mathrm{E}(n)$- or $\mathrm{E}(p,q)$-equivariant CNNs, respectively. This constraint was solved and implemented for $\mathrm{O}(n)$ (Lang & Weiler, 2021; Cesa et al., 2022), however, $\mathrm{O}(p,q)$-steerable kernels are so far still missing.

This work proposes *Clifford-steerable CNNs* (CS-CNNs), which process *multivector fields* on pseudo-Euclidean spaces $\mathbb{R}^{p,q}$, and are equivariant to the pseudo-Euclidean group $\mathrm{E}(p,q)$: the isometries of $\mathbb{R}^{p,q}$. Multivectors are elements of the Clifford (or geometric) algebra $\mathrm{Cl}(\mathbb{R}^{p,q})$ of $\mathbb{R}^{p,q}$. Neural networks based on Clifford algebras have seen a recent surge in popularity in the field of deep learning and were used to build both non-equivariant (Brandstetter et al., 2023; Ruhe et al., 2023b) and equivariant (Ruhe et al., 2023a; Brehmer et al., 2023) models. While multivectors do not cover all possible field types, e.g. general tensor fields, they include those most relevant in physics. For instance, the Maxwell or Dirac equation and General Relativity can be formulated using the spacetime algebra $\mathrm{Cl}(\mathbb{R}^{1,3})$.

The steerability constraint on convolution kernels is usually either solved analytically or numerically, however, such solutions are not yet known for $\mathrm{O}(p,q)$. Observing that the $G$-steerability constraint is just a $G$-equivariance constraint, Zhdanov et al. (2023) propose to implement $G$-steerable kernels *implicitly* via $G$-equivariant MLPs. Our CS-CNNs follow this approach, implementing implicit $\mathrm{O}(p,q)$-steerable kernels via the $\mathrm{O}(p,q)$-equivariant neural networks for multivectors developed by Ruhe et al. (2023a).

We demonstrate the efficacy of our approach by predicting the evolution of several physical systems. In particular, we consider a fluid dynamics forecasting task on $\mathbb{R}^2$, as well as relativistic electrodynamics simulations on both $\mathbb{R}^3$

[*]Equal contribution [1]AMLab, Informatics Institute, University of Amsterdam [2]AI4Science Lab, Informatics Institute, University of Amsterdam [3]Anton Pannekoek Institute for Astronomy, University of Amsterdam [4]AI4Science, Microsoft Research [5]ELLIS Unit Linz, Institute for Machine Learning, JKU Linz, Austria [6]NXAI GmbH. Correspondence to: Maksim Zhdanov <m.zhdanov@uva.nl>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

and $\mathbb{R}^{1,2}$. CS-CNNs are the first models respecting the full spacetime symmetries of these problems. They significantly outperform competitive baselines, including conventional steerable CNNs and non-equivariant Clifford CNNs. This result remains consistent over dataset sizes. When evaluating the empirical equivariance error of our approach for $\mathrm{E}(2)$ symmetries, we find that we perform on par with the analytical solutions of Weiler & Cesa (2019).

The main contributions of this work are the following:

- While prior work considered only individual multivectors, CS-CNNs process full multivector fields on pseudo-Euclidean spaces or manifolds.
- We investigate the representation theory of $\mathrm{O}(p,q)$-steerable kernels for multivector fields and develop an implicit implementation via $\mathrm{O}(p,q)$-equivariant MLPs.
- The resulting $\mathrm{E}(p,q)$-equivariant CNNs are evaluated on various PDE simulation tasks, where they consistently outperform strong baselines.

This paper is organized as follows: Section 2 introduces the theoretical background underlying our method. CS-CNNs are then developed in Section 3, and empirically evaluated in Section 4. A generalization from flat spaces to general *pseudo-Riemannian manifolds* is presented in Appendix G.

## 2. Theoretical Background

The core contribution of this work is to provide a framework for the construction of *steerable CNNs* for processing *multivector fields* on general *pseudo-Euclidean spaces*. We provide background on pseudo-Euclidean spaces and their symmetries in Section 2.1, on equivariant (steerable) CNNs in Section 2.2, and on multivectors and the Clifford algebra formed by them in Section 2.3.

### 2.1. Pseudo-Euclidean spaces and groups

Conventional Euclidean spaces are metric spaces, i.e. they are equipped with a metric that assigns *positive* distances to any pair of distinct points. Pseudo-Euclidean spaces allow for more general *indefinite metrics*, which relax the positivity requirement on distances. Pseudo-Euclidean spaces appear in our theory in two distinct settings: First, the (affine) base spaces on which feature vector fields are supported, e.g. Minkowski spacetime, are pseudo-Euclidean. Second, the feature vectors attached to each point of spacetime are themselves elements of pseudo-Euclidean vector spaces. We introduce these spaces and their symmetries in the following.

#### 2.1.1. Pseudo-Euclidean vector spaces

**Definition 2.1** (Pseudo-Euclidean vector space)**.** *A* pseudo-Euclidean vector space *(inner product space)* $(V,\eta)$ *of signature* $(p,q)$ *is a* $p+q$*-dimensional vector space* $V$ *over* $\mathbb{R}$ *equipped with an* inner product $\eta$*, which we define as a* non-degenerate[1] *symmetric bilinear form*

$$\eta : V \times V \to \mathbb{R}, \qquad (v_1, v_2) \mapsto \eta(v_1, v_2) \tag{1}$$

*with* $p$ *and* $q$ *positive and negative eigenvalues, respectively.*

If $q=0$, $\eta$ becomes positive-definite, and $(V,\eta)$ is a conventional Euclidean inner product space. For $q \geq 1$, $\eta(v,v)$ can be negative, rendering $(V,\eta)$ pseudo-Euclidean.

Since every inner product space $(V,\eta)$ of signature $(p,q)$ has an orthonormal basis, we can always find a linear isometry with the standard pseudo-Euclidean space $\mathbb{R}^{p,q} \cong (V,\eta)$, to which we mostly will restrict our attention in this paper.

**Definition 2.2** (Standard pseudo-Euclidean vector spaces)**.** *Let* $e_1, \dots, e_{p+q}$ *be the standard basis of* $\mathbb{R}^{p+q}$*. Define an inner product of signature* $(p,q)$

$$\eta^{p,q}(v_1, v_2) := v_1^\top \Delta^{p,q} v_2 \tag{2}$$

*in this basis via its matrix representation*

$$\Delta^{p,q} := \mathrm{diag}(\underbrace{1, \dots, 1}_{p \text{ times}}, \underbrace{-1, \dots, -1}_{q \text{ times}}) . \tag{3}$$

*We call the inner product space* $\mathbb{R}^{p,q} := (\mathbb{R}^{p+q}, \eta^{p,q})$ *the* standard pseudo-Euclidean vector space *of signature* $(p,q)$.

**Example 2.3.** $\mathbb{R}^{3,0} \equiv \mathbb{R}^3$ *recovers the 3-dimensional Euclidean vector space with its standard positive-definite inner product* $\Delta^{3,0} = \mathrm{diag}(1,1,1)$*. The signature* $(p,q) = (1,3)$ *corresponds, instead, to Minkowski spacetime* $\mathbb{R}^{1,3}$ *with Minkowski inner product* $\Delta^{1,3} = \mathrm{diag}(1,-1,-1,-1)$*.*[2]

#### 2.1.2. Pseudo-Euclidean groups

We are interested in neural networks that respect (i.e., commute with, or are *equivariant* to) the symmetries of pseudo-Euclidean spaces, which we define here. For concreteness, we give these definitions for the standard pseudo-Euclidean vector spaces $\mathbb{R}^{p,q}$. Let us start with the two cornerstone groups that define such symmetries:

**Definition 2.4** (Translation groups)**.** *The* translation group $(\mathbb{R}^{p,q}, +)$ *associated with* $\mathbb{R}^{p,q}$ *is formed by its set of vectors and its (canonical) vector addition.*

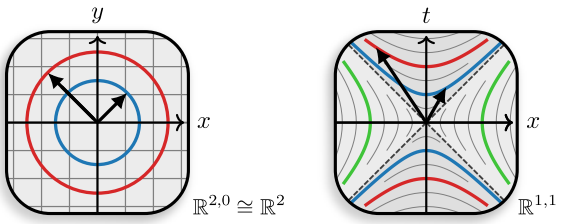



*Figure 2.* Examples of pseudo-Euclidean spaces $\mathbb{R}^{2,0}$ and $\mathbb{R}^{1,1}$. Colors depict $\mathrm{O}(p,q)$-orbits, given by sets of all points $v \in \mathbb{R}^{p,q}$ with the *same squared distance* $\eta^{p,q}(v,v)$ from the origin.

[1] Note that we explicitly refrain from imposing *positive-definiteness* onto the definition of inner product, in order to include typical Minkowski spacetime inner products, etc.

[2] There exist different conventions regarding whether time or space components are assigned the negative sign.

**Definition 2.5** (Pseudo-orthogonal groups)**.** *The* pseudo-orthogonal group $\mathrm{O}(p,q)$ *associated to* $\mathbb{R}^{p,q}$ *is formed by all invertible linear maps that preserve its inner product,*

$$\mathrm{O}(p,q) \;:=\; \big\{g \in \mathrm{GL}(\mathbb{R}^{p,q}) \,\big|\, g^\top \Delta^{p,q} g = \Delta^{p,q}\big\}, \qquad (4)$$

*together with matrix multiplication.* $\mathrm{O}(p,q)$ *is compact for* $p=0$ *or* $q=0$*, and non-compact for mixed signatures.*

**Example 2.6.** *For* $(p,q)=(3,0)$*, we obtain the usual orthogonal group* $\mathrm{O}(3)$*, i.e. rotations and reflections, while* $(p,q)=(1,3)$ *corresponds to the relativistic* Lorentz group $\mathrm{O}(1,3)$*, which also includes boosts between inertial frames.*

Taken together, translations and pseudo-orthogonal transformations of $\mathbb{R}^{p,q}$ form its *pseudo-Euclidean group*, which is the group of all metric preserving symmetries (isometries).[3]

**Definition 2.7** (Pseudo-Euclidean groups)**.** *The* pseudo-Euclidean group *for* $\mathbb{R}^{p,q}$ *is defined as semidirect product*

$$\mathrm{E}(p,q) \;:=\; (\mathbb{R}^{p,q},+) \rtimes \mathrm{O}(p,q) \qquad (5)$$

*with group multiplication defined by* $(\tilde{t},\tilde{g})\cdot(t,g) = (\tilde{t}+\tilde{g}t, \tilde{g}g)$*. Its canonical action on* $\mathbb{R}^{p,q}$ *is given by*

$$\mathrm{E}(p,q) \times \mathbb{R}^{p,q} \to \mathbb{R}^{p,q}, \quad \big((t,g),\,x\big) \mapsto gx+t \qquad (6)$$

**Example 2.8.** *The usual Euclidean group* $\mathrm{E}(3)$ *is reproduced for* $(p,q)=(3,0)$*. For Minkowski spacetime,* $(p,q)=(1,3)$*, we obtain the Poincaré group* $\mathrm{E}(1,3)$*.*

### 2.2. Feature vector fields & Steerable CNNs

Convolutional neural networks operate on spatial signals, formalized as *fields of feature vectors* on a base space $\mathbb{R}^{p,q}$. Transformations of the base space imply corresponding transformations of the feature vector fields defined on them, see Fig. 1 (left column). The specific transformation laws depend thereby on their geometric "field type" (e.g., scalar, vector, or tensor fields). Equivariant CNNs commute with such transformations of feature fields. The theory of *steerable CNNs* shows that this requires a $G$-equivariance constraint on convolution kernels (Weiler et al., 2023). We briefly review the definitions and basic results of feature fields and steerable CNNs in Sections 2.2.1 and 2.2.2 below.

For generality, this section considers topologically closed matrix groups $G \leq \mathrm{GL}(\mathbb{R}^{p,q})$ and affine groups $\mathrm{Aff}(G) = (\mathbb{R}^{p,q},+)\rtimes G$, and allows for any field type. Section 3 will more specifically focus on pseudo-orthogonal groups $G=\mathrm{O}(p,q)$, pseudo-Euclidean groups $\mathrm{Aff}(\mathrm{O}(p,q))=\mathrm{E}(p,q)$, and multivector fields. For a detailed review of Euclidean steerable CNNs and their generalization to Riemannian manifolds we refer to Weiler et al. (2023).

[3] As the translations contained in $\mathrm{E}(p,q)$ move the origin of $\mathbb{R}^{p,q}$, they do not preserve the *vector* space structure of $\mathbb{R}^{p,q}$, but only its structure as *affine* space.

#### 2.2.1. Feature vector fields

Feature vector fields are functions $f:\mathbb{R}^{p,q}\to W$ that assign to each point $x\in\mathbb{R}^{p,q}$ a feature $f(x)$ in some feature vector space $W$. They are additionally equipped with an $\mathrm{Aff}(G)$-action determined by a $G$-representation $\rho$ on $W$.

The specific choice of $(W,\rho)$ fixes the geometric "type" of feature vectors. For instance, $W=\mathbb{R}$ and trivial $\rho(g)=1$ corresponds to *scalars*, $W=\mathbb{R}^{p,q}$ and $\rho(g)=g$ describes *tangent vectors*. Higher order tensor spaces and representations give rise to *tensor fields*. Later on, $W=\mathrm{Cl}(\mathbb{R}^{p,q})$ will be the Clifford algebra and feature vectors will be *multivectors* with a natural $\mathrm{O}(p,q)$-representation $\rho_{\mathrm{Cl}}$.

**Definition 2.9** (Feature vector field)**.** *Consider a pseudo-Euclidean "base space"* $\mathbb{R}^{p,q}$*. Fix any* $G\leq\mathrm{GL}(\mathbb{R}^{p,q})$ *and consider a* $G$*-representation* $(W,\rho)$*, called "field type".*

*Let* $\Gamma(\mathbb{R}^{p,q},W) := \{f:\mathbb{R}^{p,q}\to W\}$ *denote the vector space of* $W$*-feature fields. Define an* $\mathrm{Aff}(G)$*-action*

$$\triangleright_\rho : \mathrm{Aff}(G)\times\Gamma(\mathbb{R}^{p,q},W)\to\Gamma(\mathbb{R}^{p,q},W) \qquad (7)$$

*by setting* $\forall\,(t,g)\in\mathrm{Aff}(G),\; f\in\Gamma(\mathbb{R}^{p,q},W),\; x\in\mathbb{R}^{p,q}$*:*

$$\big[(t,g)\triangleright_\rho f\big](x) := \rho(g)f\big((t,g)^{-1}x\big) = \rho(g)f\big(g^{-1}(x-t)\big).$$

*Since* $\Gamma(\mathbb{R}^{p,q},W)$ *is a vector space and* $\triangleright_\rho$ *is linear, the tuple* $\big(\Gamma(\mathbb{R}^{p,q},W),\triangleright_\rho\big)$ *forms the* $\mathrm{Aff}(G)$*-representation of* feature vector fields *of type* $(W,\rho)$*.*[4]

**Remark 2.10.** *Intuitively,* $(t,g)$ *acts on* $f$ *by*

1. *moving feature vectors across the base space, from points* $g^{-1}(x-t)$ *to new locations* $x$*, and*
2. $G$*-transforming individual feature vectors* $f(x)\in W$ *themselves by means of the* $G$*-representation* $\rho(g)$*.*

Besides the field types mentioned above, equivariant neural networks often rely on *irreducible, regular* or *quotient* representations. More choices of field types are discussed and benchmarked in Weiler & Cesa (2019).

#### 2.2.2. Steerable CNNs

Steerable convolutional neural networks are composed of layers that are $\mathrm{Aff}(G)$-equivariant, that is, which commute with affine group actions on feature fields:

**Definition 2.11** ($\mathrm{Aff}(G)$-equivariance)**.** *Consider any two* $G$*-representations* $(W_{\text{in}},\rho_{\text{in}})$ *and* $(W_{\text{out}},\rho_{\text{out}})$*. Let* $L:\Gamma(\mathbb{R}^{p,q},W_{\text{in}})\to\Gamma(\mathbb{R}^{p,q},W_{\text{out}})$ *be a function ("layer") between the corresponding spaces of feature fields. This layer is said to be* $\mathrm{Aff}(G)$*-equivariant iff it satisfies*

$$L\big((t,g)\triangleright_{\rho_{\text{in}}} f\big) \;=\; (t,g)\triangleright_{\rho_{\text{out}}} L(f) \qquad (8)$$

[4] $\big(\Gamma(\mathbb{R}^{p,q},W),\triangleright_\rho\big)$ is called *induced representation* $\mathrm{Ind}_G^{\mathrm{Aff}(G)}\rho$ (Cohen et al., 2019b). From a differential geometry perspective, it can be viewed as the space of bundle sections of a $G$-associated feature vector bundle; see Defs. G.6, G.7 and (Weiler et al., 2023).

*for any* $(t,g) \in \mathrm{Aff}(G)$ *and any* $f \in \Gamma(\mathbb{R}^{p,q}, W_{\text{in}})$. *Equivalently, the following diagram should commute:*

$$\begin{array}{ccc} \Gamma(\mathbb{R}^{p,q}, W_{\text{in}}) & \xrightarrow{L} & \Gamma(\mathbb{R}^{p,q}, W_{\text{out}}) \\ {\scriptstyle (t,g)\triangleright_{\rho_{\text{in}}}}\downarrow & & \downarrow{\scriptstyle (t,g)\triangleright_{\rho_{\text{out}}}} \\ \Gamma(\mathbb{R}^{p,q}, W_{\text{in}}) & \xrightarrow[L]{} & \Gamma(\mathbb{R}^{p,q}, W_{\text{out}}) \end{array} \tag{9}$$

The most basic operations used in neural networks are parameterized *linear* layers. If one demands *translation equivariance*, these layers are necessarily *convolutions* (see Theorem 3.2.1 in (Weiler et al., 2023)). Similarly, linearity and $\mathrm{Aff}(G)$-equivariance requires *steerable convolutions*, that is, convolutions with $G$-steerable kernels:

**Theorem 2.12** (Steerable convolution)**.** *Consider a layer* $L : \Gamma(\mathbb{R}^{p,q}, W_{\text{in}}) \to \Gamma(\mathbb{R}^{p,q}, W_{\text{out}})$ *mapping between feature fields of types* $(W_{\text{in}}, \rho_{\text{in}})$ *and* $(W_{\text{out}}, \rho_{\text{out}})$*, respectively. If* $L$ *is demanded to be* linear *and* $\mathrm{Aff}(G)$-equivariant*, then:*

1. *$L$ needs to be a* convolution *integral* [5]

$$L\big(f_{\text{in}}\big)(u) = \big[K * f_{\text{in}}\big](u) := \int_{\mathbb{R}^{p,q}} K(v)\big[f_{\text{in}}(u-v)\big]\,dv,$$

*parameterized by a convolution kernel*

$$K : \mathbb{R}^{p,q} \to \mathrm{Hom}_{\text{Vec}}(W_{\text{in}}, W_{\text{out}})\,. \tag{10}$$

*The kernel is* operator-valued *since it aggregates input features in* $W_{\text{in}}$ *linearly into output features in* $W_{\text{out}}$*.* [6][7]

2. *The kernel is required to be* $G$-steerable*, that is, it needs to satisfy the* $G$*-equivariance constraint* [8]

$$\begin{aligned} K(gx) &= \frac{1}{|\det(g)|}\rho_{\text{out}}(g)K(x)\rho_{\text{in}}(g)^{-1} \\ &=: \rho_{\text{Hom}}(g)(K(x)) \end{aligned} \tag{11}$$

*for any* $g \in G$ *and* $x \in \mathbb{R}^{p,q}$*. This constraint is diagrammatically visualized by the commutativity of:*

$$\begin{array}{ccc} \mathbb{R}^{p,q} & \xrightarrow{K} & \mathrm{Hom}_{\text{Vec}}(W_{\text{in}}, W_{\text{out}}) \\ {\scriptstyle g\cdot}\downarrow & & \downarrow{\scriptstyle \rho_{\text{Hom}}(g)} \\ \mathbb{R}^{p,q} & \xrightarrow[K]{} & \mathrm{Hom}_{\text{Vec}}(W_{\text{in}}, W_{\text{out}}) \end{array} \tag{12}$$

*Proof.* See Theorem 4.3.1 in (Weiler et al., 2023). □

[5] $dv$ is the usual Lebesgue measure on $\mathbb{R}^{p+q}$. For the integral to exist, we assume $f$ to be bounded and have compact support.

[6] $\mathrm{Hom}_{\text{Vec}}(W_{\text{in}}, W_{\text{out}})$, the space of vector space homomorphisms, consists of all *linear maps* $W_{\text{in}} \to W_{\text{out}}$. When putting $W_{\text{in}} = \mathbb{R}^{C_{\text{in}}}$ and $W_{\text{out}} = \mathbb{R}^{C_{\text{out}}}$, this space can be identified with the space $\mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$ of $C_{\text{out}} \times C_{\text{in}}$ matrices.

[7] $K : \mathbb{R}^{p,q} \to \mathrm{Hom}_{\text{Vec}}(W_{\text{in}}, W_{\text{out}})$ itself need *not* be linear.

