Metric Learning for Clifford Group Equivariant Neural Networks

Riccardo Ali [*1] **Paulina Kulytė** [*1] **Haitz Sáez de Ocáriz Borde** [2] **Pietro Liò** [1]

Editors: *S. Vadgama, E.J. Bekkers, A. Pouplin, S.O. Kaba, H. Lawrence, R. Walters, T. Emerson, H. Kvinge, J.M. Tomczak, S. Jegelka*

Abstract

Clifford Group Equivariant Neural Networks (CGENNs) leverage Clifford algebras and multivectors as an alternative approach to incorporating group equivariance to ensure symmetry constraints in neural representations. In principle, this formulation generalizes to orthogonal groups and preserves equivariance regardless of the metric signature. However, previous works have restricted internal network representations to Euclidean or Minkowski (pseudo-)metrics, handpicked depending on the problem at hand. In this work, we propose an alternative method that enables the metric to be learned in a data-driven fashion, allowing the CGENN network to learn more flexible representations. Specifically, we populate metric matrices fully, ensuring they are symmetric by construction, and leverage eigenvalue decomposition to integrate this additional learnable component into the original CGENN formulation in a principled manner. Additionally, we motivate our method using insights from category theory, which enables us to explain Clifford algebras as a categorical construction and guarantee the mathematical soundness of our approach. We validate our method in various tasks and showcase the advantages of learning more flexible latent metric representations. The code and data are available at https://github.com/rick-ali/Metric-Learning-for-CGENNs

[*]Equal contribution [1]University of Cambridge [2]University of Oxford. Correspondence to: Riccardo Ali <rma55@cam.ac.uk>, Paulina Kulytė <pk556@cam.ac.uk>.

Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at 41st International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024.
Copyright 2024 by the author(s).

1. Introduction

Clifford (or geometric) algebras play a significant role in physics (Baylis, 2004), where they have been used to represent operators such as spinors and Dirac matrices in quantum mechanics, Maxwell's equations in electromagnetism, and Lorentz transformations in relativity. They leverage addition and the geometric product as their two fundamental operations, providing a powerful mathematical language for expressing geometric concepts in a unified manner. In particular, they can describe rotations, reflections, translations, and other geometric transformations succinctly, which frequently arise in physical phenomena.

Recently, Clifford algebras have been introduced to deep learning: Clifford Group Equivariant Neural Networks (CGENNs) (Ruhe et al., 2023) harness the mathematical framework of Clifford algebras to represent data in a way that maintains geometric symmetries and equivariance with respect to several groups such as $O(n)$, $SO(n)$, $E(n)$, and $SE(n)$. However, they currently support only diagonal and fixed metrics to model internal network representations. These typically involve the standard metric on Euclidean space, represented as $Q_E = \text{diag}(1, \ldots, 1)$, and the Minkowski pseudo-metric, $Q_M = \text{diag}(-1, \ldots, 1)$, which, in principle, have inherent physical significance but must be chosen a priori. Ideally, we would like to enable the model to learn as rich internal representations as possible without being constrained to diagonal metric matrices. Thus, inspired by recent work on latent trainable geometries (Borde & Kratsios, 2024; Lu et al., 2023), we advocate for learning the metric in a data-driven fashion via gradient descent.

Our contributions are as follows:

1. We extend CGENNs by integrating learnable metrics, allowing the network to adapt its internal representations dynamically rather than relying on fixed, diagonal metrics.

2. We employ eigenvalue decomposition to transform the full metric matrix representation into an intermediate computationally tractable diagonal form that can easily be integrated into CGENNs, while ensuring that the

input and output data remain consistent across different geometric spaces.

3. We leverage category theory to provide a theoretical foundation for our method. By viewing Clifford Algebras as categorical constructions, we justify the transformations applied within the network, ensuring that our approach is mathematically sound.

We validate our approach through experiments on various tasks, including n-body simulations, signed volume computations, and top-tagging in particle physics.

2. Related Work

The Role of Symmetries in Deep Learning Broadly speaking, a symmetry of a system is a transformation that maintains a particular property of that system, ensuring it remains unchanged or invariant. Symmetries are widespread in real-world problems and data distributions that we aim to model in deep learning. While symmetries could be learned solely from data, integrating invariance and equivariance within artificial neural network architectures concerning specific group actions has emerged as an effective inductive bias, especially in scenarios with limited data (Bronstein et al., 2021).

From Complex Numbers and Quaternions to Multivectors in Neural Representations Initially, the motivation for complex-valued neural representations stemmed from their superior performance within the realm of sequence modeling and Recurrent Neural Networks (RNNs) (Wisdom et al., 2016; Arjovsky et al., 2016). Likewise, other works also motivated them from the perspective of optimization (Nitta, 2002), generalization (Hirose & Yoshida, 2012), and faster learning (Danihelka et al., 2016). Following this line of research, complex-valued neural networks proposed a number of new building blocks that incorporated complex value representations and generalized traditional operations in neural networks such as batch normalization, weight initialization, ReLU activations, convolutions, etc., enabling them to discern intricate patterns that conventional real-valued networks might struggle to capture (Trabelsi et al., 2018). Quaternion based neural networks (Gaudet & Maida, 2018; Parcollet et al., 2019) further expanded on this line of research and introduced three imaginary components for data representation. More recently, the focus has shifted towards leveraging complex representations in the context of geometric deep learning (Bronstein et al., 2021) rather than in sequence modelling. Going beyond complex numbers and quaternions, Clifford algebra can encode richer representations all the way from scalars, vectors, bivectors, and trivectors, to other k-vectors. This capability has been showcased in studies that substitute convolution and Fourier operations in neural PDE surrogates with Clifford counterparts for both 2D and 3D tasks (Brandstetter et al., 2023). Particularly relevant to our work are Clifford Group Equivariant Neural Networks (CGENNs) (Ruhe et al., 2023) which leverage Clifford algebra to model equivariance under orthogonal transformations of the Clifford group. However, current CGENNs are constrained by their support solely for diagonal and fixed metrics, thereby limiting the internal geometric representations they can effectively capture.

Metric Learning enables models to learn how to measure similarity between data points by employing an optimal distance metric tailored to specific learning tasks, rather than relying on predefined, static metrics (Kulis, 2013). Generalizing metrics beyond simple Euclidean embeddings has been extensively studied in the literature by leveraging constant curvature Riemannian manifolds, stereographic projections, and product manifolds (Gu et al., 2018; Ganea et al., 2018; Skopek et al., 2020; Borde et al., 2022; 2023; 2024; Kratsios et al., 2023). However, most of these works, similar to CGENN, pre-define the used metric before optimization. Other recent studies have proposed embeddings with associated differentiable metrics instead, which more closely resemble our proposed approach (Lu et al., 2023; Borde & Kratsios, 2024).