[8] This is in particular *not* demanding $K(v)$ to be (equivariant) homomorphisms of $G$-representations in $\mathrm{Hom}_G(W_{\text{in}}, W_{\text{out}})$, despite $(W_{\text{in}}, \rho_{\text{in}})$ and $(W_{\text{out}}, \rho_{\text{out}})$ being $G$-representations. Only $K$ itself is $G$-equivariant as map $\mathbb{R}^{p,q} \to \mathrm{Hom}_{\text{Vec}}(W_{\text{in}}, W_{\text{out}})$.

**Remark 2.13** (Discretized kernels)**.** *In practice, kernels are often discretized as arrays of shape*

$$\big(X_1, \dots, X_{p+q},\, C_{\text{out}}, C_{\text{in}}\big)$$

*with* $C_{\text{out}} = \dim(W_{\text{out}})$ *and* $C_{\text{in}} = \dim(W_{\text{in}})$*. The first* $p+q$ *axes are indexing a pixel grid on the domain* $\mathbb{R}^{p,q}$*, while the last two axes represent the linear operators in the codomain by* $C_{\text{out}} \times C_{\text{in}}$ *matrices.*

The main takeaway of this section is that *one needs to implement* $G$*-steerable kernels in order to implement* $\mathrm{Aff}(G)$*-equivariant CNNs.* This is a notoriously difficult problem, requiring specialized approaches for different categories of groups $G$ and field types $(W, \rho)$. Unfortunately, the usual approaches do not immediately apply to our goal of implementing $\mathrm{O}(p,q)$-steerable kernels for multivector fields. These include the following cases:

*Analytical*: Most commonly, steerable kernels are parameterized in *analytically derived steerable kernel bases*.[9] Solutions are known for $\mathrm{SO}(3)$ (Weiler et al., 2018a), $\mathrm{O}(3)$ (Geiger et al., 2020) and any $G \leq \mathrm{O}(2)$ (Weiler & Cesa, 2019). Lang & Weiler (2021) and Cesa et al. (2022) generalized this to any *compact* groups $G \leq \mathrm{U}(d)$. However, their solutions still require knowledge of irreps, Clebsch-Gordan coefficients and harmonic basis functions, which need to be derived and implemented for each single group individually. Furthermore, these solutions do not cover pseudo-orthogonal groups $\mathrm{O}(p,q)$ of mixed signature, since these are *non-compact*.

*Regular*: For regular and quotient representations, steerable kernels can be implemented via channel permutations in the matrix dimensions. This is, for instance, done in regular group convolutions (Cohen & Welling, 2016; Weiler et al., 2018b; Bekkers et al., 2018; Cohen et al., 2019a; Finzi et al., 2020). However, these approaches require *finite* $G$ or rely on sampling *compact* $G$, again ruling out general (non-compact) $\mathrm{O}(p,q)$.

*Numerical*: Cohen & Welling (2017) solved the kernel constraint for *finite* $G$ numerically. For $\mathrm{SO}(2)$, Haan et al. (2021) derived numerical solutions based on Lie-algebra representation theory. The numerical routine by Shutty & Wierzynski (2022) solves for Lie-algebra irreps given their structure constants. Corresponding Lie group irreps follow via the matrix exponential, however, only on *connected* groups like the subgroups $\mathrm{SO}^+(p,q)$ of $\mathrm{O}(p,q)$.

*Implicit*: Steerable kernels are merely $G$-equivariant maps between vector spaces $\mathbb{R}^{p,q}$ and $\mathrm{Hom}_{\text{Vec}}(W_{\text{in}}, W_{\text{out}})$. Based on this insight, Zhdanov et al. (2023) parameterize them implicitly via $G$-equivariant MLPs. However, to

[9] Unconstrained kernels, Eq. (10), can be linearly combined, and therefore form a vector space. The steerability constraint, Eq. (11) is *linear*. Steerable kernels span hence a linear subspace and can be parameterized in terms of a *basis* of steerable kernels.

implement these MLPs, one usually requires irreps, irrep endomorphisms and Clebsch-Gordan coefficients for each $G$ of interest.

Our approach presented in Section 3 is based on the implicit kernel parametrization via neural networks by Zhdanov et al. (2023), which requires us to implement $\mathrm{O}(p,q)$-equivariant neural networks. Fortunately, the *Clifford group equivariant neural networks* by Ruhe et al. (2023a) establish $\mathrm{O}(p,q)$-equivariance for the practically relevant case of *Clifford-algebra* representations $\rho_{\mathrm{Cl}}$, i.e., $\mathrm{O}(p,q)$-actions on multivectors. The Clifford algebra, and Clifford group equivariant neural networks, are introduced in the next section.

### 2.3. The Clifford Algebra & Clifford Group Equivariant Neural Networks

This section introduces *multivector features*, a specific type of geometric feature vectors with $\mathrm{O}(p,q)$-action. Multivectors are the elements of a *Clifford algebra* $\mathrm{Cl}(V,\eta)$ corresponding to a pseudo-Euclidean $\mathbb{R}$-vector space $(V,\eta)$. The most relevant properties of Clifford algebras in relation to applications in geometric deep learning are the following:

- $\mathrm{Cl}(V,\eta)$ is, in itself, an $\mathbb{R}$-vector space of dimension $2^d$ with $d := \dim(V) = p+q$. This allows to use multivectors as feature vectors of neural networks (Brandstetter et al., 2023; Ruhe et al., 2023b; Brehmer et al., 2023).
- As an *algebra*, $\mathrm{Cl}(V,\eta)$ comes with an $\mathbb{R}$-bilinear operation
$$\bullet:\ \mathrm{Cl}(V,\eta)\times\mathrm{Cl}(V,\eta)\ \to\ \mathrm{Cl}(V,\eta),$$
called *geometric product*.[10] We can therefore multiply multivectors with each other, which will be a key aspect in various neural network operations.
- $\mathrm{Cl}(V,\eta)$ is furthermore a *representation* space of the pseudo-orthogonal group $\mathrm{O}(V,\eta)$ via $\rho_{\mathrm{Cl}}$, defined in Eq (19) below. This allows to use multivectors as features of $\mathrm{O}(V,\eta)$-*equivariant* networks (Ruhe et al., 2023a).

A formal definition of Clifford algebras can be found in Appendix E. Section 2.3.1 offers a less technical introduction, highlighting basic constructions and results. Sections 2.3.2 and 2.3.3 focus on the natural $\mathrm{O}(p,q)$-action on multivectors, and on Clifford group equivariant neural networks. While we will later mostly be interested in $(V,\eta)=\mathbb{R}^{p,q}$ and $\mathrm{O}(V,\eta)=\mathrm{O}(p,q)$, we keep the discussion here general.

#### 2.3.1. Introduction to the Clifford Algebra

Multivectors are constructed by multiplying and summing vectors. Specifically, $l$ vectors $v_1,\dots,v_l\in V$ multiply to $v_1\bullet\ldots\bullet v_l\in\mathrm{Cl}(V,\eta)$. A general multivector arises as a linear combination of such products,

$$x=\sum\nolimits_{i\in I} c_i\cdot v_{i,1}\bullet\cdots\bullet v_{i,l_i}\,, \tag{13}$$

with some finite index set $I$ and $v_{i,k}\in V$ and $c_i\in\mathbb{R}$.

The main algebraic property of the Clifford algebra is that it relates the geometric product of vectors $v\in V$ to the inner product $\eta$ on $V$ by requiring:

$$v\bullet v\ \overset{!}{=}\ \eta(v,v)\cdot 1_{\mathrm{Cl}(V,\eta)}\qquad\forall\, v\in V\subset\mathrm{Cl}(V,\eta) \tag{14}$$

Intuitively, this means that the product of a vector with itself collapses to a scalar value $\eta(v,v)\in\mathbb{R}\subseteq\mathrm{Cl}(V,\eta)$, from which all other properties of the algebra follow by bilinearity. This leads in particular to the *fundamental relation*[11]:

$$v_2\bullet v_1=-v_1\bullet v_2\ +\ 2\eta(v_1,v_2)\cdot 1_{\mathrm{Cl}(V,\eta)}\quad\forall\, v_1,v_2\in V.$$

For the standard orthonormal basis $[e_1,\dots,e_{p+q}]$ of $\mathbb{R}^{p,q}$ this reduces to the following simple rules:

$$e_i\bullet e_j\ =\ \begin{cases} -e_j\bullet e_i & \text{for } i\neq j \quad (15a)\\ \eta(e_i,e_i)=+1 & \text{for } i=j\leq p \quad (15b)\\ \eta(e_i,e_i)=-1 & \text{for } i=j>p \quad (15c)\end{cases}$$

An (orthonormal) basis of $\mathrm{Cl}(V,\eta)$ is constructed by repeatedly taking geometric products of any basis vectors $e_i\in V$. Note that, *up to sign flip*, (1) the *ordering* of elements in any product is *irrelevant* due to Eq. (15a), and (2) any *elements occurring twice cancel out* due to Eqs. (15b,15c).

The basis elements constructed this way can be identified with (and labeled by) subsets $A\subseteq[d]:=\{1,\dots,d\}$, where the presence or absence of an index $i\in A$ signifies whether the corresponding $e_i$ appears in the product. Agreeing furthermore on an ordering to disambiguate signs, we define

$$e_A:=e_{i_1}\bullet e_{i_2}\bullet\ldots\bullet e_{i_k}\quad\text{for}\quad A=\{i_1<\cdots<i_k\}\neq\varnothing$$

and $e_\varnothing:=1_{\mathrm{Cl}(V,\eta)}$. From this, it is clear that $\dim\mathrm{Cl}(V,\eta)=2^d$. Table 1 gives a specific example for $(V,\eta)=\mathbb{R}^{1,2}$.

| name | grade $k$ | dim $\binom{d}{k}$ | basis $k$-vectors | norm |
|---|---|---|---|---|
| scalar | 0 | 1 | 1 | $+1$ |
| vector | 1 | 3 | $e_1$ | $+1$ |
| | | | $e_2,\ e_3$ | $-1$ |
| pseudovector | 2 | 3 | $e_{12},\ e_{13}$ | $-1$ |
| | | | $e_{23}$ | $+1$ |
| pseudoscalar | 3 | 1 | $e_{123}$ | $+1$ |

*Table 1.* Orthonormal basis for $\mathrm{Cl}(\mathbb{R}^{p,q})$ with $(p,q)=(1,2)$. "Norm" refers to $\bar\eta(e_A,e_A)=\eta_A$; see Eq. (18).

Any multivector $x\in\mathrm{Cl}(V,\eta)$ can be uniquely expanded in this basis,

$$x\ =\ \sum\nolimits_{A\subseteq[d]} x_A\cdot e_A, \tag{16}$$

where $x_A\in\mathbb{R}$ are coefficients.

[10] The geometric product is *unital, associative, non-commutative*, and $\mathrm{O}(V,\eta)$-*equivariant*. Its main defining property is highlighted in Eq. (14). A proper definition is given in Definition E.2, Eq. (73).

[11] To see this, use $v:=v_1+v_2$ in Eq. (14) and expand.

Note that there are $\binom{d}{k}$ basis elements $e_A$ of *"grade"* $|A|=k$, i.e., which are composed from $k$ out of the $d$ distinct $e_i \in V$. These span $d+1$ linear subspaces $\mathrm{Cl}^{(k)}(V,\eta)$, the elements of which are called $k$*-vectors*. They include scalars ($k=0$), vectors ($k=1$), bivectors ($k=2$), etc. The full Clifford algebra decomposes thus into a direct sum over grades:

$$\mathrm{Cl}(V,\eta) = \bigoplus_{k=0}^{d} \mathrm{Cl}^{(k)}(V,\eta), \quad \dim \mathrm{Cl}^{(k)}(V,\eta) = \binom{d}{k}.$$

Given any multivector $x$, expanded as in Eq. (16), we can define its $k$-th *grade projection* on $\mathrm{Cl}^{(k)}(V,\eta)$ as:

$$x^{(k)} = \sum\nolimits_{A\subseteq[d],\,|A|=k} x_A \cdot e_A. \tag{17}$$

Finally, the inner product $\eta$ on $V$ is naturally extended to $\mathrm{Cl}(V,\eta)$ by defining $\bar{\eta} : \mathrm{Cl}(V,\eta)\times\mathrm{Cl}(V,\eta)\to\mathbb{R}$ as

$$\bar{\eta}(x,y) := \sum\nolimits_{A\subseteq[d]} \eta_A \cdot x_A \cdot y_A, \tag{18}$$

where $\eta_A := \prod_{i\in A}\eta(e_i,e_i) \in \{\pm 1\}$ are sign factors. The tuple $(e_A)_{A\subseteq[d]}$ is an orthonormal basis of $\mathrm{Cl}(V,\eta)$ w.r.t. $\bar{\eta}$.

All of these constructions and statements are more formally defined and proven in the appendix of (Ruhe et al., 2023b).

### 2.3.2. Clifford grades as $\mathrm{O}(p,q)$-representations

The individual grades $\mathrm{Cl}^{(k)}(V,\eta)$ turn out to be representation spaces of the (abstract) pseudo-orthogonal group

$$\mathrm{O}(V,\eta) := \big\{ g\in \mathrm{GL}(V) \,\big|\, \forall v\in V : \eta(gv,gv)=\eta(v,v)\big\}, \tag{19}$$

which coincides for $(V,\eta)=\mathbb{R}^{p,q}$ with $\mathrm{O}(p,q)$ in Def. 2.2. $\mathrm{O}(V,\eta)$ acts thereby on multivectors by *individually multiplying each* 1*-vector* from which they are constructed *with* $g$.

**Definition/Theorem 2.14** ($\mathrm{O}(V,\eta)$-action on $\mathrm{Cl}(V,\eta)$)**.** *Let $(V,\eta)$ be a pseudo-Euclidean space, $g, g_i \in \mathrm{O}(V,\eta)$, $c_i\in\mathbb{R}$, $v_{i,j}\in V$, $x, x_i \in \mathrm{Cl}(V,\eta)$, and $I$ a finite index set. Define the orthogonal algebra representation*

$$\rho_{\mathrm{Cl}} : \mathrm{O}(V,\eta) \to \mathrm{O}_{\mathrm{Alg}}\left(\mathrm{Cl}(V,\eta),\bar{\eta}\right)^{12} \tag{20}$$

*of $\mathrm{O}(V,\eta)$ via the canonical $\mathrm{O}(V,\eta)$-action on each of the contained* 1*-vectors:*

$$\rho_{\mathrm{Cl}}(g)\Big(\sum\nolimits_{i\in I} c_i\cdot v_{i1}\bullet\ldots\bullet v_{ij_i}\Big) := \sum\nolimits_{i\in I} c_i\cdot (gv_{i1})\bullet\ldots\bullet(gv_{ij_i}). \tag{21}$$

*$\rho_{\mathrm{Cl}}$ is well-defined as an* orthogonal representation:

*linear:* $\rho_{\mathrm{Cl}}(g)(c_1\cdot x_1 + c_2\cdot x_2) = c_1\cdot\rho_{\mathrm{Cl}}(g)(x_1) + c_2\cdot\rho_{\mathrm{Cl}}(g)(x_2)$

*composing:* $\rho_{\mathrm{Cl}}(g_2)\left(\rho_{\mathrm{Cl}}(g_1)(x)\right) = \rho_{\mathrm{Cl}}(g_2g_1)(x)$

*invertible:* $\rho_{\mathrm{Cl}}(g)^{-1}(x) = \rho_{\mathrm{Cl}}(g^{-1})(x)$,

*orthogonal:* $\bar{\eta}(\rho_{\mathrm{Cl}}(g)(x_1),\rho_{\mathrm{Cl}}(g)(x_2)) = \bar{\eta}(x_1,x_2)$

*Moreover, the geometric product is* $\mathrm{O}(V,\eta)$-equivariant*, making $\rho_{\mathrm{Cl}}$ an (orthogonal)* algebra representation:

$$\rho_{\mathrm{Cl}}(g)(x_1)\bullet\rho_{\mathrm{Cl}}(g)(x_2) = \rho_{\mathrm{Cl}}(g)(x_1\bullet x_2). \tag{22}$$

$$\begin{array}{ccc} \mathrm{Cl}(V,\eta)\times\mathrm{Cl}(V,\eta) & \xrightarrow{\ \bullet\ } & \mathrm{Cl}(V,\eta) \\ \Big\downarrow{\scriptstyle \rho_{\mathrm{Cl}}(g)\times\rho_{\mathrm{Cl}}(g)} & & \Big\downarrow{\scriptstyle \rho_{\mathrm{Cl}}(g)} \\ \mathrm{Cl}(V,\eta)\times\mathrm{Cl}(V,\eta) & \xrightarrow{\ \bullet\ } & \mathrm{Cl}(V,\eta) \end{array} \tag{23}$$

This representation $\rho_{\mathrm{Cl}}$ reduces furthermore to independent sub-representations on individual $k$-vectors.

**Theorem 2.15** ($\mathrm{O}(V,\eta)$-action on grades $\mathrm{Cl}^{(k)}(V,\eta)$)**.** *Let $g\in\mathrm{O}(V,\eta)$, $x\in\mathrm{Cl}(V,\eta)$ and $k\in 0,\ldots,d$ a grade.*

*The grade projection $(\,\cdot\,)^{(k)}$ is $\mathrm{O}(V,\eta)$-equivariant:*

$$\big(\rho_{\mathrm{Cl}}(g)\,x\big)^{(k)} = \rho_{\mathrm{Cl}}(g)\big(x^{(k)}\big) \tag{24}$$

$$\begin{array}{ccc} \mathrm{Cl}(V,\eta) & \xrightarrow{\ (\cdot)^{(k)}\ } & \mathrm{Cl}^{(k)}(V,\eta) \\ \Big\downarrow{\scriptstyle \rho_{\mathrm{Cl}}(g)} & & \Big\downarrow{\scriptstyle \rho_{\mathrm{Cl}}(g)} \\ \mathrm{Cl}(V,\eta) & \xrightarrow[\ (\cdot)^{(k)}\ ]{} & \mathrm{Cl}^{(k)}(V,\eta) \end{array} \tag{25}$$

*This implies in particular that $\mathrm{Cl}(V,\eta)$ is reducible to sub-representations $\mathrm{Cl}^{(k)}(V,\eta)$, i.e. $\rho_{\mathrm{Cl}}(g)$ does not mix grades.*

*Proof.* Both theorems are proven in (Ruhe et al., 2023a). □

### 2.3.3. $\mathrm{O}(p,q)$-equivariant Clifford Neural Nets

Based on those properties, Ruhe et al. (2023a) proposed *Clifford group equivariant neural networks* (CGENNs). Due to a group isomorphism, this is equivalent to the network's $\mathrm{O}(V,\eta)$-equivariance.

**Definition/Theorem 2.16** (Clifford Group Equivariant NN)**.** *Consider a grade $k=0,\ldots,d$ and weights $w^k_{mn}\in\mathbb{R}$. A* Clifford group equivariant neural network *(CGENN) is constructed from the following functions, operating on one or more multivectors $x_i\in\mathrm{Cl}(V,\eta)$.*

Linear layers: *mix $k$-vectors. For each $1\le m\le c_{\mathrm{out}}$:*

$$L^{(k)}_m(x_1,\ldots,x_{c_{\mathrm{in}}}) := \sum\nolimits_{n=1}^{c_{\mathrm{in}}} w^k_{mn}\cdot x^{(k)}_n \tag{26}$$

*Such weighted linear mixing within sub-representations $\mathrm{Cl}^{(k)}(V,\eta)$ is common in equivariant MLPs.*

Geometric product layers: *compute weighted geometric products with grade-dependent weights:*

$$P^{(k)}(x_1,x_2) := \sum\nolimits_{m=0}^{d}\sum\nolimits_{n=0}^{d} w^k_{mn}\cdot\big(x^{(m)}_1\bullet x^{(n)}_2\big)^{(k)} \tag{27}$$

[12] $\mathrm{O}_{\mathrm{Alg}}\big(\mathrm{Cl}(V,\eta),\bar{\eta}\big)$ is the group of all linear orthogonal transformations of $\mathrm{Cl}(V,\eta)$ that are also multiplicative w.r.t. $\bullet$.

*This is similar to the irrep-feature tensor products in MACE (Batatia et al., 2022).*

Nonlinearity: *As activations, we use* $A(x) := x \cdot \Phi\big(x^{(0)}\big)$ *where* $\Phi$ *is the CDF of the Gaussian distribution. This is inspired by* GatedGELU *from Brehmer et al. (2023).*

*All of these operations are by Theorems 2.14 and 2.15* $\mathrm{O}(V,\eta)$*-equivariant.*

## 3. Clifford-Steerable CNNs

This section presents *Clifford-Steerable Convolutional Neural Networks* (CS-CNNs), which operate on multivector fields on $\mathbb{R}^{p,q}$, and are equivariant to the isometry group $\mathrm{E}(p,q)$ of $\mathbb{R}^{p,q}$. To achieve $\mathrm{E}(p,q)$-equivariance, we need to find a way to implement $\mathrm{O}(p,q)$-steerable kernels (Section 2.2), which we do by leveraging the connection between $\mathrm{Cl}(\mathbb{R}^{p,q})$ and $\mathrm{O}(p,q)$ presented in Section 2.3.

CS-CNNs process (multi-channel) *multivector fields*

$$f : \mathbb{R}^{p,q} \to \mathrm{Cl}(\mathbb{R}^{p,q})^c \tag{28}$$

of type $(W,\rho) = (\mathrm{Cl}(\mathbb{R}^{p,q})^c, \rho_{\mathrm{Cl}}^c)$ with $c \geq 1$ channels. The representation

$$\rho_{\mathrm{Cl}}^c = \textstyle\bigoplus_{i=1}^c \rho_{\mathrm{Cl}} : \mathrm{O}(p,q) \to \mathrm{GL}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^c\big) \tag{29}$$

is given by the action $\rho_{\mathrm{Cl}}$ from Definition/Theorem 2.14, however, applied to each of the $c$ components individually.

Following Theorem 2.12, our main goal is the construction of a convolution operator

$$L : \Gamma\big(\mathbb{R}^{p,q}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}\big) \to \Gamma\big(\mathbb{R}^{p,q}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}\big),$$
$$L(f_{\mathrm{in}})(u) := \int_{\mathbb{R}^{p,q}} K(v)\big[f_{\mathrm{in}}(u-v)\big]\, dv, \tag{30}$$

parameterized by a convolution kernel

$$K : \mathbb{R}^{p,q} \to \mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}\big) \tag{31}$$

that satisfies the following $\mathrm{O}(p,q)$*-steerability (equivariance) constraint* for every $g \in \mathrm{O}(p,q)$ and $v \in \mathbb{R}^{p,q}$.[13] (32)

$$K(gv) \overset{!}{=} \rho_{\mathrm{Cl}}^{c_{\mathrm{out}}}(g)\, K(v)\, \rho_{\mathrm{Cl}}^{c_{\mathrm{in}}}(g^{-1}) =: \rho_{\mathrm{Hom}}(g)(K(v)),$$

As mentioned in Section 2.2.2, constructing such $\mathrm{O}(p,q)$-steerable kernels is typically difficult. To overcome this challenge, we follow Zhdanov et al. (2023) and implement the kernels *implicitly*. Specifically, they are based on $\mathrm{O}(p,q)$-equivariant *"kernel networks"*[14]

$$\mathcal{K} : \mathbb{R}^{p,q} \to \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}}, \tag{33}$$

implemented as CGENNs (Section 2.3.3).

Unfortunately, the codomain of $\mathcal{K}$ is $\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}}$ instead of $\mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}\big)$, as required by steerable kernels, Eq. (31). To bridge the gap between these spaces, we introduce an $\mathrm{O}(p,q)$-equivariant linear layer, called *kernel head* $H$. Its purpose is to transform the kernel network's output $\cancel{k} := \mathcal{K}(v) \in \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}}$ into the desired $\mathbb{R}$-linear map between multivector channels $H(\cancel{k}) \in \mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}\big)$. The relation between kernel network $\mathcal{K}$, kernel head $H$, and the resulting steerable kernel $K := H \circ \mathcal{K}$ is visualized in Fig. 3 (right).

To achieve $\mathrm{O}(p,q)$-equivariance (steerability) of $K = H \circ \mathcal{K}$, we have to make the kernel head $H$ of a specific form:

**Definition 3.1** (Kernel head)**.** *A* kernel head *is a map*

$$H\colon \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}} \to \mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}\big)$$
$$\cancel{k} \mapsto H(\cancel{k}), \tag{34}$$

*where the* $\mathbb{R}$*-linear operator*

$$H(\cancel{k}) : \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}} \to \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}, \quad \mathfrak{f} \mapsto H(\cancel{k})[\mathfrak{f}],$$

*is defined on each output channel* $i \in [c_{\mathrm{out}}]$ *and grade component* $k = 0, \dots, d$*, by:* (35)

$$H(\cancel{k})[\mathfrak{f}]_i^{(k)} := \sum_{\substack{j \in [c_{\mathrm{in}}] \\ m,n=0,\dots,d}} w_{mn,ij}^k \cdot \Big(\cancel{k}_{ij}^{(m)} \bullet \mathfrak{f}_j^{(n)}\Big)^{(k)}$$

$m, n = 0, \dots, d$ *label grades and* $j \in [c_{\mathrm{in}}]$ *input channels. The* $w_{mn,ij}^k \in \mathbb{R}$ *are parameters that allow for weighted mixing between grades and channels.*

Our implementation of the kernel head is discussed in Appendix A.5. Note that the kernel head $H$ can be seen as a linear combination of partially evaluated geometric product layers $P^{(k)}(\cancel{k}_{ij}, \cdot)$ from (27), which mixes input channels to get the output channels. The specific form of the kernel head $H$ comes from the following, most important property:

**Proposition 3.2** (Equivariance of the kernel head)**.** *The* kernel head $H$ *is* $\mathrm{O}(p,q)$*-equivariant w.r.t.* $\rho_{\mathrm{Cl}}^{c_{\mathrm{out}} \times c_{\mathrm{in}}}$ *and* $\rho_{\mathrm{Hom}}$*, i.e. for* $g \in \mathrm{O}(p,q)$ *and* $\cancel{k} \in \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}}$ *we have:*

$$H\big(\rho_{\mathrm{Cl}}^{c_{\mathrm{out}} \times c_{\mathrm{in}}}(g)(\cancel{k})\big) = \rho_{\mathrm{Hom}}(g)(H(\cancel{k})). \tag{36}$$

*Proof.* The proof relies on the $\mathrm{O}(p,q)$-equivariance of the geometric product and of linear combinations within grades. It can be found in the Appendix in Proof F.1. □

With these obstructions out of the way, we can now give the core definition of this paper:

**Definition 3.3** (Clifford-steerable kernel)**.** *A* Clifford-steerable kernel $K$ *is a map as in Eq. (31) that factorizes as:* $K = H \circ \mathcal{K}$ *with a* kernel head $H$ *from Eq. (35) and a* kernel network $\mathcal{K}$ *given by a Clifford group equivariant neural network (CGENN)*[15] *from Definition/Theorem 2.16:*

$$\mathcal{K} = [\mathcal{K}_{ij}]_{\substack{i \in [c_{\mathrm{out}}] \\ j \in [c_{\mathrm{in}}]}} : \mathbb{R}^{p,q} \to \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}}. \tag{37}$$

[13] The volume factor $|\det g| = 1$ drops out for $g \in \mathrm{O}(p,q)$.

[14] The kernel network's output $\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \cdot c_{\mathrm{in}}}$ is here reshaped to matrix form $\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}}$.

[15] More generally we could employ any $\mathrm{O}(p,q)$-equivariant neural network $\mathcal{K}$ w.r.t. the standard action $\rho(g) = g$ and $\rho_{\mathrm{Cl}}^{c_{\mathrm{out}} \times c_{\mathrm{in}}}$.

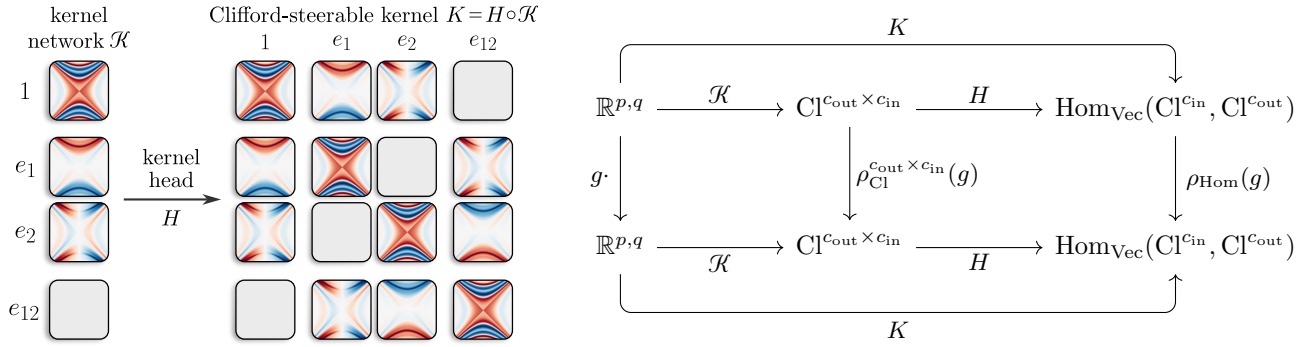



*Figure 3. Left:* Multi-vector valued output of the kernel-network $\mathcal{K}$ for $c_{\mathrm{in}} = c_{\mathrm{out}} = 1$, $(p,q) = (1,1)$, and its expansion to a full $\mathrm{O}(1,1)$-steerable kernel via the kernel head $H$. *Right:* Commutative diagram of the construction and $\mathrm{O}(p,q)$-equivariance of implicit steerable kernels $K = H \circ \mathcal{K}$, composed from a kernel network $\mathcal{K}$ with $c_{\mathrm{out}} \times c_{\mathrm{in}}$ multivector outputs and the kernel head $H$. The two inner squares show the individual equivariance of $\mathcal{K}$ and $H$, from which the kernels' overall equivariance follows. We abbreviate $\mathrm{Cl}(\mathbb{R}^{p,q})$ by Cl.

The main theoretical result of this paper is that Clifford-steerable kernels are always $\mathrm{O}(p,q)$-steerable:

**Theorem 3.4** (Equivariance of Clifford-steerable kernels)**.** *Every* Clifford-steerable kernel $K = H \circ \mathcal{K}$ *is* $\mathrm{O}(p,q)$*-steerable w.r.t. the standard action* $\rho(g) = g$ *and* $\rho_{\mathrm{Hom}}$*:*

$$K(gv) \;=\; \rho_{\mathrm{Hom}}(g)(K(v)) \qquad \forall\, g \in \mathrm{O}(p,q),\; v \in \mathbb{R}^{p,q}$$

*Proof.* $\mathcal{K}$ and $H$ are $\mathrm{O}(p,q)$-equivariant by Definition/Theorem 2.16 and Proposition 3.2, respectively. The $\mathrm{O}(p,q)$-equivariance of the composition $K = H \circ \mathcal{K}$ then follows from Fig. 3 or by direct calculation:

$$\begin{aligned} K(gv) \;&=\; H\big(\mathcal{K}(gv)\big) \\ &=\; H\big(\rho_{\mathrm{Cl}}^{c_{\mathrm{out}} \times c_{\mathrm{in}}}(g)(\mathcal{K}(v))\big) \\ &=\; \rho_{\mathrm{Hom}}(g)\,\big(H\big(\mathcal{K}(v)\big)\big) \\ &=\; \rho_{\mathrm{Hom}}(g)\big(K(v)\big). \end{aligned} \tag{38}$$

□

A direct Corollary of Theorem 3.4 and Theorem 2.12 is:

**Corollary 3.5.** *Let* $K = H \circ \mathcal{K}$ *be a* Clifford-steerable kernel. *The corresponding convolution operator* $L$ *(Eq.* (30)*) is then* $\mathrm{E}(p,q)$*-equivariant, i.e.* $\forall\, f_{\mathrm{in}} \in \Gamma\left(\mathbb{R}^{p,q}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}\right)$*:*

$$(t,g) \triangleright L(f_{\mathrm{in}}) = L\big((t,g) \triangleright f_{\mathrm{in}}\big) \qquad \forall\, (t,g) \in \mathrm{E}(p,q)$$

**Definition 3.6** (Clifford-steerable CNN)**.** *We call a convolutional network (that operates on multivector fields and is) based on Clifford-steerable kernels a* Clifford-Steerable Convolutional Neural Network *(CS-CNN).*

**Remark 3.7.** *Brandstetter et al. (2023) use a similar kernel head* $H$ *as ours, Eq.* (35)*. However, their kernel network* $\mathcal{K}$ *is not* $\mathrm{O}(p,q)$*-equivariant, making their overall architecture merely translation- but not* $\mathrm{E}(p,q)$*-equivariant.*

**Remark 3.8.** *The vast majority of parameters of CS-CNNs reside in their kernel networks* $\mathcal{K}$*. Further parameters are found in the kernel heads' weighted geometric product operation and summation of steerable biases to scalar grades.*

**Remark 3.9.** *While CS-CNNs are formalized in* continuous space*, they are in practice typically applied to* discretized fields*. Our implementation allows for any sampling points, thus covering both* pixel grids *and* point clouds*.*

Appendix G generalizes CS-CNNs from flat spacetimes to general curved *pseudo-Riemannian manifolds*. Appendix A provides details on our implementation of CS-CNNs, available at `https://github.com/maxxxzdn/clifford-group-equivariant-cnns`.

## 4. Experimental Results

To assess CS-CNNs, we investigate how well they can learn to simulate dynamical systems by testing their ability to predict future states given a history of recent states (Gupta & Brandstetter, 2022). We consider three tasks:

(1) Fluid dynamics on $\mathbb{R}^2$ (incompressible Navier-Stokes)
(2) Electrodynamics on $\mathbb{R}^3$ (Maxwell's Eqs.)
(3) Electrodynamics on $\mathbb{R}^{1,2}$ (Maxwell's Eqs., relativistic)

Only the last setting is properly incorporating time into $1+2$-dimensional *spacetime*, while the former two are treating time steps improperly as *feature channels*. The improper setting allows us to compare our method with prior work, which was not able to incorporate the full spacetime symmetries $\mathrm{E}(1,n)$, but only the spatial subgroup $\mathrm{E}(n)$ (which is also covered by CS-CNNs).

**Data & Tasks:** For both tasks (1) and (2), the goal is to predict the next state given previous 4 time steps. In (1), the inputs are scalar pressure and vector velocity fields. In (2) the inputs are vector electric and bivector magnetic fields. For task (3), the goal is to predict 16 future states given the previous 16 time steps. In this case, the *entire electromagnetic field forms a bivector* (Orbán & Mira, 2021). More details on the datasets are found in Appendix D.3.

**Architectures:** We evaluate six network architectures:

| architecture | matrix group $G$ | isometry group |
|---|---|---|
| Conventional ResNet | $\{e\}$ | translations |
| Clifford ResNet | $\{e\}$ | translations |
| Fourier Neural Operators | $\{e\}$ | translations |
| $G$-Fourier Neural Operators | $\mathrm{D}_4 < \mathrm{O}(2)$ | $\approx \mathrm{E}(2)$ |
| Steerable ResNet | $\mathrm{O}(n)$ | $\mathrm{E}(n)$ |
| Clifford-Steerable ResNet | $\mathrm{O}(p,q)$ | $\mathrm{E}(p,q)$ |

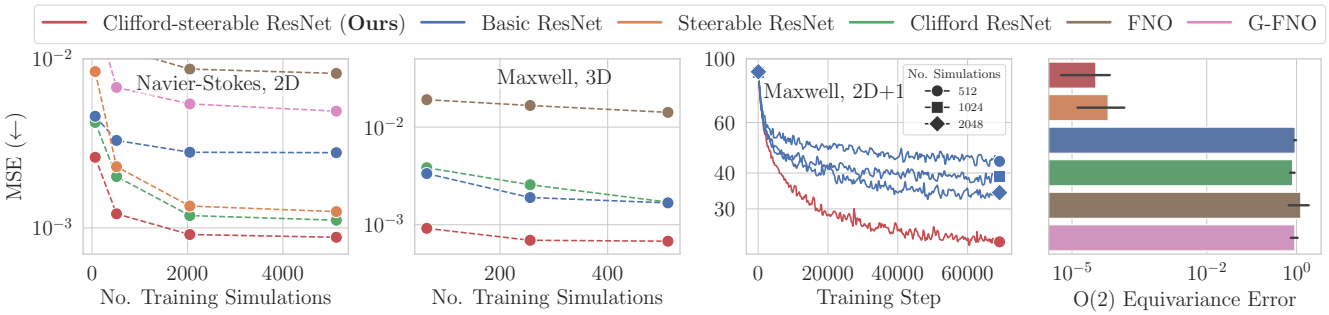



*Figure 4. Plots 1 & 2:* Mean squared errors (MSEs) on the Navier-Stokes 2D and Maxwell 3D forecasting tasks (one-step loss) as a function of number of training simulations. *Plot 3:* Convergence (test loss) of our model vs. a basic ResNet on the relativistic Maxwell task. *Plot 4:* Relative $\mathrm{O}(2)$-equivariance errors of different models. $G$-FNOs fail as they cannot correctly ingest multivector data.

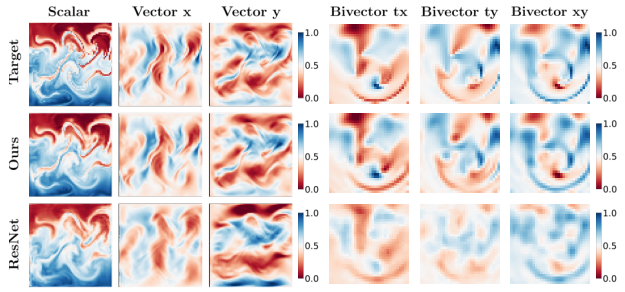



*Figure 5.* Visual comparison of target and predicted fields. *Left*: Our CS-ResNet clearly produces better results than the basic ResNet on Navier Stokes, despite only being trained on 64 instead of 5120 simulations. *Right*: On Maxwell 2D+1, CS-ResNets capture crisp details like wavefronts more accurately.

The basic ResNet model is described in Apx. D. Clifford, Steerable, and our CS-ResNets are variations of it that substitute vanilla convolutions with their Clifford (Brandstetter et al., 2023), $\mathrm{O}(n)$-steerable (Weiler & Cesa, 2019; Cesa et al., 2022), and Clifford-Steerable counterparts, respectively. We also test Fourier Neural Operators (FNO) (Li et al., 2021) and $G$-FNO (Helwig et al., 2023). The latter add equivariance to the Dihedral group $\mathrm{D}_4 < \mathrm{O}(2)$. Assuming scalar or regular representations, they are incapable of digesting multivector-valued data. We address this by replacing the initial lifting and final projection with *unconstrained* operations that are able to learn a geometrically correct mapping from/to multivectors. All models scale their number of channels to match the parameter count of the basic ResNet.

**Results:** To evaluate the models, we report mean-squared error losses (MSE) on test sets. As shown in Fig. 4, our *CS-ResNets outperform all baselines on all tasks*, especially when modeling Maxwell's equations. CS-ResNets are extremely sample-efficient: for the Navier-Stokes experiment, they require only 64 training samples to outperform the basic ResNet and FNOs *trained on 80× more data*.

Plot 1 proves CS-CNNs to be a good alternative to classical $\mathrm{O}(2)$-steerable CNNs in the nonrelativistic case. We didn't run $\mathrm{O}(3)$-steerable CNNs on Maxwell 3D due to resource constraints and on 2D+1 as they are not Lorentz-equivariant. $G$-FNO does not support either of these symmetries.

The Maxwell data on spacetime $\mathbb{R}^{1,2}$ is naturally modeled by *space-time algebra* $\mathrm{Cl}(\mathbb{R}^{1,2})$ (Hestenes, 2015). Contrary to tasks (1) and (2), time appears here as a proper grid dimension, not as a feature channel. The light cone structure of CS-CNN kernels (Fig. 3) ensures the models' consistency across different inertial frames of reference. This is relevant as the simulated electromagnetic fields are induced by particles moving at relativistic velocities. We see in Plot 3 that CS-CNNs converge significantly faster and are more sample efficient than basic ResNets.

**Equivariance error:** To assess the models' $\mathrm{E}(2)$-equivariance, we measure the relative error $\frac{|f(g.x)-g.f(x)|}{|f(g.x)+g.f(x)|}$ between (1) the output computed from a transformed input; and (2) the transformed output, given the original input. As shown in Fig. 4 (right), both steerable models are equivariant up to numerical artefacts. Despite training, the other models did not become equivariant at all. This holds in particular for $G$-FNO, which covers only a subgroup of discrete rotations.

## 5. Conclusions

We presented Clifford-Steerable CNNs, a new theoretical framework for $\mathrm{E}(p,q)$-equivariant convolutions on pseudo-Euclidean spaces such as Minkowski-spacetime. CS-CNNs process fields of multivectors – geometric features which naturally occur in many areas of physics. The required $\mathrm{O}(p,q)$-steerable convolution kernels are implemented implicitly via Clifford group equivariant neural networks. This makes so far unknown analytic solutions for the steerability constraint unnecessary. CS-CNNs significantly outperform baselines on a variety of physical dynamics tasks. Some limitations of CS-CNNs are discussed in Appendix B.

CS-CNNs are, to the best of our knowledge, the first convolutional networks respecting the full symmetries $\mathrm{E}(p,q)$ of pseudo-Euclidean spaces. They are readily extended to general *pseudo-Riemannian manifolds*; see Apx. G and (Weiler et al., 2023). They could furthermore be adapted to *steerable partial differential operators* (Jenner & Weiler, 2022), connecting them to *multivector calculus* (Hestenes, 1968; Hitzer, 2002; Lasenby et al., 1993).

## Impact Statement

The broader implications of our work are primarily in the improved modeling of PDEs, other physical systems, or multi-vector based applications in computational geometry. Being able to model such systems more *accurately* can lead to better understanding about the physical systems governing our world, while being able to model such systems more *efficiently* could greatly improve the ecological footprint of training ML models for modeling physical systems.