Category Theory in Deep Learning Category theory has recently gained traction in the literature as a unifying language capable of formalizing and extending existing deep learning frameworks. For instance, (Gavranović et al., 2024) use 2-monads to generalize geometric deep learning to non-invertible operations; (Fong et al., 2019) formalize backpropagation as a functor; and (Villani & Schoots, 2023) show that any deep ReLU network has a functionally equivalent three-layer network. Furthermore, category theory has also inspired novel and successful learning schemes: (de Haan et al., 2020) use functors to construct natural graph networks, generalizing permutation equivariance; and (Hansen & Gebhart, 2020; Bodnar et al., 2023; Barbero et al., 2022b;a) augment the message-passing procedure underlying most graph neural networks with geometric information carried by sheaves, a well-known categorical construction in algebraic topology and geometry. Similar sheaf based approaches have also been extended to hypergraphs (Duta et al., 2023).

3. Background

Next, we review the mathematical foundations of our approach, including metric spaces, Clifford algebras, eigenvalue decomposition, and category theory. We also discuss the key components comprising CGENNs, such as algebra embedding layers, generalized linear layers, geometric product layers, normalization layers, and grade-wise nonlinear activation functions.

Inner Product and Metric Spaces An inner product $\langle \cdot, \cdot \rangle$ on a vector space V over a field $K \in \{\mathbb{R}, \mathbb{C}\}$ is a map $V \times V \to K$ such that, for all $x, y \in V$ and $a, b \in K$, satisfies: conjugate-symmetry, $\langle x, y \rangle = \overline{\langle y, x \rangle}$ where $\bar{\cdot}$ denotes complex conjugation; linearity in the first argument, $\langle ax + by, z \rangle = a\langle x, z \rangle + b\langle y, z \rangle$; and positive-definiteness, $\langle x, x \rangle > 0$ for all $x \neq 0$. The inner product is often expressed by the matrix representation Q of a bilinear form satisfying the conditions above. In this case, the inner product takes the form $\langle x, y \rangle = x^T Q y$. For example, if $Q = I$, then this is the usual dot product. A positive definite bilinear form Q, i.e. a matrix whose eigenvalues are all positive, induces a *norm* $|| \cdot ||$ on V by $||x|| = \sqrt{x^T Q x}$. As for the previous example, if $Q = I$, then this is the standard Euclidean norm. Furthermore, a norm induces a *metric* (or *distance*) d on V by setting $d(x, y) = ||x - y||$, making (V, d) a *metric* space. However, we are mainly concerned with metrics arising from matrices Q, and therefore we will refer to such bilinear forms as *metrics*. In this work, we focus on *learning* the metric for CGNNs. During training, our algorithm initializes metrics to $Q = \text{diag}(1, \ldots, 1)$, the standard metric on Euclidean space, and $Q = \text{diag}(-1, \ldots, 1)$, the Minkowski pseudo-metric[1], and iteratively updates them via gradient-descent.

Clifford Algebras A Clifford Algebra, denoted $Cl(V, Q)$, extends the scope of classical algebra and subsumes algebraic structures such as complex numbers and quaternions. It is defined on a vector space V over a field K, together with a quadratic form $Q : V^2 \to K$ that maps pairs of vectors to the field. Its algebra operation, the geometric product, expresses geometric transformations, such as the inner $\langle ., . \rangle$ and wedge product $\wedge$ [2] in algebraic terms. For all $v, w \in V$, the geometric product vw is:

$$vw = \langle v, w \rangle + v \wedge w.$$

The Clifford Algebra $Cl(V, Q)$ is defined as a quotient of the tensor algebra $T(V) = \bigoplus_{k=0}^{\infty} T^k(V)$, where $T^k(V) = V \otimes V \cdots \otimes V$ k times, with the convention $T^0(V) = K$, the underlying field. The quotient defining $Cl(V, Q)$ is $T(V)$ modulo the ideal generated by $v \otimes v - Q(v, v) \cdot 1$, which essentially imposes the equation $v \otimes v = Q(v, v) \cdot 1$ on $T(V)$. Clifford Algebras also have a categorical construction (see Section 4.3) which will prove relevant to our methodology.

With an orthogonal basis $\{e_1, e_2, \ldots, e_n\}$ for V, the algebra is spanned by elements of the form $e_{i_1} e_{i_2} \cdots e_{i_k}$ where $0 \leq k \leq n$ and $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. These span vectors, bivectors, and higher-dimensional constructs, representing directional, area, and volumetric information respectively. Hence, an element of the Clifford Algebra $Cl(V, Q)$ is called a *multivector*, sometimes also referred to as Clifford numbers or multors (Snygg, 2011). The various dimensions (e.g., scalar, vector, bivector, etc.) of a multivector are referred to as *grades*.

The dimension of a Clifford Algebra $Cl(V, Q)$ scales as 2^n for a vector space V of dimension n. For instance, in a two-dimensional space ($n = 2$), an element $x \in Cl(V, Q)$ is represented as:

$$x = x^{(0)} \cdot 1 + x^{(1)} \cdot e_1 + x^{(2)} \cdot e_2 + x^{(12)} \cdot e_1 e_2,$$

with $x^{(i)} \in K$, 1 is the algebra's unit, and e_i are basis vectors of V. The geometric product for multivectors distributes over $+$, analogously to the product of polynomials. Indeed, one could compute the geometric product between multivectors v and w by treating each e_i as a variable in a polynomial and performing polynomial multiplication without assuming commutativity of multiplication of the variables ($e_i e_j \neq e_j e_i$). Additionally, the result should incorporate the relations $v^2 = Q(v, v)$ for all vectors $v \in V$ and $vw = -wv$ for orthogonal vectors $v, w \in V$. Hence, unlike the polynomial's case, the geometric product is not commutative.

The *Clifford Group*, often denoted $Cl^{\times}(V, Q)$ is the set of elements $x \in Cl(V, Q)$ that have an inverse $x^{-1} \in Cl(V, Q)$ with respect to the geometric product, i.e. $xx^{-1} = x^{-1}x = 1$. This group stands out for its capacity to algebraically represent geometric transformations such as rotations, reflections, translations and screws.