## Acknowledgements

This research was supported by Microsoft Research AI4Science. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers/sponsors.

## References

Batatia, I., Kovács, D. P., Simm, G. N. C., Ortner, C., and Csányi, G. Mace: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Bekkers, E. B-spline CNNs on Lie groups. *International Conference on Learning Representations (ICLR)*, 2020.

Bekkers, E. J., Lafarge, M. W., Veta, M., Eppenhof, K. A. J., Pluim, J. P. W., and Duits, R. Roto-Translation Covariant Convolutional Networks for Medical Image Analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2018.

Brandstetter, J., Berg, R. v. d., Welling, M., and Gupta, J. K. Clifford Neural Layers for PDE Modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Brehmer, J., Haan, P. d., Behrends, S., and Cohen, T. S. Geometric Algebra Transformer. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Cesa, G., Lang, L., and Weiler, M. A Program to Build E(N)-Equivariant Steerable CNNs. In *International Conference on Learning Representations (ICLR)*, 2022.

Cohen, T. and Welling, M. Group Equivariant Convolutional Networks. In *International Conference on Machine Learning (ICML)*, pp. 2990–2999, 2016.

Cohen, T., Weiler, M., Kicanaoglu, B., and Welling, M. Gauge Equivariant Convolutional Networks and the Icosahedral CNN. In *International Conference on Machine Learning (ICML)*, pp. 1321–1330, 2019a.

Cohen, T. S. and Welling, M. Steerable CNNs. In *International Conference on Learning Representations (ICLR)*, 2017.

Cohen, T. S., Geiger, M., and Weiler, M. A General Theory of Equivariant CNNs on Homogeneous Spaces. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019b.

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

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing Convolutional Neural Networks for Equivariance to Lie Groups on Arbitrary Continuous Data. In *International Conference on Machine Learning (ICML)*, pp. 3165–3176, 2020.

Finzi, M., Welling, M., and Wilson, A. G. A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups. In *International Conference on Machine Learning (ICML)*, 2021.

Geiger, M., Smidt, T., Alby, M., Miller, B. K., Boomsma, W., Dice, B., Lapchevskyi, K., Weiler, M., Tyszkiewicz, M., Batzner, S., et al. Euclidean neural networks: e3nn. *Zenodo. https://doi. org/10.5281/zenodo*, 2020.

Ghosh, R. and Gupta, A. Scale Steerable Filters for Locally Scale-Invariant Convolutional Neural Networks. *ArXiv*, abs/1906.03861, 2019.

Gupta, J. K. and Brandstetter, J. Towards Multi-spatiotemporal-scale Generalized PDE Modeling. *ArXiv*, abs/2209.15616, 2022.

Haan, P. d., Weiler, M., Cohen, T., and Welling, M. Gauge Equivariant Mesh CNNs: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations (ICLR)*, 2021.

Helwig, J., Zhang, X., Fu, C., Kurtin, J., Wojtowytsch, S., and Ji, S. Group Equivariant Fourier Neural Operators for Partial Differential Equations. In *International Conference on Machine Learning (ICML)*, 2023.

Hendrycks, D. and Gimpel, K. Gaussian Error Linear Units (GELUs). *arXiv: Learning*, 2016.

Hestenes, D. Multivector calculus. *J. Math. Anal. Appl*, 24 (2):313–325, 1968.

Hestenes, D. *Space-time algebra*. Springer, 2015.

Hitzer, E. M. Multivector differential calculus. *Advances in Applied Clifford Algebras*, 12:135–182, 2002.

Holl, P., Thuerey, N., and Koltun, V. Learning to Control PDEs with Differentiable Physics. In *International Conference on Learning Representations (ICLR)*, 2020.

Jenner, E. and Weiler, M. Steerable Partial Differential Operators for Equivariant Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, volume abs/1412.6980, 2015.

Lang, L. and Weiler, M. A Wigner-Eckart Theorem for Group Equivariant Convolution Kernels. In *International Conference on Learning Representations (ICLR)*, 2021.

Lasenby, A., Doran, C., and Gull, S. A multivector derivative approach to lagrangian field theory. *Foundations of Physics*, 23(10):1295–1327, 1993.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. M., and Anandkumar, A. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Lindeberg, T. Scale-space. 2009.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Marcos, D., Kellenberger, B., Lobry, S., and Tuia, D. Scale equivariance in CNNs with vector fields. *arXiv preprint arXiv:1807.11783*, 2018.

Orbán, X. P. and Mira, J. Dimensional scaffolding of electromagnetism using geometric algebra. *European Journal of Physics*, 42(1):015204, 2021.

Romero, D. W., Bekkers, E., Tomczak, J. M., and Hoogendoorn, M. Wavelet Networks: Scale-Translation Equivariant Learning From Raw Time-Series. *Transactions on Machine Learning Research*, 2024.

Ruhe, D., Brandstetter, J., and Forré, P. Clifford Group Equivariant Neural Networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume abs/2305.11141, 2023a.

Ruhe, D., Gupta, J. K., Keninck, S. D., Welling, M., and Brandstetter, J. Geometric Clifford Algebra Networks. In *International Conference on Machine Learning (ICML)*, pp. 29306–29337, 2023b.

Shutty, N. and Wierzynski, C. Computing Representations for Lie Algebraic Networks. *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022.

Sosnovik, I., Szmaja, M., and Smeulders, A. W. M. Scale-Equivariant Steerable Networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Wang, R., Walters, R., and Yu, R. Incorporating Symmetry into Deep Dynamics Models for Improved Generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

Wang, S. Extensions to the navier–stokes equations. *Physics of Fluids*, 34(5), 2022.

Weiler, M. and Cesa, G. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 14334–14345, 2019.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. 3d Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 10402–10413, 2018a.

Weiler, M., Hamprecht, F. A., and Storath, M. Learning Steerable Filters for Rotation Equivariant CNNs. In *Computer Vision and Pattern Recognition (CVPR)*, 2018b.

Weiler, M., Forré, P., Verlinde, E., and Welling, M. Coordinate Independent Convolutional Networks – Isometry and Gauge Equivariant Convolutions on Riemannian Manifolds. *arXiv preprint arXiv:2106.06020*, 2021.

Weiler, M., Forré, P., Verlinde, E., and Welling, M. *Equivariant and Coordinate Independent Convolutional Networks*. 2023. URL https://maurice-weiler.gitlab.io/cnn_book/EquivariantAndCoordinateIndependentCNNs.pdf.

Worrall, D. E. and Welling, M. Deep Scale-spaces: Equivariance Over Scale. In *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 7364–7376, 2019.

Wu, Y. and He, K. Group Normalization. In *European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Zhang, X. and Williams, L. R. Similarity equivariant linear transformation of joint orientation-scale space representations. *arXiv preprint arXiv:2203.06786*, 2022.

Zhdanov, M., Hoffmann, N., and Cesa, G. Implicit Convolutional Kernels for Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Zhu, W., Qiu, Q., Calderbank, A. R., Sapiro, G., and Cheng, X. Scaling-Translation-Equivariant Networks with Decomposed Convolutional Filters. *Journal of Machine Learning Research (JMLR)*, 23:68:1–68:45, 2022.

# Appendix

## A. Implementation details

This appendix provides details on the implementation of CS-CNNs.[16]

Before detailing the Clifford-steerable kernels and convolutions, we first define the following "kernel shell" operation, which is used twice in the final kernel computation. Recall that given the base space $\mathbb{R}^{p,q}$ equipped with the inner product $\eta^{p,q}$, we have a Clifford algebra $\mathrm{Cl}(\mathbb{R}^{p,q})$. We want to compute a kernel that maps from $c_{\text{in}}$ multivector input channels to $c_{\text{out}}$ multivector output channels, i.e.,

$$K : \mathbb{R}^{p,q} \to \mathrm{Hom}_{\mathrm{Vec}}\left(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\text{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\text{out}}}\right). \quad (39)$$

$K$ is defined on any $v \in \mathbb{R}^{p,q}$, which allows to model *point clouds*. In this work, however, we sample it on a grid of shape $X_1, \ldots, X_{p+q}$, analogously to typical CNNs.

### A.1. Clifford Embedding

We briefly discuss how one is able to embed scalars and vectors into the Clifford algebra. This extends to other grades such as bivectors.

Let $s \in \mathbb{R}$ and $v \in \mathbb{R}^{p,q}$. Using the natural isomorphisms $\mathcal{E}^{(0)} : \mathbb{R} \xrightarrow{\sim} \mathrm{Cl}(\mathbb{R}^{p,q})^{(0)}$ and $\mathcal{E}^{(1)} : \mathbb{R}^{p,q} \xrightarrow{\sim} \mathrm{Cl}(\mathbb{R}^{p,q})^{(1)}$, we embed the scalar and vector components into a multivector as

$$m := \mathcal{E}^{(0)}(s) + \mathcal{E}^{(1)}(v) \in \mathrm{Cl}(\mathbb{R}^{p,q}). \quad (40)$$

This is a standard operation in Clifford algebra computations, where we leave the other components of the multivector zero. We denote such embeddings in the algorithms provided below jointly as "CL_EMBED($[s, v]$)".

### A.2. Scalar Orbital Parameterizations

Note that the $\mathrm{O}(p, q)$-steerability constraint

$$K(gv) \overset{!}{=} \rho_{\mathrm{Cl}}^{c_{\text{out}}}(g)\, K(v)\, \rho_{\mathrm{Cl}}^{c_{\text{in}}}(g^{-1}) =: \rho_{\mathrm{Hom}}(g)(K(v)) \qquad \forall\, v \in \mathbb{R}^{p,q},\ g \in \mathrm{O}(p,q)$$

couples kernel values *within* but *not across* different $\mathrm{O}(p, q)$-orbits

$$\begin{aligned} \mathrm{O}(p,q).v &:= \{gv \,|\, g \in \mathrm{O}(p,q)\} \quad (41)\\ &= \{w \,|\, \eta(w,w) = \eta(v,v)\}. \end{aligned}$$

The first line here is the usual definition of group orbits, while the second line makes use of the Def. 2.5 of pseudo-orthogonal groups as metric-preserving linear maps.

[16] https://github.com/maxxxzdn/clifford-group-equivariant-cnns

In the positive-definite case of $\mathrm{O}(n)$, this means that the only degree of freedom is the radial distance from the origin, resulting in (hyper)spherical orbits. Examples of such kernels can be seen in Fig. 7. Other radial kernels are obtained typically through e.g. Gaussian shells, Bessel functions, etc.

In the nondefinite case of $\mathrm{O}(p, q)$, the orbits are hyperboloids, resulting in hyperboloid shells for e.g. the Lorentz group $\mathrm{O}(1, 3)$ as in Fig. 3 (left). In this case, we extend the input to the kernel with a scalar component that now relates to the hyperbolic (squared) distance from the origin.

Specifically, we define an exponentially decaying $\eta^{p,q}$-induced (parameterized) scalar *orbital shell* (analogous to the radial shell of typical Steerable CNNs) in the following way. We parameterize a kernel width $\sigma$ and compute the shell value as

$$s_\sigma(v) = \mathrm{sgn}\left(\eta^{p,q}(v,v)\right) \cdot \exp\left(-\frac{|\eta^{p,q}(v,v)|}{2\sigma^2}\right). \quad (42)$$

The width $\sigma \sim \mathcal{U}(0.4, 0.6)$ is, inspired by (Cesa et al., 2022), initialized with a uniform distribution. Since $\eta^{p,q}(v, v)$ can be negative in the nondefinite case, we take the absolute value and multiply the result by the sign of $\eta^{p,q}(v, v)$. Computation of the kernel shell (SCALARSHELL) is outlined in Function 1. Intuitively, we obtain exponential decay for points far from the origin. However, the sign of the inner product ensures that we clearly disambiguate between "light-like" and "space-like" points. I.e., they are close in Euclidean distance but far in the $\eta^{p,q}$-induced distance. Note that this choice of parameterizing scalar parts of the kernel is not unique and can be experimented with.

### A.3. Kernel Network

Recall from Section 3 that the kernel $K$ is parameterized by a *kernel network*, which is a map

$$\mathcal{K} : \mathbb{R}^{p,q} \to \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\text{out}} \times c_{\text{in}}} \quad (43)$$

implemented as an $\mathrm{O}(p, q)$-equivariant CGENN. It consists of (linearly weighted) geometric product layers followed by multivector activations.

Let $\{v_n\}_{n=1}^N$ be a set of sampling points, where $N := X_1 \cdot \ldots \cdot X_{p+q}$. In the remainder, we leave iteration over $n$ implicit and assume that the operations are performed for each $n$. We obtain a sequence of scalars using the kernel shell

$$s_n := s_\sigma(v_n). \quad (44)$$

The input to the kernel network is a batch of multivectors

$$x_n := \text{CL\_EMBED}([s_n, v_n]). \quad (45)$$

**Function 1** SCALARSHELL

**input** $\eta^{p,q}$, $v \in \mathbb{R}^{p,q}$, $\sigma$.
$s \leftarrow \operatorname{sgn}\left(\eta^{p,q}(v,v)\right) \cdot \exp\left(-\frac{|\eta^{p,q}(v,v)|}{2\sigma^2}\right)$
**return** $s$

**Function 2** CLIFFORDSTEERABLEKERNEL

**input** $p, q$ $\Lambda$, $c_{\text{in}}$, $c_{\text{out}}$, $(v_n)_{n=1}^N \in \mathbb{R}^{p,q}$, CGENN
**output** $\kappa \in \mathbb{R}^{(c_{\text{out}} \cdot 2^d) \times (c_{\text{in}} \cdot 2^d) \times X_1 \times \cdots \times X_{p+q}}$

```
# Weighted Cayley.
```
**for** $i = 1 \dots c_{\text{in}}, o = 1 \dots c_{\text{out}}, a, b, c = 1 \dots p+q$ **do**
  $w^c_{oiab} \sim \mathcal{N}(0, \frac{1}{\sqrt{c_{\text{in}} \cdot N}})$ # Weight init.
  $W^c_{oiab} \leftarrow \Lambda^c_{ab} \cdot w^c_{oiab}$
**end for**

$\sigma \sim \mathcal{U}(0.4, 0.6)$ # Init if needed.
# Compute scalars.
$s_n \leftarrow$ SCALARSHELL$(\eta^{p,q}, v_n, \sigma)$
# Embed $s$ and $v$ into a multivector.
$x_n \leftarrow$ CL_EMBED$([s_n, v_n])$

# Evaluate kernel network.
$\kappa^{io}_n := \text{CGENN}(x_n)$

# Reshape to kernel matrix.
$\kappa \leftarrow$ RESHAPE$(\kappa, (N, c_{\text{out}}, c_{\text{in}}))$

# Compute kernel mask.
**for** $i = 1 \dots c_{\text{in}}, o = 1 \dots c_{\text{out}}, k = 0 \dots p+q$ **do**
  $\sigma_{kio} \sim \mathcal{U}(0.4, 0.6)$ # Init if needed.
  $s^k_{noi} \leftarrow$ SCALARSHELL$(\eta^{p,q}, v_n, \sigma_{kio})$
**end for**

$\kappa^{(k)}_{noi} \leftarrow \kappa^{(k)}_{noi} \cdot s^k_{noi}$ # Mask kernel.

# Kernel head.
$\kappa^c_{noib} \leftarrow \sum_{a=1}^{2^d} \kappa^a_{noi} \cdot W^c_{oiab}$ # Partial weighted geometric product.

# Reshape to final kernel.
$\kappa \leftarrow$ RESHAPE$\left(\kappa, \left(c_{\text{out}} \cdot 2^d, c_{\text{in}} \cdot 2^d, X_1, \dots, X_{p+q}\right)\right)$
**return** $\kappa$

**Function 3** CLIFFORDSTEERABLECONVOLUTION

**input** $F_{\text{in}}$, $(v_n)_{n=1}^N$, ARGS
**output** $F_{\text{out}}$
$F_{\text{in}} \leftarrow$ RESHAPE$(F_{\text{in}}, (B, c_{\text{in}} \cdot 2^d, Y_1, \dots, Y_{p+q}))$
$\kappa \leftarrow$ CLIFFORDSTEERABLEKERNEL$((v_n)_{n=1}^N, \text{ARGS})$
$F_{\text{out}} \leftarrow$ CONV$(F_{\text{in}}, \kappa)$
$F_{\text{out}} \leftarrow$ RESHAPE$(F_{\text{out}}, (B, c_{\text{out}}, Y_1, \dots, Y_{p+q}, 2^d))$
**return** $F_{\text{out}}$

I.e., taking $s$ and $v$ together, they form the scalar and vector components of the CEGNN's input multivector. We found including the scalar component crucial for the correct scaling of the kernel to the range of the grid.

Let $i = 1, \dots, c_{\text{in}}$ and $o = 1, \dots, c_{\text{out}}$ be a sequence of input and output channels. We then have the kernel network output

$$\kappa_{noi} := \mathcal{K}(v_n)_{oi} := \text{CGENN}(x_n)_{oi}\,, \tag{46}$$

where $\kappa_{noi} \in \text{Cl}(\mathbb{R}^{p,q})$ is the output of the kernel network for the input multivector $x_n$ (embedded from the scalar $s_n$ and vector $v_n$). Once the output stack of multivectors is computed, we reshape it from shape $(N, c_{\text{out}} \cdot c_{\text{in}})$ to shape $(N, c_{\text{out}}, c_{\text{in}})$, resulting in the kernel matrix

$$\kappa \leftarrow \text{RESHAPE}\left(\kappa, (N, c_{\text{out}}, c_{\text{in}})\right)\,, \tag{47}$$

where now $\kappa \in \text{Cl}(\mathbb{R}^{p,q})^{N \times c_{\text{out}} \times c_{\text{in}}}$. Note that $k_n \in \text{Cl}(\mathbb{R}^{p,q})^{c_{\text{out}} \times c_{\text{in}}}$ is a matrix of multivectors, as desired.

### A.4. Masking

We compute a second set of scalars which will act as a mask for the kernel. This is inspired by Steerable CNNs to ensure that the (e.g., radial) orbits of compact groups are fully represented in the kernel, as shown in Figure 7. However, note that for $\text{O}(p,q)$-steerable kernels with both $p, q \neq 0$ this is never fully possible since $\text{O}(p,q)$ is in general not compact, and all orbits except for the origin extend to infinity. This can e.g. be seen in the hyperbolic-shaped kernels in Figure 3.

For equivariance to hold in practice, whole orbits would need to be present in the kernel, which is not possible if the kernel is sampled on a grid with finite support. This is not specific to our architecture, but is a consequence of the orbits' non-compactness. The same issue arises e.g. in *scale-equivariant* CNNs (Romero et al., 2024; Worrall & Welling, 2019; Ghosh & Gupta, 2019; Sosnovik et al., 2020; Bekkers, 2020; Zhu et al., 2022; Marcos et al., 2018; Zhang & Williams, 2022). Further experimenting is needed to understand the impact of truncating the kernel on the final performance of the model.

We invoke the kernel shell function again to compute a mask for each $k = 0, \dots, p+q$, $i = 1, \dots, c_{\text{in}}$, $o = 1, \dots, c_{\text{out}}$. That is, we have a weight array $\sigma_{kio}$, initialized identically as earlier, which is reused for each position in the grid.

$$s^k_{nio} := s_{\sigma_{kio}}(v_n)\,. \tag{48}$$

We then mask the kernel by scalar multiplication with the shell, i.e.,

$$\kappa^{(k)}_{kio} \leftarrow \kappa^{(k)}_{nio} \cdot s^k_{nio}\,. \tag{49}$$

### A.5. Kernel Head

Finally, the *kernel head* turns the "multivector matrices" into a kernel that can be used by, for example, `torch.nn.ConvNd` or `jax.lax.conv`. This is done by a partial evaluation of a (weighted) geometric product. Let $\mu, \nu \in \mathrm{Cl}(\mathbb{R}^{p,q})$ be two multivectors. Recall that $\dim \mathrm{Cl}(\mathbb{R}^{p,q}) = 2^{p+q} = 2^d$.

$$(\mu \bullet \nu)^C = \sum\nolimits_A \sum\nolimits_B \mu^A \cdot \nu^B \cdot \Lambda^C_{AB}\,, \tag{50}$$

where $A, B, C \subseteq [d]$ are multi-indices running over the $2^d$ basis elements of $\mathrm{Cl}(\mathbb{R}^{p,q})$. Here, $\Lambda \in \mathbb{R}^{2^d \times 2^d \times 2^d}$ is the *Clifford multiplication table* of $\mathrm{Cl}(\mathbb{R}^{p,q})$, also sometimes called a *Cayley table*. It is defined as

$$\Lambda^C_{A,B} = \begin{cases} 0 & \text{if } A \triangle B \neq C \\ \mathrm{sgn}^{A,B} \cdot \bar{\eta}(e_{A\cap B}, e_{A \cap B}) & \text{if } A \triangle B = C \end{cases}. \tag{51}$$

Here, $\triangle$ denotes the symmetric difference of sets, i.e., $A \triangle B = (A \setminus B) \cup (B \setminus A)$. Further,

$$\mathrm{sgn}^{A,B} := (-1)^{n_{A,B}}, \tag{52}$$

where $n_{A,B}$ is the number of adjacent "swaps" one needs to fully sort the tuple $(i_1, \ldots, i_s, j_1, \ldots, j_t)$, where $A = \{i_1, \ldots, i_s\}$ and $B = \{j_1, \ldots, j_t\}$. In the following, we identify the multi-indices $A$, $B$, and $C$ with a relabeling $a$, $b$, and $c$ that run from 1 to $2^d$.

Altogether, $\Lambda$ defines a multivector-valued bilinear form which represents the geometric product relative to the chosen multivector basis. We can weight its entries with parameters $w^c_{oiab} \in \mathbb{R}$, initialized as $w^c_{oiab} \sim \mathcal{N}(0, \frac{1}{\sqrt{c_{\text{in}} \cdot N}})$. These weightings can be redone for each input channel and output channel, as such we have a weighted Cayley table $W \in \mathbb{R}^{2^d \times 2^d \times 2^d \times c_{\text{in}} \times c_{\text{out}}}$ with entries

$$W^c_{oiab} \;:=\; \Lambda^c_{ab} w^c_{oiab}\,. \tag{53}$$

An ablation study in appendix D.4 demonstrates the great relevance of the weighting parameters empirically.

Given the kernel matrix $\hbar$, we compute the kernel by partial (weighted) geometric product evaluation, i.e.,

$$\hbar^c_{noib} \leftarrow \sum\nolimits_{a=1}^{2^d} \hbar^a_{noi} \cdot W^c_{oiab}\,. \tag{54}$$

Finally, we reshape and permute $\hbar^c_{noib}$ from shape $(N, c_{\text{out}}, c_{\text{in}}, 2^d, 2^d)$ to its final shape, i.e.,

$$\hbar \leftarrow \textsc{Reshape}\left(\hbar, \left(c_{\text{out}} \cdot 2^d, c_{\text{in}} \cdot 2^d, X_1, \ldots, X_{p+q}\right)\right)\,.$$

This is the final kernel that can be used in a convolutional layer, and can be interpreted (at each sample coordinate) as an element of $\mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\text{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\text{out}}}\big)$. The pseudocode for the Clifford-steerable kernel (CliffordSteerableKernel) is given in Function 2.

### A.6. Clifford-steerable convolution:

As defined in Section 3, Clifford-steerable convolutions can be efficiently implemented with conventional convolutional machinery such as `torch.nn.ConvNd` or `jax.lax.conv` (see Function 3 (CliffordSteerableConvolution) for pseudocode). We now have a kernel $\hbar \in \mathbb{R}^{(c_{\text{out}} \cdot 2^d) \times (c_{\text{in}} \cdot 2^d) \times X_1 \times \cdots \times X_{p+q}}$ that can be used in a convolutional layer. Given batch size $B$, we now reshape the input stack of multivector fields $(B, c_{\text{in}}, Y_1, \ldots, Y_{p+q}, 2^d)$ into $(B, c_{\text{in}} \cdot 2^d, Y_1, \ldots, Y_{p+q})$. The output array of shape $(B, c_{\text{out}} \cdot 2^d, Y_1, \ldots, Y_{p+q})$ is obtained by convolving the input with the kernel, which is then reshaped to $(B, c_{\text{out}}, Y_1, \ldots, Y_{p+q}, 2^d)$, which can then be interpreted as a stack of multivector fields again.

## B. Limitations

From the viewpoint of general steerable CNNs, there are some limitations:

- There exist *more general field types* ($\mathrm{O}(p,q)$-representations) than multivectors, for which CS-CNNs do not provide steerable kernels. For connected Lie groups, such as the subgroups $\mathrm{SO}^+(p,q)$, these types can in principle be computed numerically (Shutty & Wierzynski, 2022).
- CGENNs and CS-CNNs rely on equivariant operations that treat multivector-grades $\mathrm{Cl}^{(k)}(V, \eta)$ as "atomic" features. However, it is not clear whether grades are always *irreducible* representations, that is, there might be further equivariant degrees of freedom which would treat irreducible sub-representations independently.
- We observed that the steerable kernel spaces of CS-CNNs are not necessarily *complete*, that is, certain degrees of freedom might be missing. However, we show in Appendix C how they are recovered by composing multiple convolutions.
- $\mathrm{O}(p,q)$ and their group orbits on $\mathbb{R}^{p,q}$ are for $p, q \neq 0$ *non-compact*; for instance, the hyperbolas in spacetimes $\mathbb{R}^{1,q}$ extend to infinity. In practice, we sample convolution kernels on a finite sized grid as shown in Fig. 3 (left). This introduces a cutoff, breaking equivariance for large transformations. Note that this is an issue not specific to CS-CNNs, but it applies e.g. to scale-equivariant CNNs as well (Bekkers, 2020; Romero et al., 2024).

Despite these limitations, CS-CNNs excel in our experiments. A major advantage of CGENNs and CS-CNNs is that they allow for a simple, unified implementation for arbitrary signatures $(p,q)$. This is remarkable, since steerable kernels usually need to be derived for each symmetry group individually. Furthermore, our implementation applies both to multivector fields sampled on pixel grids and point clouds.

## C. Completeness of kernel spaces

In order to not over-constrain the model, it is essential to parameterize a *complete basis* of $\mathrm{O}(p,q)$-steerable kernels. Comparing our implicit $\mathrm{O}(2,0) = \mathrm{O}(2)$-steerable kernels with the analytical solution by (Weiler & Cesa, 2019), we find that certain *degrees of freedom are missing*; see Fig. 7.

However, while these degrees of freedom are *missing in a single convolution operation*, they can be *fully recovered by applying two consecutively convolutions*. This suggests that the overall expressiveness of CS-CNNs is (at least for $\mathrm{O}(2)$) not diminished. Moreover, two convolutions with kernels $\widehat{K}$ and $K$ can always be expressed as a single convolution with a composed kernel $\widehat{K} * K$. As visualized below, this composed kernel recovers the full degrees of freedom reported in (Weiler & Cesa, 2019):

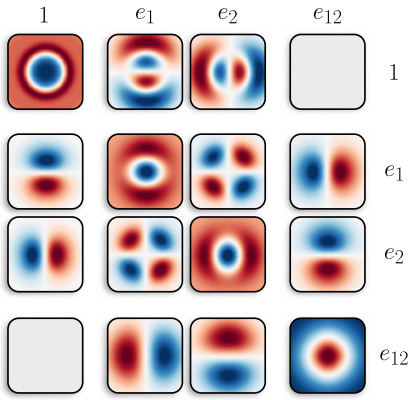



*Figure 6.*

The following two sections discuss the initial differences in kernel parametrizations and how they are resolved by adding a second linear or convolution operation. Unless stated otherwise, we focus here on $c_{\text{in}} = c_{\text{out}} = 1$ channels to reduce clutter.

### C.1. Coupled radial dependencies in CS-CNN kernels

The first issue is that the CS-CNN parametrization implies a *coupling of radial degrees of freedom*. To make this precise, note that the $\mathrm{O}(2)$-steerability constraint

$$K(gv) \overset{!}{=} \rho_{\mathrm{Cl}}^{c_{\text{out}}}(g)\, K(v)\, \rho_{\mathrm{Cl}}^{c_{\text{in}}}(g^{-1}) \quad \forall\, v \in \mathbb{R}^2,\, g \in \mathrm{O}(2)$$

decouples into independent constraints on individual $\mathrm{O}(2)$-orbits on $\mathbb{R}^2$, which are rings at different radii (and the origin); visualized in Fig. 2 (left). (Weiler et al., 2018a; Weiler & Cesa, 2019) parameterize the kernel therefore in (hyper)spherical coordinates. In our case these are *polar coordinates* of $\mathbb{R}^2$, i.e. a radius $r \in \mathbb{R}_{\geq 0}$ and angle $\phi \in S^1$:

$$K(r,\phi) \;:=\; R(r)\kappa(\phi) \tag{55}$$

The $\mathrm{O}(2)$-steerability constraint affects only the angular part and leaves the radial part entirely free, such that it can be parameterized in an arbitrary basis or via an MLP.

**e2cnn:** Weiler & Cesa (2019) solved analytically for complete bases of the angular parts. Specifically, they derive solutions

$$K^k_{\ n}(r,\phi) \;=\; R^k_{\ n}(r)\kappa^k_{\ n}(\phi) \tag{56}$$

for any pair of input and output field types (irreps of grades) $n$ and $k$, respectively. This complete basis of $\mathrm{O}(2)$-steerable kernels is shown in the bottom table of Fig. 7.

**CS-CNNs:** CS-CNNs parameterize the kernel in terms of a kernel network $\mathcal{K} : \mathbb{R}^{p,q} \to \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\text{out}} \times c_{\text{in}}}$, visualized in Fig. 7 (top). Expressed in polar coordinates, assuming $c_{\text{in}} = c_{\text{out}} = 1$, and considering the independence of $\mathcal{K}$ on different orbits due to its $\mathrm{O}(2)$-equivariance, we get the factorization

$$\mathcal{K}(r,\phi)^{(m)} \;=\; R_m(r)\kappa_m(\phi)\,, \tag{57}$$

where $m$ is the grade of the multivector-valued output. As described in Appendix A.5 (Eq. (53)), the kernel head operation $H$ expands this output by multiplying it with weights $W^k_{mn} = \Lambda^k_{mn} w^k_{mn}$, where $w^k_{mn} \in \mathbb{R}$ are parameters and $\Lambda^k_{mn} \in \{-1, 0, 1\}$ represents the geometric product relative to the standard basis of $\mathbb{R}^{p,q}$. Note that we do not consider multiple in or output channels here. The final expanded kernel for CS-CNNs is hence given by

$$\begin{aligned} K^k_{\ n}(r,\phi) \;&=\; \sum_m W^k_{mn}\mathcal{K}(r,\phi)^{(m)} \\ &=\; \sum_m \Lambda^k_{mn} w^k_{mn} R_m(r)\kappa_m(\phi)\,. \end{aligned} \tag{58}$$

These solutions are listed in the top table in Fig. 7, and visualized in the graphics above.[17]

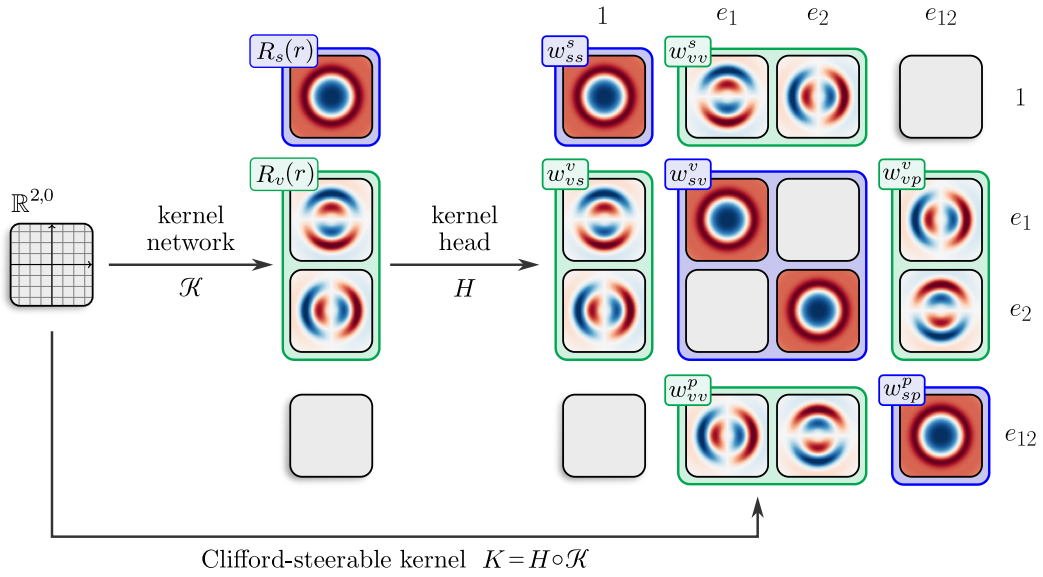
**Comparison:** Note that the complete solutions by (Weiler & Cesa, 2019) allow for a *different radial part* $R^k_{\ n}$ for each pair of input and output type (grade/irrep). In contrast, the CS-CNN parametrization expands *coupled radial parts* $R_m$, additionally multiplying them with weights $w^k_{mn}$ (highlighted in the table in blue and green). The CS-CNN parametrization is therefore clearly less general (incomplete).

**Solutions:** One idea to resolve this shortcoming is to make the weighted geometric product parameters themselves radially dependent,

$$w^k_{mn} :\; \mathbb{R}_{\geq 0} \to \mathbb{R}, \quad r \mapsto w^k_{mn}(r)\,, \tag{59}$$

for instance by parameterizing the weights with a neural network. This would fully resolve the under-parametrization,

[17] The parameter $\Lambda^k_{mn}$ appears in the table as selecting to which entry $k, n$ of the table grade $\mathcal{K}(r,\phi)^{(m)}$ is added (optionally with minus signs).




CS-CNN parametrization

| out \ in | scalar<br>$1$ | vector<br>$[e_1, e_2]^\top$ | pseudoscalar<br>$e_{12}$ |
|---|---|---|---|
| $1$ | $w^s_{ss} R_s(r) \,[1]$ | $w^s_{vv} R_v(r) \,[-\sin(\phi)\ \cos(\phi)]$ | $\varnothing$ |
| $\begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$ | $w^v_{vs} R_v(r) \begin{bmatrix} -\sin(\phi) \\ \cos(\phi) \end{bmatrix}$ | $w^v_{sv} R_s(r) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ | $w^v_{vp} R_v(r) \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix}$ |
| $e_{12}$ | $\varnothing$ | $w^p_{vv} R_v(r) \,[\cos(\phi)\ \ \sin(\phi)]$ | $w^p_{sp} R_s(r) \,[1]$ |

complete `e2cnn` parametrization (Weiler & Cesa, 2019)

| out \ in | $1$ | $[e_1, e_2]^\top$ | $e_{12}$ |
|---|---|---|---|
| $1$ | $R^s{}_s(r) \,[1]$ | $R^s{}_v(r) \,[-\sin(\phi)\ \cos(\phi)]$ | $\varnothing$ |
| $\begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$ | $R^v{}_s(r) \begin{bmatrix} -\sin(\phi) \\ \cos(\phi) \end{bmatrix}$ | $R^v{}_v(r) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\widehat{R}^v{}_v(r) \begin{bmatrix} \cos(2\phi) & \sin(2\phi) \\ \sin(2\phi) & -\cos(2\phi) \end{bmatrix}$ | $R^v{}_p(r) \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix}$ |
| $e_{12}$ | $\varnothing$ | $R^p{}_v(r) \,[\cos(\phi)\ \ \sin(\phi)]$ | $R^p{}_p(r) \,[1]$ |

*Figure 7.* Comparison of the parametrization of $\mathrm{O}(2)$-steerable kernels in CS-CNNs (top and middle) and `e2cnn` (bottom). While the `e2cnn` solutions are proven to be *complete*, CS-CNN seems to miss certain degrees of freedom:

(1) Their *radial parts are coupled* in the components highlighted in blue and green, while `escnn` allows for independent radial parts. By "coupled" we mean that they are merely scaled relative to each other with weights $w^k_{mn}$ from the weighted geometric product operation in the kernel head $H$, where $m$ labels grade $\mathcal{K}^{(m)}$ of the kernel network output while $n, k$ label input and output grades of the expanded kernel in $\mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q}),\, \mathrm{Cl}(\mathbb{R}^{p,q})\big)$;

(2) CS-CNN is missing kernels of angular frequency 2 that are admissible for mapping between vector fields; highlighted in red.

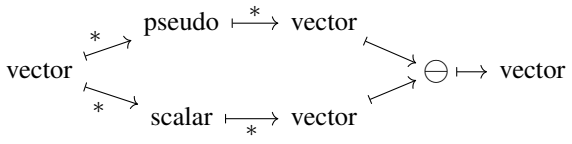
As explained in Appendix C, these *missing degrees of freedom are recovered* when *composing two convolution layers*. A kernel corresponding to the composition of two convolutions in a single one is visualized in Fig. 6.

and would preserve equivariance, since $\mathrm{O}(2)$-steerability depends only on the angular variable.

However, doing this is actually not necessary, since the missing flexibility of radial parts can always be resolved by running a convolution followed by a linear layer (or a second convolution) when $c_{\text{out}} > 1$. The reason for this is that different channels $i = 1, \dots, c_{\text{out}}$ of a kernel network $\mathcal{K} : \mathbb{R} \to \mathrm{Cl}(\mathbb{R})^{c_{\text{out}} \times c_{\text{in}}}$ do have independent radial parts. Their convolution responses in different channels can by a subsequent linear layer be mixed with grade-dependent weights. By linearity, this is equivalent to immediately mixing the channels' radial parts with grade-dependent weights, resulting in effectively *decoupled radial parts*.

### C.2. Circular harmonics order 2 kernels

A second issue is that the CS-CNN parametrization is *missing a basis kernel of angular frequency* 2 that maps between vector fields; highlighted in red in the bottom table of Fig. 7. However, it turns out that this degree of freedom is reproduced as the difference of two consecutive convolutions ($*$), one mapping vectors to pseudoscalars and back to vectors, the other one mapping vectors to scalars and back to vectors, as suggested in the (non-commutative!) computation flow diagram below:




As background on the angular frequency 2 kernel, note that $\mathrm{O}(2)$-steerable kernels between irreducible field types of angular frequencies $j$ and $l$ contain angular frequencies $|j - l|$ and $j + l$ – this is a consequence of the *Clebsch-Gordan decomposition* of $\mathrm{O}(2)$-irrep tensor products (Lang & Weiler, 2021). We identify multivector grades $\mathrm{Cl}(\mathbb{R}^{2,0})^{(k)}$ with the following $\mathrm{O}(2)$-irreps:[18][19]

$$\begin{aligned} \text{scalars} \in \mathrm{Cl}(\mathbb{R}^{2,0})^{(0)} &\leftrightarrow \text{trivial irrep } (j=0) \\ \text{vectors} \in \mathrm{Cl}(\mathbb{R}^{2,0})^{(1)} &\leftrightarrow \text{defining irrep } (j=1) \\ \text{pseudo-scalars} \in \mathrm{Cl}(\mathbb{R}^{2,0})^{(2)} &\leftrightarrow \text{sign-flip irrep } (j=0) \end{aligned}$$

Kernels that map vector fields ($j=1$) to vector fields ($l=1$) should hence contain angular frequencies $|j-l| = 0$ and $j+l = 2$. The latter is missing since $\mathrm{O}(2)$-irreps of order 2 are not represented by any grade of $\mathrm{Cl}(\mathbb{R}^{2,0})$.

To solve this issue, it seems like one would have to replace the CEGNNs underlying the kernel network $\mathcal{K}$ with a more general $\mathrm{O}(2)$-equivariant MLP, e.g. (Finzi et al., 2021). However, it can as well be implemented as a succession of two convolution operations. To make this claim plausible, observe first that convolutions are associative, that is, two consecutive convolutions with kernels $K$ and $\widehat{K}$ are equivalent to a single convolution with kernel $\widehat{K} * K$:

$$\widehat{K} * \big(K * f\big) \;=\; \big(\widehat{K} * K\big) * f \tag{60}$$

Secondly, convolutions are linear, such that

$$\alpha(\widehat{K} * f) + \beta(K * f) \;=\; \big(\alpha\widehat{K} + \beta K\big) * f \tag{61}$$

for any $\alpha, \beta \in \mathbb{R}$.

Using associativity, we can express two consecutive convolutions, first going from vector to scalar fields via

$$K^s{}_v(r,\phi) \;=\; R^s{}_v(r) \begin{pmatrix} -\sin(\phi) & \cos(\phi) \end{pmatrix} \tag{62}$$

then going back from scalars to vectors via

$$K^v{}_s(r,\phi) \;=\; R^v{}_s(r) \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \end{pmatrix} \tag{63}$$

as a single convolution between vector fields, where the combined kernel is given by:

$$\Sigma^v{}_v \;:=\; K^v{}_s * K^s{}_v \tag{64}$$

$$= \quad * \quad = \quad$$

We can similar define a convolution going from vector to pseudoscalar fields via

$$K^p{}_v(r,\phi) \;=\; R^p{}_v(r) \begin{pmatrix} \cos(\phi) & \sin(\phi) \end{pmatrix} \tag{65}$$

and back to vector fields via

$$K^v{}_p(r,\phi) \;=\; R^v{}_p(r) \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix} \tag{66}$$

as a single convolution with combined kernel:

$$\Pi^v{}_v \;:=\; K^p{}_v * K^v{}_p \tag{67}$$

$$= \quad * \quad = \quad$$

By linearity, we can define yet another convolution between vector fields by taking the difference of these kernels, which results in:

$$\Pi^v{}_v - \Sigma^v{}_v \;=\; \tag{68}$$

[18] As mentioned earlier, multivector grades may in general not be *irreducible*, however, for $(p, q) = (2, 0)$ they are.

[19] There are two different $\mathrm{O}(2)$-irreps corresponding to $j = 0$ (trivial and sign-flip); see (Weiler et al., 2023)[Section 5.3.4].

Such kernels parameterize exactly the missing $\mathrm{O}(2)$-steerable kernels of angular frequency 2; highlighted in red in the bottom table in Fig. 7. This shows that the missing kernels can be recovered by two convolutions, if required.