Norms in Clifford Algebras While it is clear how to calculate a norm of $v \in V$ with Q, namely $||v|| = \sqrt{v^T Q v}$, we still need to specify its corresponding operation for $x \in Cl(V, Q)$. Let $\beta : Cl(V, Q) \to Cl(V, Q)$ be the *main anti-involution of* $Cl(V, Q)$, the function that takes x and 'inverts' the order of its components:

$$\beta(x) = \beta\left(\sum_{i \in I} c_i \, v_{i,1} \cdot v_{i,k_i}\right) = \sum_{i \in I} c_i \, v_{i,k_i} \cdot v_{i,1}.$$

For example, in 2 dimensions:

$$\begin{aligned}\beta(x) &= \beta(x^{(0)} \cdot 1 + x^{(1)} \cdot e_1 + x^{(2)} \cdot e_2 + x^{(12)} \cdot e_1 e_2) \\ &= x^{(0)} \cdot e_1 e_2 + x^{(1)} \cdot e_2 + x^{(2)} \cdot e_1 + x^{(12)} \cdot 1.\end{aligned}$$

Now, let $x_1, x_2 \in Cl(V, Q)$ and $b : Cl(V, Q) \to Cl(V, Q)$ be the function:

$$b(x_1, x_2) = (\beta(x_1) x_2)^{(0)},$$

where juxtaposition is the geometric product and $(\cdot)^{(0)}$ denotes the projection on the scalar component of the resulting multivector. Finally, by denoting $x^T Q x$ as $Q(x)$ for

[1] The Minkowski metric is technically a pseudo-metric, as it violates positive-definiteness.

[2] $(v \wedge w)_{ij} = (v_i w_j - v_j w_i)$ for all $v, w \in V$. We refer the interested reader to (Darling, 1994).

$x \in V$, we can extend this map to $\overline{Q}(x)$ for $x \in Cl(V,Q)$ by setting $\overline{Q}(x) = b(x,x)$. With this notation, the norm of $x \in Cl(V,Q)$ is $||x|| = (\overline{Q}(x))^{\frac{1}{2}}$. This will be important for the normalisation operations of the CGENN.

Eigenvalue Decomposition In linear algebra, a matrix M in $K^{n\times n}$ over the field K is said to be diagonalizable if there exists an invertible matrix P also in $K^{n\times n}$ such that $M = P^{-1}\Delta P$, where Δ is a diagonal matrix. This process is known as the eigendecomposition of M, because the diagonal entries of Δ represent the eigenvalues of M, and the columns of P correspond to their respective eigenvectors.

One can interpret P as a change of coordinates that represents a vector x in K^n using the coordinates defined by the eigenvectors of M. Indeed, the ith column of an invertible matrix T is precisely Te_i, indicating where T maps the canonical basis element e_i; and because a linear transformation is entirely characterised by its effect on the basis elements, this information is sufficient to fully describe the matrix's action. Thus, Px is simply x expressed in the basis provided by P. Within this basis, the operation of M is expressed by the diagonal matrix Δ: $Mx = \Delta Px = \Delta(Px)$.

Specifically, if $K = \mathbb{R}$ and M is symmetric, the Spectral Theorem ensures that M can be diagonalized with P being orthogonal, that is, $P^{-1} = P^T$. Now, if M functions as a metric, the inner product between two vectors x, y is calculated as follows:

$$x^T My = x^T P^{-1}\Delta Py = x^T P^T \Delta Py = (Px)^T \Delta(Py),$$

indicating that when the vectors are expressed in terms of the basis provided by P, the metric calculations remain consistent as if M were diagonal. This observation is crucial for our implementation since the algorithm used for metric calculations only supports diagonal metrics.

Basics of Category Theory Category theory is the study of *compositionality*, where "objects" can be studied only in their relationships to other objects. For example, to characterise a singleton set, we will not say that it is a set S with only one element, but rather, that there is a unique map from any other set to S.

In the first instance, we can think of a category as a collection of objects of a certain kind, such as sets, groups, and vector spaces, along with maps between them (also called morphisms or arrows) that preserve their structure, such as functions, group homomorphisms, and linear maps. Furthermore, these should be able to be composed sensibly, so that if $f : A \to B$, $g : B \to C$ and $h : C \to D$, where A, B, C, D are objects in a category and f, g, h are maps, then $(h \circ g) \circ f = h \circ (g \circ f)$. A formal definition of a category is given in Appendix B.

Functors, which we will use to motivate and justify our metric learning algorithm, are maps between categories that preserve their structure. Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A *functor* between them, $F : \mathcal{C} \to \mathcal{D}$, is an assignment of objects A in $\mathcal{C}$ to objects $F(A)$ in $\mathcal{D}$, and morphisms $f : A \to B$ in $\mathcal{C}$ to morphisms $F(f) : F(A) \to F(B)$ in $\mathcal{D}$. This assignment should also respect the structure of the categories, i.e., composition: $F(g \circ f) = F(g) \circ F(f)$. A formal treatment of functors is given in Appendix B.

Hence, functors map one category onto another. In particular, they allow us to translate a linear map between vector spaces, a morphism in the category of vector spaces, to a map between Clifford algebras, a morphism in the category of associative algebras, thereby guaranteeing the soundness of our proposed algorithm.

3.1. Clifford Group Equivariant Neural Networks

We now outline the main components of the CGENN, as presented in (Ruhe et al., 2023). The overarching idea of this neural architecture is to accept multivectors as inputs and to process them in each of their grades separately. One could visualise such a computation scheme by picturing k parallel neural networks, each dedicated to one grade, interacting with each other via the geometric product layer. Importantly, each layer needs to be equivariant with respect to any Clifford group transformations.

Algebra Embeddings As the CGENN accepts multivectors as inputs, we need to embed inputs x into the Clifford algebra $Cl(V,Q)$. We do so with the function Embed, whose form is application-specific. For example, if x is a scalar quantity (such as the charge of a particle), it is embedded as a scalar, i.e. $\mathrm{Embed}(x) = x\cdot 1 + 0e_1 + 0e_2 + \cdots + 0e_1e_2\ldots e_n$. Alternatively, if x is a point in V (such as the position of a particle), it is embedded as a 1-dimensional multivector, i.e. $\mathrm{Embed}(x) = 0\cdot 1 + x_1e_1 + \cdots + x_ne_n + 0e_1e_2 + \cdots + 0e_1\ldots e_n$. If x is a volume, it is embedded as $\mathrm{Embed}(x) = 0\cdot 1 + \cdots + xe_1\ldots e_n$, and so on.

Linear Layers The first component of this architecture is the linear layer. For $x_1, \ldots x_l \in Cl(V,Q)$, it is defined as:

$$y^{(k)}_{c_{\text{out}}} = T^{\text{lin}}_{\phi_{c_{\text{out}}}}(x_1,\ldots,x_l)^{(k)} := \sum_{c_{in}=1}^{l} \phi_{c_{\text{out}}c^k_{\text{in}}} x^{(k)}_{c_{\text{in}}},$$

where $\phi_{c_{\text{out}}c^k_{\text{in}}}$ is a learnt scalar parameter depending on the grade k, c_{in} and c_{out} are the input and output channels, and $(\cdot)^{(k)}$ is the projection on the kth grade. Therefore, the map $T^{\text{lin}}_{\phi_{c_{\text{out}}}}(x_1,\ldots,x_l)^{(k)}$ is linear in each grade separately. This is indeed an equivariant layer, as actions of the Clifford group operate separately in each grade.

Geometric Product Layers The Geometric Product layer is the main tool that allows different grades to communicate

in a geometrically principled manner. It is parameterised as follows:

$$P_\phi(x_1, x_2)^{(k)} := \sum_{i=0}^{n} \sum_{j=0}^{n} \phi_{ijk} (x_1^{(i)} x_2^{(j)})^{(k)},$$

where ϕ_{ijk} are learnable scalars for $i, j, k \in \{0, \dots, n\}$. The equivariance of this layer is guaranteed by the fact that the actions of the Clifford group respect grade projections, linearity, scalar multiplication, and products.

Normalisation To ensure numerical stability, CGENNs implement equivariant normalisation layers, defined for $x^{(m)} \in Cl^{(m)}(V, Q)$ as:

$$\text{Norm}(x^{(m)}) := \frac{x^{(m)}}{\sigma(a_m)(\overline{Q}(x^{(m)}) - 1) + 1},$$

where σ is the sigmoid function and $a_m \in \mathbb{R}$ is a learnable parameter. Again, the operation is performed grade-wise and is also equivariant because $\overline{Q}(x^{(m)})$ is, as shown in (Ruhe et al., 2023) (Theorem 3.2). Here the logistic function is used so that the denominator interpolates between 1 (no normalisation) and $\overline{Q}(x^{(m)})$ (geometric normalisation).

Nonlinearities are implemented grade-wise to maintain equivariance and are defined, for $x \in Cl(V, Q))$ as:

$$\text{NonLinear}^{(k)}(x) := \psi(f_k(x)) \cdot x^{(k)},$$

where ψ is any non-linear function $\mathbb{R} \to \mathbb{R}$ and f_k is a linear function of the components of $x^{(k)}$.

4. Method

The following section details the metric learning method proposed in this work, focusing on the initialisation and processing of the metric through its eigenvalue decomposition.

4.1. Metric Initialization

In our method, the transition from a static metric Q, typically initialized as a diagonal matrix to reflect basic geometric properties of the space (e.g., $Q = \text{diag}(1, 1, 1)$), to a learnable metric M involves introducing small perturbations. The process begins with Q, representing the initial geometric configuration. To facilitate learning, a perturbation is added to Q through:

$$\tilde{M} = \frac{1}{2}Q + \epsilon R,$$

where R is a random matrix with the same dimensions as Q, and ϵ controls the amount of initial perturbation. To ensure that M is symmetric, we add the transpose of $\tilde{M}$ to itself, since the sum of two symmetric matrices is symmetric. Hence, we obtain:

$$M = \tilde{M} + \tilde{M}^\top = Q + \epsilon(R + R^\top),$$

since Q is diagonal and $Q = Q^\top$. Therefore, M is equivalent to adding "symmetric" noise, $R + R^\top$, controlled by ϵ, to the initial metric. M is then passed to all downstream layers. The result is a learnable metric that enables the CGENN to dynamically refine its internal geometric representation in a data-driven fashion.

4.2. Learnable Metric via Eigenvalue Decomposition

The original CGENN (Ruhe et al., 2023) implements layers consistent with a Clifford algebra $Cl(V, Q)$ for a fixed bilinear form Q. In particular, this bilinear form is taken as a metric, and it is used in the network's computation of *norms*. This setup only supports diagonal metrics, simplifying the complexity of the space by considering distances that scale linearly along each axis independently. However, real-world data often exhibits correlations that are not captured well by such simplistic assumptions.

The transition from a diagonal to a non-diagonal metric introduces computational and theoretical challenges, particularly in the context of Clifford algebras. In this algebra, metric computations are not as straightforward as in Euclidean space. The norm calculation algorithm used in CGENNs, originally described in (Dorst et al., 2009), only supports diagonal metrics.

This requires us to map a non-diagonal metric to a diagonal one in a geometrically principled way. We achieve this with the metric's eigendecomposition. A metric M can be decomposed into its eigencomponents $M = P^{-1}\Delta P$, where P is a matrix of eigenvectors and Δ is a diagonal matrix of eigenvalues. Therefore, we will use Δ as the diagonal matrix to compute normalisations, and carefully modify the pipeline to make it geometrically meaningful and theoretically sound. We assume that the input is given in any basis $\{e_i\}$ in V. The metric learning procedure can be summarised as follows:

Algorithm 1 Metric Learning for Clifford Group Equivariant Neural Networks – Forward Pass

1: $\Delta, P \leftarrow \text{eigendecomposition}(M)$
2: $x \leftarrow \text{Embed}(x)$
3: $x \leftarrow \overline{P}(x)$
4: $y \leftarrow \text{CliffordNetwork}(x, \Delta, *args)$
5: $y \leftarrow \overline{P^{-1}}(y)$

Next, we proceed to motivate our approach.

Consistency of the Input Across Spaces One could simply adopt Δ as a diagonal metric, but in this case, we would compute norms according *not* to the standard basis of V, but with respect to the basis given by P, introducing inconsistencies. Therefore, we need to express the input in the appropriate basis to make sense of Δ. We do so by applying the change of coordinates P, as reflected in the experiments in Section 5, by replacing the basis vectors e_i with $\xi_i = Pe_i$, which is still a basis as P is invertible. We extend the linear map $x \mapsto Px$ to a map $\overline{P} : Cl(V, M) \to Cl(V, \Delta)$ such that, for $x \in Cl(V, M)$

$$x = x^{(0)} \cdot 1 + x^{(1)} \cdot e_1 + x^{(2)} \cdot e_2 + \cdots + x^{(12\ldots n)} e_1 e_2 \ldots e_n,$$

$$\overline{P}(x) = x^{(0)} \cdot 1 + x^{(1)} \cdot \xi_1 + x^{(2)} \cdot \xi_2 + \cdots + x^{(12\ldots n)} \xi_1 \xi_2 \ldots \xi_n.$$

This is motivated by the categorical construction of Clifford algebras (Section 4.3). Importantly, we preserve the algebra structure when translating multivectors from $Cl(V, M)$ to $Cl(V, \Delta)$:

$$vw = -wv \iff PvPw = -PwPv,$$
$$z^2 = (Pz)^2,$$

for any orthogonal $v, w \in V$ and any $z \in V$. The proof is given in Appendix A. Notably, we preserve consistency with the embedding function outlined in Section 3.1. Operationally, when we deal with inputs $v \in V$, we can transform them with Pv and embed the result, i.e.:

$$\overline{P}(\text{Embed}(v)) = \text{Embed}(Pv),$$

as $Pv = P(v_i e^i) = v_i(Pe^i) = v_i \xi^i$ using the Einstein summation convention $v_i e^i = \sum_i v_i e_i$. Another special case is when the input v is a volume. In this case, we have the following relation:

$$\overline{P}(\text{Embed}(v)) = \text{Embed}(\det(P)v),$$

which we also prove in Appendix A.