The "visual proof" by convolving kernels is clearly only suggestive. To make it precise, it would be required to compute the convolutions of two kernels analytically. This is easily done by identifying circular harmonics with derivatives of Gaussian kernels; a relation that is well known in classical computer vision (Lindeberg, 2009).

## D. Experimental details

### D.1. Model details:

For ResNets, we follow the setup of Wang et al. (2021); Brandstetter et al. (2023); Gupta & Brandstetter (2022): the ResNet baselines consist of 8 residual blocks, each comprising two convolution layers with $7 \times 7$ (or $7 \times 7 \times 7$ for 3D) kernels, shortcut connections, group normalization (Wu & He, 2018), and GeLU activation functions (Hendrycks & Gimpel, 2016). We use two embedding and two output layers, i.e., the overall architectures could be classified as Res-20 networks. Following (Gupta & Brandstetter, 2022; Brandstetter et al., 2023), we abstain from employing down-projection techniques and instead maintain a consistent spatial resolution throughout the networks. The best models have approx. 7M parameters for Navier-Stokes and 1.5M parameters for Maxwell's equations, in both 2D and 3D.

### D.2. Optimization:

For each experiment and each model, we tuned the learning rate to find the optimal value. Each model was trained until convergence. For optimization, we used Adam optimizer (Kingma & Ba, 2015) with no learning decay and cosine learning rate scheduler (Loshchilov & Hutter, 2017) to reduce the initial value by the factor of 0.01. Training was done on a single node with 4 NVIDIA GeForce RTX 2080 Ti GPUs.

### D.3. Datasets

**Navier Stokes:** We use the Navier-Stokes data from Gupta & Brandstetter (2022), which is based on ΦFlow (Holl et al., 2020). It is simulated on a grid with spatial resolution of $128 \times 128$ pixels of size $\Delta x = \Delta y = 0.25$m and temporal resolution of $\Delta t = 1.5$s. For validation and testing, we randomly selected 1024 trajectories from corresponding partitions.

**Maxwell 3D:** Simulations of the 3D Maxwell equations are taken from Brandstetter et al. (2023). This data is discretized on a grid with a spatial resolution of $32 \times 32 \times 32$ voxels with $\Delta x = \Delta y = \Delta z = 5 \cdot 10^{-7}$m and was reported to have a temporal resolution of $\Delta t = 50$s. In the *non-relativistically modeled* setting $\mathrm{Cl}(\mathbb{R}^{3,0})$, $\mathbf{E}$ is treated as a vector field, and $\mathbf{B}$ as a bivector field. Validation and test sets comprise 128 simulations.

**Maxwell 2D:** We simulate data for Maxwell's equations on spacetime $\mathbb{R}^{2,1}$ using `PyCharge` (Filipovich & Hughes, 2022). Electromagnetic fields are emitted by point sources that move, orbit and oscillate at relativistic speeds. The spacetime grid has a resolution of 128 points in both spatial and the temporal dimension. Its spatial extent are 50nm and the temporal extent are $3.77 \cdot 10^{-14}$s.

Sampled simulations contain between 2 to 4 oscillating charges and 1 to 2 orbiting charges. The sources have charges sampled uniformly as integer values between $-3$e and 3e. Their positions are sampled uniformly on the grid, with a predefined minimum initial distance between them. Each charge has a random linear velocity and either oscillates in a random direction or orbits with a random radius. Oscillation and rotation frequencies, as well as velocities are sampled such that the overall particle velocity does not exceed 0.85c, which is necessary since the PyCharge simulation becomes unstable beyond this limit.

As the field strengths span many orders of magnitude, we normalize the generated fields by dividing bivectors by their Minkowski norm and multiplying them by the logarithm of this norm. This step is non-trivial sincewMinkowski-norms can be zero or negative, however, we found that they are always positive in the generated data. We filter out numerical artifacts by removing outliers with a standard deviation greater than 20. The final dataset comprises 2048 training, 256 validation and 256 test simulations.

**Dataset symmetries:** The classical Navier Stokes equations are *Galilean invariant* (Wang, 2022). Our CS-CNN for $\mathrm{Cl}(\mathbb{R}^2)$ is $\mathrm{E}(2)$-equivariant, capturing the subgroup of isometries without boosts.

Maxwell's equations are *Poincaré invariant*. Similar to the case of Navier Stokes, our model for $\mathrm{Cl}(\mathbb{R}^3)$ is $\mathrm{E}(3)$-equivariant. The relativistic spacetime model for $\mathrm{Cl}(\mathbb{R}^{1,2})$ is fully equivariant w.r.t. the Poincaré group $\mathrm{E}(1,2)$.

The invariance of a system's equations of motion imply an equivariant system dynamics. This statement assumes that the system is transformed *as a whole*, i.e. together with boundary conditions or background fields. It does obviously not hold when fixed symmetry-breaking boundary conditions or background fields are given. However, implicit kernels may in this case be informed about the symmetry breaking geometric structure by providing it in form of additional inputs to the kernel network as described in (Zhdanov et al., 2023).

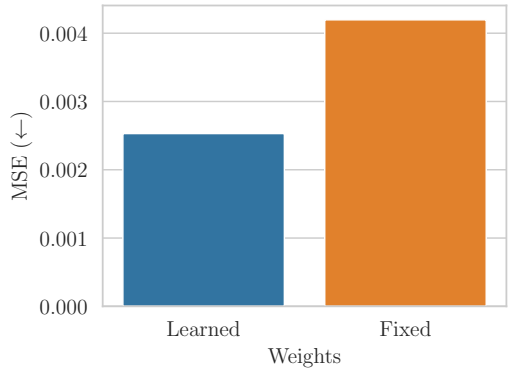



*Figure 8.* Performance of CS-CNNs with freely *learned* weights in the kernel head and such that ablate to *fixed* weights $w^k_{mn,ij} = 1$.

### D.4. Kernel head weight ablation

As discussed in Def. 3.1 and Appendix A.5, the kernel head is essentially a partially evaluated geometric product operation with additional weighting parameters that are learned during training. To check how relevant this weighting is in practice, we ran an ablation study that fixed all kernel head weights to $w^k_{mn,ij} = 1$. It turns out that the weighting is quite relevant: Our fully weighted CS-CNN achieved a test MSE of $2.53 \cdot 10^{-3}$ on the Navier Stokes forecasting task, while the MSE for the fixed weight CS-CNN increased to $4.30 \cdot 10^{-3}$; see Fig. 8. This drastic loss in performance is explained by the fact that these weights allow to scale different kernel channels relative to each other as visualized in Fig. 7, which is essential to parameterize the complete space of steerable kernels.

## E. The Clifford Algebra

For completeness purposes and to complement Section 2.3, in this sections, we give a short and formal definition of the Clifford algebra. For this, we first need to introduce the tensor algebra of a vector space.

**Definition E.1** (The tensor algebra)**.** *Let $V$ be finite dimensional $\mathbb{R}$-vector space of dimension $d$. Then the* tensor algebra *of $V$ is defined as follows:*

$$\begin{aligned}\mathrm{Tens}(V) &:= \bigoplus_{m=0}^{\infty} V^{\otimes m} \\ &= \mathrm{span}\left\{v_1 \otimes \cdots \otimes v_m \,\middle|\, m \geq 0, v_i \in V\right\},\end{aligned} \tag{69}$$

*where we used the following abbreviations for the $m$-times tensor product of $V$ for $m \geq 0$:*

$$V^{\otimes m} := \underbrace{V \otimes \cdots \otimes V}_{m\text{-times}}, \qquad V^{\otimes 0} := \mathbb{R}. \tag{70}$$

Note that the above definition turns $(\mathrm{Tens}(V), \otimes)$ into a (non-commutative, infinite dimensional, unital, associative) algebra over $\mathbb{R}$. In fact, the tensor algebra $(\mathrm{Tens}(V), \otimes)$ is, in some sense, the biggest algebra generated by $V$.

We now have the tools to give a proper definition of the Clifford algebra:

**Definition E.2** (The Clifford algebra)**.** *Let $(V, \eta)$ be a finite dimensional innner product space over $\mathbb{R}$ of dimension $d$. The* Clifford algebra *of $(V, \eta)$ is then defined as the following quotient algebra:*

$$\mathrm{Cl}(V, \eta) := \mathrm{Tens}(V)/I(\eta), \tag{71}$$

$$\begin{aligned}I(\eta) &:= \left\langle v \otimes v - \eta(v, v) \cdot 1_{\mathrm{Tens}(V)} \middle| v \in V \right\rangle \\ &:= \mathrm{span}\Big\{ x \otimes \left(v \otimes v - \eta(v, v) \cdot 1_{\mathrm{Tens}(V)}\right) \otimes y \\ &\qquad\qquad \Big| \; v \in V, x, y \in \mathrm{Tens}(V) \Big\},\end{aligned} \tag{72}$$

*where $I(\eta)$ denotes the* two-sided ideal *of* $\mathrm{Tens}(V)$ *generated by the relations $v \otimes v \sim \eta(v, v) \cdot 1_{\mathrm{Tens}(V)}$ for all $v \in V$.*

*The product on* $\mathrm{Cl}(V, \eta)$ *that is induced by the tensor product $\otimes$ is called the* geometric product $\bullet$ *and will be denoted as follows:*

$$x_1 \bullet x_2 := [z_1 \otimes z_2], \tag{73}$$

*with the equivalence classes $x_i = [z_i] \in \mathrm{Cl}(V, \eta)$, $i = 1, 2$.*

Note that, since $I(\eta)$ is a two-sided ideal, the geometric product is well-defined. The above construction turns $(\mathrm{Cl}(V, \eta), \bullet)$ into a (non-commutative, unital, associative) algebra over $\mathbb{R}$.

In some sense, $(\mathrm{Cl}(V, \eta), \bullet)$ is the biggest (non-commutative, unital, associative) algebra $(\mathcal{A}, \bullet)$ over $\mathbb{R}$ that is generated by $V$ and satisfies the relations $v \bullet v = \eta(v, v) \cdot 1_{\mathcal{A}}$ for all $v \in V$.

It turns out that $(\mathrm{Cl}(V, \eta), \bullet)$ is of the finite dimension $2^d$ and carries a *parity grading* of algebras and a *multivector grading* of vector spaces, see (Ruhe et al., 2023b) Appendix D. More properties are also explained in Section 2.3.

From an abstract, theoretical point of view, the most important property of the Clifford algebra is its *universal property*, which fully characterizes it:

**Theorem E.3** (The universal property of the Clifford algebra)**.** *Let $(V, \eta)$ be a finite dimensional innner product space over $\mathbb{R}$ of dimension $d$. For every (non-commutative, unital, associative) algebra $(\mathcal{A}, *)$ over $\mathbb{R}$ and every $\mathbb{R}$-linear map $f : V \to \mathcal{A}$ such that for all $v \in V$ we have:*

$$f(v) * f(v) = \eta(v, v) \cdot 1_{\mathcal{A}}, \tag{74}$$

*there exists a unique algebra homomorphism (over $\mathbb{R}$):*

$$\bar{f} : (\mathrm{Cl}(V, \eta), \bullet) \to (\mathcal{A}, *), \tag{75}$$

*such that $\bar{f}(v) = f(v)$ for all $v \in V$.*

*Proof.* The map $f : V \to \mathcal{A}$ uniquely extends to an algebra homomorphism on the tensor algebra:

$$f^{\otimes} : \mathrm{Tens}(V) \to \mathcal{A}, \tag{76}$$

given by:

$$f^{\otimes}\left(\sum_{i\in I} c_i \cdot v_{i,1} \otimes \cdots \otimes v_{i,l_i}\right) := \sum_{i\in I} c_i \cdot f(v_{i,1}) * \cdots * f(v_{i,l_i}). \tag{77}$$

Because of Equation (74) we have for every $v \in V$:

$$\begin{aligned} &f^{\otimes}\left(v \otimes v - \eta(v,v)\cdot 1_{\mathrm{Tens}(V)}\right) \\ &= f(v) * f(v) - \eta(v,v)\cdot 1_{\mathcal{A}} &(78)\\ &= 0, &(79)\end{aligned}$$

and thus:

$$f^{\otimes}(I(\eta)) = 0. \tag{80}$$

This shows that $f^{\otimes}$ then factors through the thus well-defined induced quotient map of algebras:

$$\begin{aligned} \bar{f} : \mathrm{Cl}(V,\eta) = \mathrm{Tens}(V)/I(\eta) &\to \mathcal{A} &(81)\\ \bar{f}([z]) &:= f^{\otimes}(z). &(82)\end{aligned}$$

This shows the claim. □

**Remark E.4** (The universal property of the Clifford algebra)**.** *The universal property of the Clifford algebra can more explicitely be stated as follows:*

*If $f$ satisfies Equation (74) and $x \in \mathrm{Cl}(V,\eta)$, then we can take* any *representation of $x$ of the following form:*

$$x = \sum_{i\in I} c_i \cdot v_{i,1} \bullet \cdots \bullet v_{i,l_i}, \tag{83}$$

*with any finite index sets $I$, any $l_i \in \mathbb{N}$ and any coefficients $c_0, c_i \in \mathbb{R}$ and any vectors $v_{i,j} \in V$, $j = 1,\dots,l_i$, $i \in I$, and, then we can compute $\bar{f}(x)$ by the following formula:*

$$\bar{f}(x) = \sum_{i\in I} c_i \cdot f(v_{i,1}) * \cdots * f(v_{i,l_i}), \tag{84}$$

*and* no ambiguity *can occur for $\bar{f}(x)$ if one uses a different such representation for $x$.*

**Example E.5.** *The universal property of the Clifford algebra can, for instance, be used to show that the action of the (pseudo-)orthogonal group:*

$$\begin{aligned} \mathrm{O}(V,\eta)\times\mathrm{Cl}(V,\eta) &\to \mathrm{Cl}(V,\eta), &(85)\\ (g,x) &\mapsto \rho_{\mathrm{Cl}}(g)(x), &(86)\end{aligned}$$

*given by:*

$$\rho_{\mathrm{Cl}}(g)\left(\sum_{i\in I} c_i \cdot v_{i,1} \bullet \cdots \bullet v_{i,l_i}\right) := \sum_{i\in I} c_i \cdot (gv_{i,1}) \bullet \cdots \bullet (gv_{i,l_i}), \tag{87}$$

*is well-defined. For this one only would need to check Equation (74) for $v \in V$:*

$$\begin{aligned} (gv)\bullet(gv) &= \eta(gv,gv)\cdot 1_{\mathrm{Cl}(V,\eta)} &(88)\\ &= \eta(v,v)\cdot 1_{\mathrm{Cl}(V,\eta)}, &(89)\end{aligned}$$

*where the first equality holds by the fundamental relation of the Clifford algebra and where the last equality holds by definition of $\mathrm{O}(V,\eta) \ni g$. So the linear map $g : V \to \mathrm{Cl}(V,\eta)$, by the universal property of the Clifford algebra, thus uniquely extends to the algebra homomorphism:*

$$\rho_{\mathrm{Cl}}(g) : \mathrm{Cl}(V,\eta) \to \mathrm{Cl}(V,\eta), \tag{90}$$

*as defined in Equation (87). One can then check the remaining rules for a group action in a straightforward way.*

*More details can be found in (Ruhe et al., 2023b) Appendix D and E.*

## F. Proofs

**Proof F.1 for Proposition 3.2** (Equivariance of the kernel head)**.** Recall the definition of the kernel head:

$$\begin{aligned} H\colon \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}\times c_{\mathrm{in}}} &\to \mathrm{Hom}_{\mathrm{Vec}}\big(\mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}}\big)\\ \mathcal{k} &\mapsto H(\mathcal{k}) = \big[\mathfrak{f} \mapsto H(\mathcal{k})[\mathfrak{f}]\big], \end{aligned} \tag{91}$$

which on each output channel $i \in [c_{\mathrm{out}}]$ and grade component $k = 0,\dots,d$, was given by:

$$H(\mathcal{k})[\mathfrak{f}]_i^{(k)} := \sum_{\substack{j\in[c_{\mathrm{in}}]\\ m,n=0,\dots,d}} w^k_{mn,ij} \cdot \left(\mathcal{k}_{ij}^{(m)} \bullet \mathfrak{f}_j^{(n)}\right)^{(k)},$$

with:

$$\begin{aligned} w^k_{mn,ij} &\in \mathbb{R},\\ \mathcal{k} = [\mathcal{k}_{i,j}]_{\substack{i\in[c_{\mathrm{out}}]\\ j\in[c_{\mathrm{in}}]}} &\in \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}\times c_{\mathrm{in}}},\\ \mathfrak{f} = [\mathfrak{f}_1,\dots,\mathfrak{f}_{c_{\mathrm{in}}}] &\in \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}. \end{aligned}$$

Clearly, $H(\mathcal{k})$ is a $\mathbb{R}$-linear map (in $\mathfrak{f}$). Now let $g \in \mathrm{O}(p,q)$. We are left to check the following equivariance formula:

$$\begin{aligned} H\big(\rho_{\mathrm{Cl}}^{c_{\mathrm{out}}\times c_{\mathrm{in}}}(g)(\mathcal{k})\big) &\overset{?}{=} \rho_{\mathrm{Hom}}(g)\big(H(\mathcal{k})\big) &(92)\\ &:= \rho_{\mathrm{Cl}}^{c_{\mathrm{out}}}(g)\, H(\mathcal{k})\, \rho_{\mathrm{Cl}}^{c_{\mathrm{in}}}(g^{-1}). \end{aligned}$$

We abbreviate

$$s := \rho_{\mathrm{Cl}}^{c_{\mathrm{in}}}(g^{-1})(\mathfrak{f}) \quad \in \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}},$$
$$Q := \rho_{\mathrm{Cl}}^{c_{\mathrm{out}}\times c_{\mathrm{in}}}(g)(\hbar) \in \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}\times c_{\mathrm{in}}}.$$

First note that we have for $j \in [c_{\mathrm{in}}]$:

$$\rho_{\mathrm{Cl}}(g)(s_j) = \mathfrak{f}_j. \tag{93}$$

We then get:

$$\begin{aligned}
&\Big[\rho_{\mathrm{Hom}}(g)\big(H(\hbar)\big)[\mathfrak{f}]\Big]_i^{(k)} \\
&= \Big[\rho_{\mathrm{Cl}}^{c_{\mathrm{out}}}(g)\Big(H(\hbar)\big[\rho_{\mathrm{Cl}}^{c_{\mathrm{in}}}(g^{-1})(\mathfrak{f})\big]\Big)\Big]_i^{(k)} \\
&= \Big[\rho_{\mathrm{Cl}}^{c_{\mathrm{out}}}(g)\big(H(\hbar)\,[s]\,\big)\Big]_i^{(k)} \\
&= \rho_{\mathrm{Cl}}(g)\left(\big[H(\hbar)\,[s]\,\big]_i^{(k)}\right) \\
&= \rho_{\mathrm{Cl}}(g)\left(\sum_{\substack{j\in[c_{\mathrm{in}}]\\ m,n=0,\dots,d}} w_{mn,ij}^k \cdot \left(\hbar_{ij}^{(m)} \bullet s_j^{(n)}\right)^{(k)}\right) \\
&= \sum_{\substack{j\in[c_{\mathrm{in}}]\\ m,n=0,\dots,d}} w_{mn,ij}^k \cdot \left(\big[\rho_{\mathrm{Cl}}(g)(\hbar_{ij})\big]^{(m)} \bullet \big[\rho_{\mathrm{Cl}}(g)(s_j)\big]^{(n)}\right)^{(k)} \\
&= \sum_{\substack{j\in[c_{\mathrm{in}}]\\ m,n=0,\dots,d}} w_{mn,ij}^k \cdot \left(Q_{ij}^{(m)} \bullet \mathfrak{f}_j^{(n)}\right)^{(k)} \\
&= \Big[H(Q)[\mathfrak{f}]\Big]_i^{(k)} \\
&= \Big[H\big(\rho_{\mathrm{Cl}}^{c_{\mathrm{out}}\times c_{\mathrm{in}}}(g)(\hbar)\big)[\mathfrak{f}]\Big]_i^{(k)}.
\end{aligned}$$

Note that we repeatedly made use of the rules in Definition/Theorem 2.14 and Theorem 2.15, i.e. the linearity, composition, multiplicativity and grade preservation of $\rho_{\mathrm{Cl}}(g)$. As this holds for all $m$, $k$ and $\mathfrak{f}$ we get the desired equation,

$$\rho_{\mathrm{Hom}}(g)(H(\hbar)) = H(\rho_{\mathrm{Cl}}^{c_{\mathrm{out}}\times c_{\mathrm{in}}}(g)(\hbar)), \tag{94}$$

which shows the claim. □

## G. Clifford-steerable CNNs on pseudo-Riemannian manifolds

In this section we will assume that the reader is already familiar with the general definitions of differential geometry, which can also be found in Weiler et al. (2021; 2023). We will in this section state the most important results for deep neural networks that process feature fields on *G-structured pseudo-Riemannian manifolds*. These results are direct generalizations from those in Weiler et al. (2023), where they were stated for ($G$-structured) Riemannian manifolds, but which verbatim generalize to ($G$-structured) *pseudo*-Riemannian manifolds if one replaces $\mathrm{O}(d)$ with $\mathrm{O}(p,q)$ everywhere.