Consistency of the Output Across Spaces The function $\overline{P^{-1}}$ is implemented depending on the specific problem and is a crucial step of our learning procedure. It is particularly important when the output of a CGENN has a physical interpretation. Suppose that the learning task is to predict the particle's position in an n-body problem (Section 5.2). Then, if no change of coordinates is applied, the output position is expressed in the basis $\{\xi_i\}$, but the input in the basis $\{e_i\}$. We harmonise the two by applying $\overline{P^{-1}} : Cl(V, \Delta) \to Cl(V, M)$, the extension of P^{-1}, which we define analogously to $\overline{P}$. For any:

$$x = x^{(0)} \cdot 1 + x^{(1)} \cdot \xi_1 + x^{(2)} \cdot \xi_2 + \cdots + x^{(12\ldots n)} \xi_1 \xi_2 \ldots \xi_n,$$

$$\overline{P^{-1}}(x) = x^{(0)} \cdot 1 + x^{(1)} \cdot e_1 + x^{(2)} \cdot e_2 + \ldots$$
$$+ x^{(12\ldots n)} e_1 e_2 \ldots e_n.$$

For example, if the output of the network is:

- a *point* $y \in V$ (e.g. in 5.2), then $\overline{P^{-1}}$ takes the form of $y \mapsto P^{-1}y$, similarly to the classical eigendecomposition.
- a *volume*, i.e. the top grade of the algebra (e.g. in 5.1), then $\overline{P^{-1}}$ maps $y \mapsto \det(P^{-1})y = \frac{1}{\det(P)}y$. This is because $\det(P)$ is precisely how much an $n-$dimensional volume in an $n-$dimensional space is deformed ('stretched') by the linear transformation P. A proof of this is the same as for $\overline{P}$.
- a *scalar* in K, then $\overline{P^{-1}}$ is the identity function, as the basis for scalars is the algebra's identity 1.
- a *probability* (e.g. in 5.3), then the conversion is the identity function, as in this case, the metric serves simply as a computational aid for the models' layers and there is no inherent physical meaning.

Equivariance Linear projections are generally not equivariant. This means that our change of basis via P is also not equivariant, as we allow for non-diagonal metric matrices during optimization. In practice, we initialize the CGENNs to be equivariant, including the input and output transformations. We start with Euclidean or Minkowski pseudo-metrics and use these as an initial prior. However, we relax this condition and break equivariance during later stages of optimization. It is important to note that internally, the latent representations remain equivariant if we disregard the change of basis projections.

4.3. Categorical Construction of Clifford Algebras

Clifford algebras have a categorical construction, which justifies and motivates the construction of Algorithm 1. See Appendix B for background on category theory. A Clifford algebra $Cl(V, Q)$ is a pair (A, i) with A a unital associative algebra over K and $i : V \to Cl(V, Q)$ a linear map with $i(v)^2 = Q(v) \cdot 1_A$ for all $v \in V$ satisfying the following *universal property:* given any unital associative algebra A over K and any linear map:

$$j : V \to A \text{ such that } j(v)^2 = Q(v) \cdot 1_A \text{ for all } v \in V,$$

there is a *unique* algebra homomorphism $f : Cl(V, Q) \to A$ such that:

$$f \circ i = j,$$

i.e. j factors through $Cl(V, Q)$ with i.

Hence, one can indeed see Cl as a functor $\mathbf{VectQ}_K \to \mathbf{AssocA}$ that maps objects (V, Q) to $Cl(V, Q)$ and morphisms (linear maps that preserve quadratic forms) $f : (V, Q) \to (W, R)$ to maps:

$$\overline{f} := Cl(f) : Cl(V, Q) \to Cl(W, R).$$

Functoriality is guaranteed by the universal property of Clifford algebras, which ensures that maps between vector spaces preserving quadratic forms extend *uniquely* to algebra homomorphisms between the respective Clifford algebras. Therefore, the maps $\overline{P}$ and $\overline{P^{-1}}$ in Algorithm 1 are precisely the extensions of the linear maps P and P^{-1} given by the functor Cl. Furthermore, our implementation of $\overline{P^{-1}}$ uses the fact that $\overline{P^{-1}} = \overline{P}^{\,-1}$ as per Section B.1

5. Experiments

In Sections 5.1, 5.2, and 5.3, we conduct experiments with signed volumes, n-body problems, and top tagging, respectively. Additionally, in Section 5.4, we explore the effect of activating metric learning at different stages of training and empirically examine its impact on optimization. In all experiments, we use the default configurations of the baseline CGENN (Ruhe et al., 2023) without any hyperparameter tuning to ensure an equitable comparison.

5.1. O(3) Experiment: Signed Volumes

The signed volumes experiment involves a synthetic dataset of random 3D tetrahedra. The network processes point clouds, aiming to predict covariant scalar quantities (pseudoscalars) under $O(3)$ transformations. The prediction accuracy is measured by the mean-squared error (MSE) between the network's output and the actual signed volumes of the tetrahedra.

We compare CGENNs with a learnable metric against conventional CGENNs. Note that in this experiment, metric learning is initialised from the beginning of training alongside all other model parameters. Other baselines include a normal MLP, an MLP-based version of $E(n)$ Equivariant Graph Neural Networks (E(n)-GNNs) (Satorras et al., 2021) (this architecture leverages artificial neural networks to update positions with scalar multiplication), Vector Neurons (VNs) (Deng et al., 2021) and Geometric Vector Perceptrons (GVPs) (Jing et al., 2020). We train our model for $130,000$ steps, the same as the original CGENN. We calculate the mean and standard deviation with 4 different seeds.

The experimental results, as presented in Table 1, indicate that the learnable metric improves the performance of the original CGENN model. It also outperforms all other baselines. VNs and GVPs perform similarly with an MSE slightly lower than 10^{-1}, while MLPs achieve better performance as the number of data samples increases, reaching MSE losses of around 10^{-3}. Hence, all other baselines result in MSE losses that are orders of magnitude higher than those of CGENNs.

Table 1. Test MSE loss for the synthetic $O(3)$ signed volume experiments.

	MSE	
Model	$n = 1000$	$n = 65536$
CGENN	$1.1 \times 10^{-5} \pm 1.1 \times 10^{-7}$	$1.8 \times 10^{-6} \pm 1.1 \times 10^{-7}$
Ours	$\mathbf{8.3 \times 10^{-8} \pm 1.0 \times 10^{-8}}$	$\mathbf{5.1 \times 10^{-8} \pm 1.0 \times 10^{-8}}$

5.2. E(3) Experiment: n-body

The n-body experiment, as introduced by (Kipf et al., 2018), sets a benchmark for assessing equivariant neural networks in the domain of physical system simulation, a topic further researched by (Han et al., 2022). This experiment challenges neural architectures to predict the three-dimensional paths of n (we use $n = 5$) charged particles, thereby evaluating their ability to accurately model dynamical systems.

We compare our model against the original CGENN, lacking metric learning features, as well as steerable SE(3)-Transformers (Fuchs et al., 2020), Tensor Field Networks (TFNs) (Thomas et al., 2018), Neural Message Passing for Quantum Chemistry Networks (NMPs) (Gilmer et al., 2017), Radial Fields (Köhler et al., 2020), E(n)-GNNs (Satorras et al., 2021), and Steerable $E(3)$ Equivariant Graph Neural Networks (SEGNNs) (Brandstetter et al., 2022).

For this experiment, we train the network for $10,000$ steps over 6 different seeds to obtain the mean and standard deviation. We activate the metric learning at 90% of the training duration. We further analyse the effect of activating metric learning earlier during the training process in Section 5.4.

The findings, as detailed in Table 2, indicate an improvement of our metric-augmented CGENN over the baseline CGENN and other alternative methods. It is worth noting that the CGENN presented in (Ruhe et al., 2023) was trained for 131,072 steps, significantly longer than ours, and nonetheless, we achieve a better performance.

Table 2. Test MSE loss on the n-body experiment for different methods.

Method	MSE ($\times 10^{-3}$)
SE(3)-Tr	24.4
TFN	24.4
NMP	10.7
Radial Field	10.4
E(n)-GNN	7.0
SEGNN	4.3
CGENN	3.9 ± 0.1
Ours	**3.369 ± 0.0436**

5.3. O(1, 3) Experiment: Top Tagging

Jet tagging is a technique in collider physics for categorizing the high-energy jets spawned by particle collisions, such as those detected by the ATLAS detector at CERN (Aad et al., 2008; Kogler et al., 2019). In particular, the experiment presented here, in line with (Ruhe et al., 2023), focuses on jet tagging for top quarks, the heaviest particles within the Standard Model (Incandela et al., 2009). Our evaluation is based on the benchmark provided by (Kasieczka et al., 2019).

We compare CGENN with metric learning against the original CGENN model with a fixed Minkowski pseudo-metric. Due to limited computational resources, we train both models for 30% of the training time reported in (Ruhe et al., 2023) (100,000 steps, which took 32 hours using 4 A100 GPUs and an effective batch size of 32, as opposed to the CGENN in (Ruhe et al., 2023), which was trained for 331,126 steps). The metric for our model is activated 80% into training. Both models are run with default hyperparameters as indicated by (Ruhe et al., 2023). At least under a constrained computational budget, our method outperforms the original CGENN as shown in Table 3, which is in line with all previous experiments as discussed in Section 5.1 and Section 5.2.

Table 3. Test accuracy on the top-tagging experiment.

Model	Accuracy
CGENN	0.8994
Ours	**0.9102**

5.4. Metric Activation

To identify the optimal timing for metric activation in CGENNs, we explore different activation regimes. Our results indicate that the timing of metric activation plays a crucial role in the efficiency of learning. The challenge in metric learning within CGENNs lies in the complexity of optimising a single metric that affects every layer of the network at once. Therefore, early activation might overcomplicate training. Conversely, late activation could function as a fine-tuning stage, making the optimisation problem more tractable.

Table 4 illustrates the outcomes of the n-body experiment, comparing the final MSE loss across three distinct metric activation timings. The results reveal that late activation, occurring at 90% of the training, leads to the lowest loss.

5.5. Reproducibility and Hyperparameters

R is always generated from a uniform distribution with all entries sampled from the interval $[0, 1)$. Depending on the experiment, the values for ϵ and Q are set as follows:

Table 4. n-body ($n = 5$) test MSE across different metric activation timings.

Metric Activation Timing	MSE ($\times 10^{-3}$)
Early (30% of Training)	3.554 ± 0.2051
Mid-Training (60% of Training)	3.489 ± 0.1190
Late (90% of Training)	$\mathbf{3.369 \pm 0.0436}$

$\epsilon = 10^{-3}$ and $Q = \mathrm{diag}(1, 1, 1)$ for the $E(3)$ n-body and $O(3)$ signed volume experiments, and $\epsilon = 10^{-7}$ and $Q = \mathrm{diag}(1, -1, -1, -1)$[3] for $O(1, 3)$ top tagging. The choice of Q is in line with (Ruhe et al., 2023).

6. Conclusion & Future Work

Our research enhances CGENNs by integrating metric learning into the original model in a geometrically meaningful way. Although, as suggested by (Ruhe et al., 2023), the CGENN formulation generalizes to orthogonal groups and preserves equivariance regardless of the metric signature, previous work fixed the metric as a predefined network configuration hyperparameter.

In this work, instead of fixing a metric from the start, we allow the model to learn a non-diagonal metric matrix as part of the optimization process in a data-driven fashion via gradient descent. By leveraging eigenvalue decomposition to perform an internal change of basis, we use the eigenvectors to map different types of data to and from the internal neural representation back to data space, ensuring consistency of both input and output across spaces without requiring explicit modification of the original CGENN network layers, which in principle only support diagonal metrics. Additionally, we employ category theory to motivate the theoretical soundness of the approach by viewing Clifford algebras as categorical constructions.

We validate our method empirically against different tasks, including n-body, signed volume, and top-tagging experiments. We find that enabling metric learning does lead to improved performance. We also analyze the effect of making the metric learnable at different stages of the optimization process, and we empirically find that doing so towards the end of training better guarantees stable training dynamics and generally leads to better model performance.

Future research possibilities include applying this approach to more complex datasets. Similar to (Ruhe et al., 2023), our evaluation did not encompass out-of-distribution tests. Finally, exploring the potential of learning a different metric at each layer may also be an interesting research avenue, but care should be taken in terms of optimization dynamics and to avoid overfitting.