Recall, that in this geometric setting a signal $f$ on the manifold $M$ is typically represented by a feature field $f : M \to \mathcal{A}$ of a certain "type", like a scalar field, vector field, tensor field, multi-vector field, etc. Here $f$ assigns to each point $z$ an $n$-dimensional feature $f(z) \in \mathcal{A}_z \cong \mathbb{R}^n$. Formally, $f$ is a global section of a *$G$-associated vector bundle* $\mathcal{A}$ with typical fibre $\mathbb{R}^n$, i.e. $f \in \Gamma(\mathcal{A})$, see Weiler et al. (2023) for details. We can consider $\Gamma(\mathcal{A})$ as the vector space of all vector fields of type $\mathcal{A}$. A deep neural network $F$ on $M$ with $N$ layers can then, as before, be considered as a composition:

$$F : \Gamma(\mathcal{A}_0) \xrightarrow{L_1} \Gamma(\mathcal{A}_1) \xrightarrow{L_2} \Gamma(\mathcal{A}_2) \xrightarrow{L_3} \cdots \xrightarrow{L_N} \Gamma(\mathcal{A}_N), \tag{95}$$

where $L_1, \dots, L_N$ are maps between the vector spaces of vector fields $\Gamma(\mathcal{A}_\ell)$, which are typically linear maps or simple fixed non-linear maps.

For the sake of analysis we can focus on one such linear layer: $L : \Gamma(\mathcal{A}_{\mathrm{in}}) \to \Gamma(\mathcal{A}_{\mathrm{out}})$.

Our goal is to describe the case, where $L$ is an integral operator with an convolution kernel[20] such that: i.) it is well-defined, i.e. independent of the choice of (allowed) local coordinate systems (*covariance*), ii.) we can use the *same* kernel $K$ (not just corresponding ones) in *any* (allowed) local coordinate system (*gauge equivariance*), iii.) it can do *weight sharing* between different locations, meaning that the *same* kernel $K$ will be applied at every location, iv.) input and output transform correspondingly under global transformations (*isometry equivariance*).

The *isometry equivariance* here is the most important property. Our main results in this Appendix will be that isometry equivariance will in fact follow from the first points, see Theorem G.27 and Theorem G.33.

Before we introduce our *Clifford-steerable CNNs* on general *pseudo-Riemannian manifolds* with multi-vector feature fields in Appendix G.2, we first recall the general theory of $G$-steerable CNNs on $G$-structured pseudo-Riemannian manifolds in total analogy to Weiler et al. (2023) in the next section, Appendix G.1.

### G.1. General $G$-steerable CNNs on $G$-structured pseudo-Riemannian manifolds

For the convenience of the reader, we will now recall the most important needed concepts from pseudo-Riemannian geometry in some more generality, but refer to Weiler et al. (2023) for further details and proofs.

We will assume that the curved space $M$ will carry a (non-

[20] Note that a convolution operator $L(f)(u) = \int K(u,v)f(v)\,dv$ can be seen as a continuous analogon to a matrix multiplication. In our theory $K$ will need to depend on only one argument, corresponding to a circulant matrix.

degenerate, possibly indefinite) metric tensor $\eta$ of signature $(p, q)$, $d = p + q$, and will also come with "internal symmetries" encoded by a closed subgroup $G \subseteq \mathrm{GL}(d)$.

**Definition G.1** ($G$-structure)**.** *Let $(M, \eta)$ be pseudo-Riemannian manifold of signature $(p, q)$, $d = p + q$, and $G \leq \mathrm{GL}(d)$ a closed subgroup. A* $G$-structure *on $(M, \eta)$ is a principle $G$-subbundle $\iota : GM \hookrightarrow \mathrm{F}M$ of the frame bundle $\mathrm{F}M$ over $M$. Note that $GM$ is supposed to carry the right $G$-action induced from $\mathrm{F}M$:*

$$\triangleleft : GM \times G \to GM, \quad [e_i]_{i\in[d]} \triangleleft g := \left[\sum_{j\in[d]} e_j\, g_{j,i}\right]_{i\in[d]}, \tag{96}$$

*which thus makes the embedding $\iota$ a $G$-equivariant embedding.*

**Definition G.2** ($G$-structured pseudo-Riemannian manifold)**.** *Let $G \leq \mathrm{GL}(d)$ be closed subgroup. A* $G$-structured pseudo-Riemannian manifold *$(M, G, \eta)$ of signature $(p, q)$ - per definition - consists of a pseudo-Riemannian manifold $(M, \eta)$ of dimension $d = p + q$ with a metric tensor $\eta$ of signature $(p, q)$, and, a* fixed choice *of a* $G$-structure *$\iota : GM \hookrightarrow \mathrm{F}M$ on $M$.*

*We will denote the $G$-structured pseudo-Riemannian manifold with the triple $(M, G, \eta)$ and keep the fixed $G$-structure $\iota : GM \hookrightarrow \mathrm{F}M$ implicit in the notation, as well as the corresponding* $G$-atlas *of local tangent bundle trivializations:*

$$\mathbb{A}_G = \left\{ (\Psi^A, U^A) \,\middle|\, \pi_{\mathrm{T}M}^{-1}(U_A) \xrightarrow[\Psi^A]{\sim} U_A \times \mathbb{R}^d \right\}_{A\in\mathcal{I}} \tag{97}$$

*where $\mathcal{I}$ is an index set and $U^A \subseteq M$ are certain open subsets of $M$.*

**Remark G.3.** *Note that for any given $G \leq \mathrm{GL}(d)$ there might not exists a corresponding $G$-structure $GM$ on $(M, \eta)$ in general. Furthermore, even if it existed it might not be unique. So, when we talk about such a $G$-structure in the following we always make the implicit assumption of its existence and we also fix a specific choice.*

**Definition G.4** (Isometry group of a $G$-structured pseudo-Riemannian manifold)**.** *Let $(M, G, \eta)$ be a $G$-structured pseudo-Riemannian manifold. Its ($G$-structure preserving)* isometry group *is defined to be:*

$$\begin{aligned}&\mathrm{Isom}(M, G, \eta)\\ &:= \big\{\phi : M \xrightarrow{\sim} M \text{ diffeo} \,|\, \forall z \in M,\, v \in \mathrm{T}_z M.\\ &\quad \eta_{\phi(z)}(\phi_{*,\mathrm{T}M}(v), \phi_{*,\mathrm{T}M}(v)) = \eta_z(v, v),\\ &\quad \phi_{*,\mathrm{F}M}(G_z M) = G_{\phi(z)} M\big\}.\end{aligned} \tag{98}$$

The intuition here is that the first condition constrains $\phi$ to be an isometry w.r.t. the metric $\eta$. The second condition constrains $\phi$ to be a symmetry of the $G$-structure, i.e. it maps $G$-frames to $G$-frames.

**Remark G.5** (Isometry group)**.** *Recall that the (usual/full)* isometry group *of a pseudo-Riemannian manifold $(M, \eta)$ is defined as:*

$$\begin{aligned}&\mathrm{Isom}(M, \eta)\\ &:= \big\{\phi : M \xrightarrow{\sim} M \text{ diffeo} \,|\, \forall z \in M,\, v \in \mathrm{T}_z M.\\ &\quad \eta_{\phi(z)}(\phi_{*,\mathrm{T}M}(v), \phi_{*,\mathrm{T}M}(v)) = \eta_z(v, v)\big\}.\end{aligned} \tag{99}$$

*Also note that for a $G$-structured pseudo-Riemannian manifold $(M, G, \eta)$ of signature $(p, q)$ such that $\mathrm{O}(p, q) \leq G$ we have:*

$$\mathrm{Isom}(M, G, \eta) = \mathrm{Isom}(M, \eta). \tag{100}$$

**Definition G.6** ($G$-associated vector bundle)**.** *Let $(M, G, \eta)$ be a $G$-structured pseudo-Riemannian manifold and let $\rho : G \to \mathrm{GL}(n)$ be a left linear representation of $G$. A vector bundle $\mathcal{A}$ over $M$ is called a* $G$-associated vector bundle (with typical fibre $(\mathbb{R}^n, \rho)$) *if there exists a vector bundle isomorphism over $M$ of the form:*

$$\mathcal{A} \xrightarrow{\sim} (GM \times \mathbb{R}^n)/{\sim_\rho} =: GM \times_\rho \mathbb{R}^n, \tag{101}$$

*where the equivalence relation is given as follows:*

$$\begin{aligned}&(e', v') \sim_\rho (e, v)\\ &: \iff \exists g \in G. \quad (e', v') = (e \triangleleft g, \rho(g^{-1})v).\end{aligned} \tag{102}$$

**Definition G.7** (Global sections of a fibre bundle)**.** *Let $\pi_{\mathcal{A}} : \mathcal{A} \to M$ be a fibre bundle over $M$. We denote the set of* global sections *of $\mathcal{A}$ as:*

$$\Gamma(\mathcal{A}) := \{f : M \to \mathcal{A} \,|\, \forall z \in M.\, f(z) \in \mathcal{A}_z\}, \tag{103}$$

*where $\mathcal{A}_z := \pi_{\mathcal{A}}^{-1}(z)$ denotes the* fibre *of $\mathcal{A}$ over $z \in M$.*

**Remark G.8** (Isometry action)**.** *For a $G$-associated vector bundle $\mathcal{A} = GM \times_\rho \mathbb{R}^n$ and $\phi \in \mathrm{Isom}(M, G, \eta)$ we can define the induced $G$-associated vector bundle automorphism $\phi_{*,\mathcal{A}}$ on $\mathcal{A}$ as follows:*

$$\phi_{*,\mathcal{A}} : \mathcal{A} \to \mathcal{A}, \tag{104}$$

$$\phi_{*,\mathcal{A}}(e, v) := (\phi_{*,GM}(e), v). \tag{105}$$

*With this we can define a left action of the group $\mathrm{Isom}(M, G, \eta)$ on the corresponding space of feature fields $\Gamma(\mathcal{A})$ as follows:*

$$\triangleright : \mathrm{Isom}(M, G, \eta) \times \Gamma(\mathcal{A}) \to \Gamma(\mathcal{A}), \tag{106}$$

$$\phi \triangleright f := \phi_{*,\mathcal{A}} \circ f \circ \phi^{-1} : M \to \mathcal{A}. \tag{107}$$

To construct a well-behaved convolution operator on $M$ we first need to introduce the idea of a transporter of feature fields along a curve $\gamma : I \to M$.

**Remark G.9** (Transporter)**.** *A* transporter $\mathfrak{T}_{\mathcal{A}}$ *on the vector bundle* $\mathcal{A}$ *over* $M$ *takes any (sufficiently smooth) curve* $\gamma : I \to M$ *with* $I \subseteq \mathbb{R}$ *some interval and two points* $s, t \in I$, $s \leq t$, *and provides an invertible linear map:*

$$\mathfrak{T}^{s,t}_{\mathcal{A},\gamma} : \mathcal{A}_{\gamma(s)} \xrightarrow{\sim} \mathcal{A}_{\gamma(t)}, \qquad v \mapsto \mathfrak{T}^{s,t}_{\mathcal{A},\gamma}(v). \tag{108}$$

$\mathfrak{T}_{\mathcal{A}}$ *is thought to transport the vector* $v \in \mathcal{A}_{\gamma(s)}$ *at location* $\gamma(s) \in M$ *along the curve* $\gamma$ *to the location* $\gamma(t) \in M$ *and outputs a vector* $\tilde{v} = \mathfrak{T}^{s,t}_{\mathcal{A},\gamma}(v)$ *in* $\mathcal{A}_{\gamma(t)}$.

*For consistency we require that* $\mathfrak{T}_{\mathcal{A}}$ *satisfies the following points for such* $\gamma$*:*

1. *For* $s \in I$ *we get:* $\mathfrak{T}^{s,s}_{\mathcal{A},\gamma} \overset{!}{=} \mathrm{id}_{\mathcal{A}_{\gamma(s)}} : \mathcal{A}_{\gamma(s)} \xrightarrow{\sim} \mathcal{A}_{\gamma(s)}$,
2. *For* $s \leq t \leq u$ *we have:*

$$\mathfrak{T}^{t,u}_{\mathcal{A},\gamma} \circ \mathfrak{T}^{s,t}_{\mathcal{A},\gamma} \overset{!}{=} \mathfrak{T}^{s,u}_{\mathcal{A},\gamma} : \mathcal{A}_{\gamma(s)} \xrightarrow{\sim} \mathcal{A}_{\gamma(u)}. \tag{109}$$

*Furthermore, the dependence on* $s$, $t$ *and* $\gamma$ *shall be "sufficiently smooth" in a certain sense.*

*We call a transporter* $\mathfrak{T}_{\mathrm{T}M}$ *on the tangent bundle* $\mathrm{T}M$ *a* metric transporter *if the map:*

$$\mathfrak{T}^{s,t}_{\mathrm{T}M,\gamma} : (\mathrm{T}_{\gamma(s)}M, \eta_{\gamma(s)}) \xrightarrow{\sim} (\mathrm{T}_{\gamma(t)}M, \eta_{\gamma(t)}) \tag{110}$$

*is always an isometry.*

To construct *transporters* we need to introduce the notion of a *connection* on a vector bundle, which formalized how vector fields change when moving from one point to the next.

**Definition G.10** (Connection)**.** *A* connection *on a vector bundle* $\mathcal{A}$ *over* $M$ *is an* $\mathbb{R}$*-linear map:*

$$\nabla : \Gamma(\mathcal{A}) \to \Gamma(\mathrm{T}^*M \otimes \mathcal{A}), \tag{111}$$

*such that for all* $c : M \to \mathbb{R}$ *and* $f \in \Gamma(\mathcal{A})$ *we have:*

$$\nabla(c \cdot f) = dc \otimes f + c \cdot \nabla(f), \tag{112}$$

*where* $dc \in \Gamma(\mathrm{T}^*M)$ *is the differential of* $c$*.*

A special form of a connection are affine connections, which live on the tangent space.

**Definition G.11** (Affine connection)**.** *An* affine connection *on* $M$ *(or more precisely, on* $\mathrm{T}M$*) is an* $\mathbb{R}$*-bilinear map:*

$$\nabla : \Gamma(\mathrm{T}M) \times \Gamma(\mathrm{T}M) \to \Gamma(\mathrm{T}M), \tag{113}$$
$$(X, Y) \mapsto \nabla_X Y, \tag{114}$$

*such that for all* $c : M \to \mathbb{R}$ *and* $X, Y \in \Gamma(\mathrm{T}M)$ *we have:*

1. $\nabla_{c \cdot X} Y = c \cdot \nabla_X Y$,
2. $\nabla_X(c \cdot Y) = (\partial_X c) \cdot Y + c \cdot \nabla_X Y$,

*where* $\partial_X c$ *denotes the directional derivative of* $c$ *along* $X$.

**Remark G.12.** *Certainly, an affine connection can also be re-written in the usual connection form:*

$$\nabla : \Gamma(\mathrm{T}M) \to \Gamma(\mathrm{T}^*M \otimes \mathrm{T}M). \tag{115}$$

Every connection defines a (parallel) transporter $\mathfrak{T}_{\mathcal{A}}$.

**Definition/Lemma G.13** (Parallel transporter of a connection)**.** *Let* $\nabla$ *be a connection on the vector bundle* $\mathcal{A}$ *over* $M$*. Then* $\nabla$ *defines a (parallel) transporter* $\mathfrak{T}_{\mathcal{A}}$ *for* $\gamma : I = [s, t] \to M$ *as follows:*

$$\mathfrak{T}^{s,t}_{\mathcal{A},\gamma} : \mathcal{A}_{\gamma(s)} \xrightarrow{\sim} \mathcal{A}_{\gamma(t)}, \qquad v \mapsto f(t), \tag{116}$$

*where* $f$ *is the unique vector field* $f \in \Gamma(\gamma^*\mathcal{A})$ *with:*

1. $(\gamma^*\nabla)(f) = 0$,
2. $f(s) = v$,

*which always exists. Here* $\gamma^*$ *denotes the corresponding pullback from* $M$ *to* $I$.

For pseudo-Riemannian manifolds there is a "canonical" choice of a metric connection, the Levi-Cevita connection, which always exists and is uniquely characterized by its two main properties.

**Definition/Theorem G.14** (Fundamental theorem of pseudo-Riemannian geometry: the Levi-Civita connection)**.** *Let* $(M, \eta)$ *be a pseudo-Riemannian manifold. Then there exists a unique affine connection* $\nabla$ *on* $(M, \eta)$ *such that the following two conditions hold for all* $X, Y, Z \in \Gamma(\mathrm{T}M)$*;*

1. *metric preservation:*

$$\partial_Z\left(\eta(X, Y)\right) = \eta(\nabla_Z X, Y) + \eta(X, \nabla_Z Y). \tag{117}$$

2. *torsion-free:*

$$\nabla_X Y - \nabla_Y X = [X, Y], \tag{118}$$

*where* $[X, Y]$ *is the Lie bracket of vector fields.*

*This affine connection is called the* Levi-Cevita connection *of* $(M, \eta)$ *and is denoted as* $\nabla^{\mathrm{LC}}$.

**Remark G.15** (Levi-Civita transporter)**.** *Let* $(M, G, \eta)$ *be a pseudo-Riemannian manifold with Levi-Cevita connection* $\nabla^{\mathrm{LC}}$.

1. *The corresponding Levi-Cevita transporter* $\mathfrak{T}_{\mathrm{T}M}$ *on* $\mathrm{T}M$ *is always a metric transporter, i.e. it always induces (linear) isometries of vector spaces:*

$$\mathfrak{T}^{s,t}_{\mathrm{T}M,\gamma} : (\mathrm{T}_{\gamma(s)}M, \eta_{\gamma(s)}) \xrightarrow{\sim} (\mathrm{T}_{\gamma(t)}M, \eta_{\gamma(t)}). \tag{119}$$

2. *Furthermore, the Levi-Cevita transporter extends to every $G$-associated vector bundle $\mathcal{A}$ as $\mathfrak{T}_{\mathcal{A}}$.*

3. *For every $G$-associated vector bundle $\mathcal{A}$, every curve $\gamma : I \to M$ and $\phi \in \mathrm{Isom}(M, G, \eta)$, the Levi-Cevita transporter $\mathfrak{T}_{\mathcal{A},\gamma}$ always satisfies:*

$$\phi_{*,\mathcal{A}} \circ \mathfrak{T}_{\mathcal{A},\gamma} = \mathfrak{T}_{\mathcal{A},\phi\circ\gamma} \circ \phi_{*,\mathcal{A}}. \tag{120}$$

**Definition G.16** (Geodesics)**.** *Let $M$ be a manifold with affine connection $\nabla$ and $\gamma : I \to M$ a curve. We call $\gamma$ a* geodesic *of $(M, \nabla)$ if for all $t \in I$ we have:*

$$\nabla_{\dot\gamma(t)}\dot\gamma(t) = 0, \tag{121}$$

*i.e. if $\gamma$ runs parallel to itself.*

*For pseudo-Riemannian manifolds $(M, \eta)$ we will typically use the Levi-Cevita connection $\nabla^{\mathrm{LC}}$ to define geodesics.*

**Definition/Lemma G.17** (Pseudo-Riemannian exponential map)**.** *For a manifold $M$ with affine connection $\nabla$, $z \in M$ and $v \in \mathrm{T}_z M$ there exists a unique geodesic $\gamma_{z,v} : I = (-s, s) \to M$ of $(M, \nabla)$ with maximal domain $I$ such that:*

$$\gamma_{z,v}(0) = z, \qquad \dot\gamma_{z,v}(0) = v. \tag{122}$$

*The* $\nabla$-exponential map *at $z \in M$ is then the map:*

$$\exp_z : \mathrm{T}^\circ_z M \to M, \qquad \exp_z(v) := \gamma_{z,v}(1), \tag{123}$$

*with domain:*

$$\mathrm{T}^\circ_z M := \{v \in \mathrm{T}_z M \,|\, \gamma_{z,v}(1) \text{ is defined}\}. \tag{124}$$

*For pseudo-Riemannian manifolds $(M, \eta)$ we will call the exponential map $\exp_z$ defined via the Levi-Cevita connection $\nabla^{\mathrm{LC}}$ the* pseudo-Riemannian exponential map *of $(M, \eta)$ at $z \in M$.*

**Remark G.18.** *For a pseudo-Riemannian manifold $(M, \eta)$ the differential $d\exp_z|_v : \mathrm{T}_v\mathrm{T}_z M \to \mathrm{T}_{\exp_z(v)} M$ is the identity map on $\mathrm{T}_z M$ at $v = 0 \in \mathrm{T}_z M$: $d\exp_z|_{v=0} \overset{!}{=} \mathrm{id}_{\mathrm{T}_z M} : \mathrm{T}_z M = \mathrm{T}_0\mathrm{T}_z M \to \mathrm{T}_{\exp_z(0)} M = \mathrm{T}_z M$.*

*Furthermore, there exist an open subset $U_z \subseteq \mathrm{T}_z M$ such that $0 \in U_z$ and $\exp_z : U_z \to \exp_z(U_z) \subseteq M$ is a diffeomorphism and $\exp_z(U_z) \subseteq M$ is an open subset.*

**Notation G.19.** *For a transporter $\mathfrak{T}_{\mathcal{A}}$ for a vector bundle on $(M, \nabla)$ we abbreviate for $z \in M$ and $v \in \mathrm{T}^\circ_z M$:*

$$\mathfrak{T}_{z,v} := \mathfrak{T}_{\mathcal{A},\gamma^-_{z,v}} : \mathcal{A}_{\exp_z(v)} \xrightarrow{\sim} \mathcal{A}_z, \tag{125}$$

*where $\gamma^-_{z,v} : [0, 1] \to M$ is given by $\gamma^-_{z,v}(t) := \exp_z((1 - t) \cdot v)$.*

**Definition G.20** (Transporter pullback, see Weiler et al. (2023) Def. 12.2.4)**.** *Let $(M, \eta)$ be a pseudo-Riemannian manifold and $\mathcal{A}$ a vector bundle over $M$. Furthermore, let $\exp_z$ denote the pseudo-Riemannian exponential map (based on the Levi-Civita connection) and $\mathfrak{T}_{\mathcal{A}}$ any transporter on $\mathcal{A}$. We then define the* transporter pullback*:*

$$\exp^*_z : \Gamma(\mathcal{A}) \to C(\mathrm{T}^\circ_z M, \mathcal{A}_z), \tag{126}$$

$$\exp^*_z(f)(v) := \mathfrak{T}_{z,v}\big(\underbrace{f(\exp_z(v))}_{\in \mathcal{A}_{\exp_z(v)}}\big) \in \mathcal{A}_z. \tag{127}$$

**Lemma G.21** (See Weiler et al. (2023) Thm. 13.1.4)**.** *For $G$-structured pseudo-Riemannian manifold $(M, G, \eta)$ and $G$-associated vector bundle $\mathcal{A}$, $z \in M$, $\phi \in \mathrm{Isom}(M, G, \eta)$ and $f \in \Gamma(\mathcal{A})$ we have:*

$$\exp^*_z(\phi \rhd f) = \phi_{*,\mathcal{A}} \circ [\exp^*_{\phi^{-1}(z)}(f)] \circ \phi^{-1}_{*,\mathrm{T}M}, \tag{128}$$

*provided the transporter map $\mathfrak{T}_{\mathcal{A}}$ satisfies Equation (120).*

*Weight sharing* for the convolution operator $I$ boils down to the use of a *template convolution kernel* $K$, which is then applied/re-used at every location $z \in M$.