[3] This is equivalent to the Minkowski pseudo-metric discussed in Section 3, but using a different sign convention.

References

Aad, G., Anduaga, X. S., Antonelli, S., Bendel, M., Breiler, B., Castrovillari, F., Civera, J., Del Prete, T., Dova, M. T., Duffin, S., et al. The atlas experiment at the cern large hadron collider. 2008.

Arjovsky, M., Shah, A., and Bengio, Y. Unitary evolution recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1120–1128. JMLR.org, 2016.

Barbero, F., Bodnar, C., de Ocáriz Borde, H. S., and Liò, P. Sheaf attention networks. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022a. URL https://openreview.net/forum?id=LIDvgVjpkZr.

Barbero, F., Bodnar, C., Sáez de Ocáriz Borde, H., Bronstein, M., Veličković, P., and Liò, P. Sheaf neural networks with connection laplacians. In Cloninger, A., Doster, T., Emerson, T., Kaul, M., Ktena, I., Kvinge, H., Miolane, N., Rieck, B., Tymochko, S., and Wolf, G. (eds.), *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*, volume 196 of *Proceedings of Machine Learning Research*, pp. 28–36. PMLR, 25 Feb–22 Jul 2022b. URL https://proceedings.mlr.press/v196/barbero22a.html.

Baylis, W. E. Applications of clifford algebras in physics. 2004. URL https://api.semanticscholar.org/CorpusID:117484492.

Bodnar, C., Giovanni, F. D., Chamberlain, B. P., Liò, P., and Bronstein, M. M. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns, 2023.

Borde, H. S. d. O. and Kratsios, A. Neural snowflakes: Universal latent graph inference via trainable latent geometries. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=djM3WzpOmK.

Borde, H. S. d. O., Kazi, A., Barbero, F., and Lio, P. Latent graph inference using product manifolds. In *The Eleventh International Conference on Learning Representations*, 2022.

Borde, H. S. d. O., Arroyo, A., and Posner, I. Projections of model spaces for latent graph inference. In *ICLR 2023 Workshop on Physics for Machine Learning*, 2023. URL https://openreview.net/forum?id=Eu42jQQgDc.

Borde, H. S. d. O., Arroyo, A., Morales, I., Posner, I., and Dong, X. Neural latent geometry search: Product manifold inference via gromov-hausdorff-informed bayesian optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J., and Welling, M. Geometric and physical quantities improve e(3) equivariant message passing. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_xwr8gOBeV1.

Brandstetter, J., van den Berg, R., Welling, M., and Gupta, J. K. Clifford neural layers for PDE modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=okwxL_c4x84.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

Danihelka, I., Wayne, G., Uria, B., Kalchbrenner, N., and Graves, A. Associative long short-term memory. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1986–1994, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/danihelka16.html.

Darling, R. W. R. *Differential Forms and Connections*. Cambridge University Press, 1994.

de Haan, P., Cohen, T., and Welling, M. Natural graph networks, 2020.

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. Vector neurons: a general framework for so(3)-equivariant networks. *arXiv preprint arXiv:2104.12229*, 2021.

Dorst, L., Fontijne, D., and Mann, S. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009. ISBN 9780080553108.

Duta, I., Cassarà, G., Silvestri, F., and Lió, P. Sheaf hypergraph networks. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 12087–12099. Curran Associates, Inc., 2023.

Fong, B., Spivak, D. I., and Tuyéras, R. Backprop as functor: A compositional perspective on supervised learning, 2019.

Fuchs, F., Worrall, D., Fischer, V., and Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

Ganea, O., Becigneul, G., and Hofmann, T. Hyperbolic neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/dbab2adc8f9d078009ee3fa810bea142-Paper.pdf.

Gaudet, C. J. and Maida, A. S. Deep quaternion networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2018. doi: 10.1109/IJCNN.2018.8489651.

Gavranović, B., Lessard, P., Dudzik, A., von Glehn, T., Araújo, J. G. M., and Veličković, P. Categorical deep learning: An algebraic theory of architectures, 2024.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Gu, A., Sala, F., Gunel, B., and Ré, C. Learning mixed-curvature representations in product spaces. In *International conference on learning representations*, 2018.

Han, J., Rong, Y., Xu, T., and Huang, W. Geometrically equivariant graph neural networks: A survey. *arXiv preprint arXiv:2202.07230*, 2022.

Hansen, J. and Gebhart, T. Sheaf neural networks, 2020. URL https://arxiv.org/abs/2012.06333.

Hirose, A. and Yoshida, S. Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):541–551, 2012. doi: 10.1109/TNNLS.2012.2183613.

Incandela, J. R., Quadt, A., Wagner, W., and Wicke, D. Status and prospects of top-quark physics. *Progress in Particle and Nuclear Physics*, 63(2):239–292, 2009.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

Kasieczka, G., Plehn, T., Thompson, J., and Russel, M. Top quark tagging reference dataset, March 2019. URL https://doi.org/10.5281/zenodo.2603256.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems. In *International conference on machine learning*, pp. 2688–2697. PMLR, 2018.

Kogler, R., Nachman, B., Schmidt, A., Asquith, L., Winkels, E., Campanelli, M., Delitzsch, C., Harris, P., Hinzmann, A., Kar, D., et al. Jet substructure at the large hadron collider. *Reviews of Modern Physics*, 91(4):045003, 2019.

Köhler, J., Klein, L., and Noé, F. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

Kratsios, A., Hong, R., and de Ocáriz Borde, H. S. Capacity bounds for hyperbolic neural network representations of latent tree structures. *arXiv*, 2023.

Kulis, B. Metric learning: A survey. *Found. Trends Mach. Learn.*, 5:287–364, 2013. URL https://api.semanticscholar.org/CorpusID:55485900.

Lu, Y., de Ocáriz Borde, H. S., and Lio, P. AMES: A differentiable embedding space selection framework for latent graph inference. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023. URL https://openreview.net/forum?id=tIrGgIn8jr.

Nitta, T. On the critical points of the complex-valued neural network. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, volume 3, pp. 1099–1103 vol.3, 2002. doi: 10.1109/ICONIP.2002.1202792.

Parcollet, T., Ravanelli, M., Morchid, M., Linarès, G., Trabelsi, C., de Mori, R., and Bengio, Y. Quaternion Recurrent Neural Networks. In *ICLR 2019*, Nouvelle Orléans, United States, May 2019. URL https://hal.science/hal-02107628.

Ruhe, D., Brandstetter, J., and Forré, P. Clifford group equivariant neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=n84bzMrGUD.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/satorras21a.html.