**Definition G.22** (Template convolution kernel)**.** *Let $M$ be a manifold of dimension $d$ and $\mathcal{A}_{\mathrm{in}}$ and $\mathcal{A}_{\mathrm{out}}$ two vector bundles over $M$ with typical fibres $W_{\mathrm{in}}$ and $W_{\mathrm{out}}$, resp. A* template convolution kernel *for $(M, \mathcal{A}_{\mathrm{in}}, \mathcal{A}_{\mathrm{out}})$ is then a (sufficiently smooth, non-linear) map:*

$$K : \mathbb{R}^d \to \mathrm{Hom}_{\mathrm{Vec}}(W_{\mathrm{in}}, W_{\mathrm{out}}), \tag{129}$$

*that is sufficiently decaying when moving away from the origin $0 \in \mathbb{R}^d$ (to make all later constructions, like convolution operations, etc., well-defined).*

The *$G$-gauge equivariance* of a convolution operator $I$ is encoded by the following *$G$-steerability* of the template convolution kernel.

**Definition G.23** ($G$-steerability convolution kernel constraints)**.** *Let $G \leq \mathrm{GL}(d)$ be a closed subgroup and $(M, G, \eta)$ be a $G$-structured pseudo-Riemannian manifold of signature $(p, q)$, $d = p + q$, and $\mathcal{A}_{\mathrm{in}}$ and $\mathcal{A}_{\mathrm{out}}$ two $G$-associated vector bundles with typical fibre $(W_{\mathrm{in}}, \rho_{\mathrm{in}})$ and $(W_{\mathrm{out}}, \rho_{\mathrm{out}})$, resp. A template convolution kernel $K$ for $(M, \mathcal{A}_{\mathrm{in}}, \mathcal{A}_{\mathrm{out}})$:*

$$K : \mathbb{R}^d \to \mathrm{Hom}_{\mathrm{Vec}}(W_{\mathrm{in}}, W_{\mathrm{out}}), \tag{130}$$

*will be called* $G$-steerable *if for all $g \in G$ and $v \in \mathbb{R}^d$ we have:*

$$K(gv) = \frac{1}{|\det g|}\, \rho_{\mathrm{out}}(g)\, K(v)\, \rho_{\mathrm{in}}(g)^{-1} \tag{131}$$

$$=: \rho_{\mathrm{Hom}}(g)(K(v)). \tag{132}$$

**Remark G.24.** *Note that the* $G$-steerability *of $K$ is expressed through Equation (131), while the* $G$-gauge equivariance *of $K$ will, more closely, be expressed through the re-interpretation in Equation (132).*

**Definition G.25** (Convolution operator, see Weiler et al. (2023) Thm. 12.2.9)**.** *Let $(M, G, \eta)$ be a $G$-structured pseudo-Riemannian manifold and $\mathcal{A}_{\text{in}}$ and $\mathcal{A}_{\text{out}}$ two $G$-associated vector bundles over $M$ with typical fibres $(W_{\text{in}}, \rho_{\text{in}})$ and $(W_{\text{out}}, \rho_{\text{out}})$ and $K$ a $G$-*steerable *template convolution kernel, see Equation (131). Let $f_{\text{in}} \in \Gamma(\mathcal{A}_{\text{in}})$ and consider a local trivialization $(\Psi^C, U^C) \in \mathbb{A}_G$ around $z \in U^C \subseteq M$ (which locally trivializes $\mathcal{A}_{\text{in}}$ and $\mathcal{A}_{\text{out}}$). Then we have a well-defined* convolution operator*:*

$$L : \Gamma(\mathcal{A}_{\text{in}}) \to \Gamma(\mathcal{A}_{\text{out}}), \quad f_{\text{in}} \mapsto L(f_{\text{in}}) := f_{\text{out}}, \tag{133}$$

*given by the local formula:*

$$f^C_{\text{out}}(z) := \int_{\mathbb{R}^d} K(v^C) \left[ [\text{Exp}^*_z f_{\text{in}}]^C(v^C) \right] dv^C, \tag{134}$$

*where $\text{Exp}^*_z$ is the transporter pullback from Definition G.20, where $\exp_z$ denotes the pseudo-Riemannian exponential map (based on the Levi-Cevita connection $\nabla^{\text{LC}}$) and $\mathfrak{T}_{\mathcal{A}_{\text{in}}}$ any transporter satisfying Equation (120) (e.g. parallel transport based on $\nabla^{\text{LC}}$).*

**Remark G.26** (Coordinate independence of the convolution operator)**.** *The coordinate independence of the convolution operator $L : \Gamma(\mathcal{A}_{\text{in}}) \to \Gamma(\mathcal{A}_{\text{out}})$ comes from the following* covariance relations *and Equation (131).*

*If we use a different local trivialization $(\Psi^B, U^B) \in \mathbb{A}_G$ in Equation (134) with $z \in U^B \cap U^C$ then there exists a $g \in G$ such that:*

$$v^C = g\, v^B \in \mathbb{R}^d, \tag{135}$$

$$dv^C = |\det g| \cdot dv^B, \tag{136}$$

$$[\text{Exp}^*_z f_{\text{in}}]^C(v^C) = \rho_{\text{in}}(g)\, [\text{Exp}^*_z f_{\text{in}}]^B(v^B) \in W_{\text{in}}, \tag{137}$$

$$f^C_{\text{out}}(z) = \rho_{\text{out}}(g) f^B_{\text{out}}(z) \in W_{\text{out}}. \tag{138}$$

*So, $f_{\text{out}} : M \to \mathcal{A}_{\text{out}}$ is a well-defined global section in $\Gamma(\mathcal{A}_{\text{out}})$.*

We are finally in the place to state the main theorem of this section, stating that every *$G$-steerable* template convolution kernel leads to an *isometry equivariant* convolution operator.

**Theorem G.27** (Isometry equivariance of convolution operator, see Weiler et al. (2023) Thm. 13.2.6)**.** *Let $G \leq \text{GL}(d)$ be closed subgroup and $(M, G, \eta)$ be a $G$-structured pseudo-Riemannian manifold of signature $(p, q)$ with $d = p + q$. Let $\mathcal{A}_{\text{in}}$ and $\mathcal{A}_{\text{out}}$ be two $G$-associated vector bundles with typical fibres $(W_{\text{in}}, \rho_{\text{in}})$ and $(W_{\text{out}}, \rho_{\text{out}})$. Let $K$ be a $G$-*steerable *template convolution kernel, see Equation (131). Consider the corresponding convolution operator $L : \Gamma(\mathcal{A}_{\text{in}}) \to \Gamma(\mathcal{A}_{\text{out}})$ given by Equation (134), where $\exp_z$ denotes the pseudo-Riemannian exponential map (based on the Levi-Cevita connection $\nabla^{\text{LC}}$) and $\mathfrak{T}_{\mathcal{A}_{\text{in}}}$ any transporter satisfying Equation (120) (e.g. parallel transport based on $\nabla^{\text{LC}}$).*

*Then the convolution operator $L : \Gamma(\mathcal{A}_{\text{in}}) \to \Gamma(\mathcal{A}_{\text{out}})$ is* equivariant *w.r.t. the $G$-structure preserving* isometry group $\text{Isom}(M, G, \eta)$*: for every $\phi \in \text{Isom}(M, G, \eta)$ and $f_{\text{in}} \in \Gamma(\mathcal{A}_{\text{in}})$ we have:*

$$L(\phi \triangleright f_{\text{in}}) = \phi \triangleright L(f_{\text{in}}). \tag{139}$$

So the main obstruction for constructing a well-behaved convolution operator $L$ are thus the kernel constraints Equation (131), which are generally notoriously difficult to solve, especially for continuous *non-compact* groups $G$ like $\text{O}(p, q)$.

### G.2. Clifford-steerable CNNs on pseudo-Riemannian manifolds

Let $(M, \eta)$ be a pseudo-Riemannian manifold of signature $(p, q)$ and dimension $d = p + q$.

Then $(M, \eta)$ carries a unique $\text{O}(p, q)$-structure $\text{O}M$ induced by $\eta$. The intuition is that $\text{O}M$ consists of all orthonormal frames w.r.t. $\eta$. In fact, the choice of an $\text{O}(p, q)$-structure on $M$ is equivalent to the choice of a metric $\eta$ of signature $(p, q)$ on $M$. That said, we will now restrict to the structure group $G = \text{O}(p, q)$ everywhere in the following.

We will further restrict to *multi-vector feature fields* $\mathcal{A}_{\text{in}} := \text{Cl}(\text{T}M, \eta)^{c_{\text{in}}}$ and $\mathcal{A}_{\text{out}} := \text{Cl}(\text{T}M, \eta)^{c_{\text{out}}}$, which we first need to formalize properly.

**Definition G.28** (Clifford algebra bundle)**.** *Let $(M, \eta)$ be a pseudo-Riemannian manifold. Then the* Clifford algebra bundle *over $M$ is defined (as a set) as the disjoint union of the Clifford algebras of the corresponding tangent spaces:*

$$\text{Cl}(\text{T}M, \eta) := \bigsqcup_{z \in M} \text{Cl}(\text{T}_z M, \eta_z). \tag{140}$$

*$\text{Cl}(TM, \eta)$ becomes an algebra bundle over $M$ with the standard constructions of local trivialization and bundle projections.*

**Definition G.29** (Othonormal frame bundle of signature $(p, q)$.)**.** *Let $(M, \eta)$ be a pseudo-Riemannian manifold of signature $(p, q)$ and dimension $d = p + q$. Abbreviate for indices $i, j \in [d]$:*

$$\delta^{p,q}_{i,j} := \begin{cases} 0 & \text{if } i \neq j, \\ +1 & \text{if } i = j \in [1, p], \\ -1 & \text{if } i = j \in [p+1, d]. \end{cases} \tag{141}$$

*Then the* orthonormal frame bundle of signature $(p, q)$ *is defined as:*

$$\text{O}M := \bigsqcup_{z \in M} \text{O}_z M, \tag{142}$$

*where we put:*

$$\mathrm{O}_z M := \Big\{ [e_1, \dots, e_d] \;\Big|\; \forall j \in [d].\ e_j \in \mathrm{T}_z M, \tag{143}$$

$$\forall i, j \in [d].\quad \eta_z(e_i, e_j) = \delta_{i,j}^{p,q} \Big\}. \tag{144}$$

*Then* $\mathrm{O}M$ *becomes an* $\mathrm{O}(p,q)$*-structure for* $(M, \eta)$ *together with the standard constructions of local trivialization, bundle projection and right group action:*

$$\triangleleft : \mathrm{O}M \times \mathrm{O}(p,q) \to \mathrm{O}M, \tag{145}$$

$$[e_i]_{i\in[d]} \triangleleft g := \left[ \sum_{j\in[d]} e_j\, g_{j,i} \right]_{i\in[d]}. \tag{146}$$

**Lemma G.30.** *Let* $(M, \eta)$ *be a pseudo-Riemannian manifold of signature* $(p, q)$ *and dimension* $d = p + q$*. We have an algebra bundle isomorphism over* $M$*:*

$$\mathrm{Cl}(\mathrm{T}M, \eta) \cong \mathrm{O}M \times_{\rho_{\mathrm{Cl}}} \mathrm{Cl}(\mathbb{R}^{p,q}), \tag{147}$$

*where* $\rho_{\mathrm{Cl}} : \mathrm{O}(p,q) \to \mathrm{O}_{\mathrm{Alg}}(\mathrm{Cl}(\mathbb{R}^{p,q}), \bar{\eta}^{p,q})$ *is the usual action of the orthogonal group* $\mathrm{O}(p,q)$ *on* $\mathrm{Cl}(\mathbb{R}^{p,q})$ *by rotating all vector components individually. In particular, the Clifford algebra bundle* $\mathrm{Cl}(\mathrm{T}M, \eta)$ *is an* $\mathrm{O}(p,q)$*-associated algebra bundle over* $M$ *with typical fibre* $\mathrm{Cl}(\mathbb{R}^{p,q})$*.*

**Definition G.31** (Multivector fields)**.** *A* multivector field *on* $M$ *is a global section* $f \in \Gamma(\mathrm{Cl}(\mathrm{T}M, \eta)^c)$ *for some* $c \in \mathbb{N}$*, i.e. a map* $f : M \to \mathrm{Cl}(\mathrm{T}M, \eta)^c$ *that assigns to every point* $z \in M$ *a tuple of multivectors:* $f(z) = [f_1(z), \dots, f_c(z)] \in \mathrm{Cl}(\mathrm{T}_z M, \eta_z)^c$*.*

**Remark G.32** (The action of the isometry group on multivector fields)**.** *Let* $\phi \in \mathrm{Isom}(M, \eta)$ *then* $\phi$ *is a diffeomorphic map* $\phi : M \xrightarrow{\sim} M$ *such that for every* $z \in M$ *the differential map is an isometry:*

$$\phi_{*,\mathrm{T}M,z} : (\mathrm{T}_z M, \eta_z) \xrightarrow{\sim} (\mathrm{T}_{\phi(z)}, \eta_{\phi(z)}). \tag{148}$$

*We can now describe the induced map* $\phi_{*,\mathrm{Cl}(\mathrm{T}M,\eta)}$ *via the general construction on associated vector fields, see Remark G.8, with help of the identification Equation (147):*

$$\phi_{*,\mathrm{Cl}(\mathrm{T}M,\eta)} : \mathrm{O}M \times_{\rho_{\mathrm{Cl}}} \mathrm{Cl}(\mathbb{R}^{p,q}) \to \mathrm{O}M \times_{\rho_{\mathrm{Cl}}} \mathrm{Cl}(\mathbb{R}^{p,q}),$$
$$\phi_{*,\mathrm{Cl}(\mathrm{T}M,\eta)}(e, x) = (\phi_{*,\mathrm{F}M}(e), x), \tag{149}$$

*or we can look at the fibres directly,* $z \in M$*:*

$$\phi_{*,\mathrm{Cl}(\mathrm{T}M,\eta),z} : \mathrm{Cl}(\mathrm{T}_z M, \eta_z) \to \mathrm{Cl}(\mathrm{T}_{\phi(z)} M, \eta_{\phi(z)}),$$
$$\phi_{*,\mathrm{Cl}(\mathrm{T}M,\eta),z} \left( \sum_{i\in I} c_i \cdot v_{i,1} \bullet \cdots \bullet v_{i,k_i} \right)$$
$$= \sum_{i\in I} c_i \cdot \phi_{*,\mathrm{T}M,z}(v_{i,1}) \bullet \cdots \bullet \phi_{*,\mathrm{T}M,z}(v_{i,k_i}). \tag{150}$$

*With this we can define a left action of the isometry group* $\mathrm{Isom}(M, \eta)$ *on the corresponding space of multivector fields* $\Gamma(\mathrm{Cl}(\mathrm{T}M, \eta)^c)$ *as follows:*

$$\triangleright : \mathrm{Isom}(M, \eta) \times \Gamma(\mathrm{Cl}(\mathrm{T}M, \eta)^c) \to \Gamma(\mathrm{Cl}(\mathrm{T}M, \eta)^c), \tag{151}$$

$$\phi \triangleright f := \phi_{*,\mathrm{Cl}(\mathrm{T}M,\eta)^c} \circ f \circ \phi^{-1} : M \to \mathrm{Cl}(\mathrm{T}M, \eta)^c. \tag{152}$$

We are now in the position to state the main theorem of this section.

**Theorem G.33** (Clifford-steerable CNNs on pseudo-Riemannian manifolds are gauge and isometry equivariant)**.** *Let* $(M, \eta)$ *be a pseudo-Riemannian manifold of signature* $(p, q)$ *and dimension* $d = p + q$*. We consider* $(M, \eta)$ *to be endowed with the structure group* $G = \mathrm{O}(p,q)$*. Consider multi-vector feature fields* $\mathcal{A}_{\mathrm{in}} = \mathrm{Cl}(\mathrm{T}M, \eta)^{c_{\mathrm{in}}}$ *and* $\mathcal{A}_{\mathrm{out}} = \mathrm{Cl}(\mathrm{T}M, \eta)^{c_{\mathrm{out}}}$ *over* $M$*.*

*Let* $K = H \circ \mathcal{K}$ *be a* Clifford-steerable kernel*, the same template convolution kernel as presented in the main paper in Section 3:*

$$K : \mathbb{R}^{p,q} \to \mathrm{Hom}_{\mathrm{Vec}} \left( \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}} \right), \tag{153}$$

*where* $\mathcal{K} : \mathbb{R}^{p,q} \to \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}}$ *is the* kernel network*, a Clifford group equivariant neural network with* $(c_{\mathrm{in}} \cdot c_{\mathrm{out}})$ *number of Clifford algebra outputs, and, where* $H$ *is the* $\mathrm{O}(p,q)$*-equivariant* kernel head*:*

$$H : \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}} \times c_{\mathrm{in}}} \to \mathrm{Hom}_{\mathrm{Vec}} \left( \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{in}}}, \mathrm{Cl}(\mathbb{R}^{p,q})^{c_{\mathrm{out}}} \right). \tag{154}$$

*Then* $K$ *is automatically* $\mathrm{O}(p,q)$*-steerable, i.e. for* $g \in \mathrm{O}(p,q)$*,* $v \in \mathbb{R}^{p,q}$ *we have*[21]*:*

$$K(gv) = \rho_{\mathrm{Cl}}^{c_{\mathrm{out}}}(g)\, K(v)\, \rho_{\mathrm{Cl}}^{c_{\mathrm{in}}}(g)^{-1}. \tag{155}$$

*Furthermore, the corresponding convolution operator* $L : \Gamma(\mathcal{A}_{\mathrm{in}}) \to \Gamma(\mathcal{A}_{\mathrm{out}})$*, given by Equation (134), is equivariant w.r.t. the full isometry group* $\mathrm{Isom}(M, \eta)$*: for every* $\phi \in \mathrm{Isom}(M, \eta)$ *and* $f_{\mathrm{in}} \in \Gamma(\mathcal{A}_{\mathrm{in}})$ *we have:*

$$L(\phi \triangleright f_{\mathrm{in}}) = \phi \triangleright L(f_{\mathrm{in}}). \tag{156}$$

**Remark G.34.** *A similar theorem to Theorem G.33 can be stated for* orientable *pseudo-Riemannian manifolds* $(M, \eta)$ *and structure group* $G = \mathrm{SO}(p,q)$*, if one reduces the Clifford group equivariant neural network parameterizing the kernel network* $\mathcal{K}$ *to be (only)* $\mathrm{SO}(p,q)$*-equivariant.*

[21] Note that the factor $|\det g|^{-1}$ does not appear here, in contrast to the general formula in Equation (131), because $|\det g| = 1$ anyways for all $g \in \mathrm{O}(p,q)$.