Skopek, O., Ganea, O.-E., and Bécigneul, G. Mixed-curvature variational autoencoders. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1g6xeSKDS.

Snygg, J. A new approach to differential geometry using clifford's geometric algebra. 2011. URL https://api.semanticscholar.org/CorpusID:118201679.

Takeuti, G. and Zaring, W. M. *Axiomatic Set Theory*. Graduate Texts in Mathematics. Springer, New York, NY, 1973 edition, April 1973.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J. F., Mehri, S., Rostamzadeh, N., Bengio, Y., and Pal, C. J. Deep complex networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1T2hmZAb.

Villani, M. J. and Schoots, N. Any deep relu network is shallow, 2023.

Wisdom, S., Powers, T., Hershey, J. R., Roux, J. L., and Atlas, L. Full-capacity unitary recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 4887–4895, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

A. Proofs for Section 4.2

A.1. Proof 1

We wish to prove that, if P is an orthogonal transformation, then the algebra structure is preserved, i.e.

$$vw = -wv \iff PvPw = -PwPv$$
$$z^2 = (Pz)^2$$

for any orthogonal $v, w \in V$ and any $z \in V$. Because $P : (V, M) \to (V, \Delta)$, z^2 is shorthand for $M(z) = z^T M z$ and $(Pz)^2$ for $\Delta(Pz) = (Pz)^T \Delta Pz$.

Direct Proof

[$\Rightarrow$] Assume $vw = -wv$. We show that $PvPw = -PwPv$ by evaluating each side of the equation.

$$PvPw = \langle Pv, Pw\rangle + Pv \wedge Pw = \langle v, w\rangle + \det(P)\, v \wedge w = \det(P)\, v \wedge w$$
$$-PwPv = -(\langle Pw, Pv\rangle + Pw \wedge Pv) = -\langle v, w\rangle - \det(P)\, w \wedge v = \det(P)\, v \wedge w$$

where we use the fact that P is orthogonal, i.e. $\langle Pv, Pw\rangle = \langle v, w\rangle$, v, w are orthogonal, i.e. $\langle v, w\rangle = 0$, and that for any linear transformation A, $Av \wedge Aw = \det(A) v \wedge w$.

[$\Leftarrow$] assume $PvPw = -PwPv$. Then, because v, w are orthogonal, we get the identity

$$\det(P) v \wedge w = -\det(P) w \wedge v \Rightarrow v \wedge w = -w \wedge v$$

Because $0 \neq \det(P) \in \{\pm 1\}$, as P is orthogonal. However, with the fact that $0 = \langle Pv, Pw\rangle = \langle v, w\rangle$ the LHS of the last equality is vw, and the RHS is $-wv$, proving the right-left direction of the 'only if' part of the proposition.

Now, $z^2 = M(z) = z^T M z = (Pz)^T \Delta Pz = \Delta(z) = (Pz)^2$

Categorical Proof

Because $P : (V, M) \to (V, \Delta)$ is a vector space isomorphism that preserves quadratic forms, i.e. $(V, M) \cong (V, \Delta)$, and Cl is a functor, we get $Cl(V, M) \cong Cl(V, \Delta)$ (Appendix B.1) via the unique extension $\overline{P} = Cl(P)$.

A.2. Proof 2

We wish to prove

$$\overline{P}(\text{Embed}(v)) = \text{Embed}(\det(P) v)$$

By direct calculation:

$$\begin{aligned}\overline{P}(\text{Embed}(v)) = \overline{P}(v \cdot e_1 e_2 \dots e_n) &= v \cdot \xi_1 \xi_2 \dots \xi_n \\ &= v \cdot (Pe_1)(Pe_2) \dots (Pe_n) \\ &= \det(P) v \cdot e_1 e_2 \dots e_n = \text{Embed}(\det(P) v)\end{aligned}$$

B. Background on Category Theory

Here, we provide some relevant background on Category Theory to support the discussion in Section 4.3.

B.1. Basics of Category Theory

Category Theory (CT) is, essentially, the study of compositionality: the study of complex systems through their simpler parts. A key difference with set theory is that, for example, we are not allowed to inspect the internal structure of the objects. Rather, we are interested in the relationships between them.

Definition B.1 (Category)**.** A category $\mathcal{C}$ consists of:

- a collection $ob(\mathcal{C})$ of *objects*

- for each $A, B \in ob(\mathcal{C})$ a collection $\mathcal{C}(A, B)$ of *maps* or *arrows* or *morphisms* from A to B
- for each $A, B, C \in ob(\mathcal{C})$, a function

$$\mathcal{C}(A, B) \times \mathcal{C}(B, C) \to \mathcal{C}(A, C)$$
$$(g, f) \mapsto g \circ f$$

  called composition
- for each $A \in \mathcal{C}$ an element $1_A \in \mathcal{C}(A, A)$, the *identity* on A.

Satisfying the following axioms:

- *associativity*: for each $f \in \mathcal{C}(A, B)$, $g \in \mathcal{C}(B, C)$, $h \in \mathcal{C}(C, D)$: $(h \circ g) \circ f = h \circ (g \circ f)$
- *identity laws*: for each $f \in \mathcal{C}(A, B)$, $f \circ 1_A = f = 1_B \circ f$

We are rather informal on the word "collection" when referring to the objects and morphisms, as they are not necessarily sets. The reader interested in foundational issues in Mathematics can consult any Axiomatic Set Theory book, such as (Takeuti & Zaring, 1973).

Examples of Categories To make our definition more concrete, we present some examples categories.

1. The category **Set** is the category with sets as objects and functions between sets as morphisms.
2. The category **Grp** is the category with groups as objects and group homomorphisms as morphisms.
3. The category **Top** is the category with topological spaces as objects and continuous maps as morphisms.
4. Let k be a field. The category $\mathbf{Vect}_k$ is the category with vector spaces over k as objects and linear maps as morphisms.
5. The category **1** has a single object (denoted $*$) and at least one morphisms (the identity).

The main categories we are concerned with in this work are:

- The category $\mathbf{VectQ}_K$ whose objects are vector spaces over a field K equipped with a quadratic form and whose morphisms are linear maps preserving the quadratic forms.
- The category **AssocA** whose objects are unital associative algebras and whose morphisms are algebra homomorphisms.

B.2. Functors

If sets have functions relating them, categories are related by *functors*. Intuitively, a functor maps objects to objects and morphisms to morphisms in a compatible way.

Definition B.2 (Functor)**.** Let $\mathcal{A}, \mathcal{B}$ be categories. A *functor* $F : \mathcal{A} \to \mathcal{B}$ consists of:

- A function $ob(\mathcal{A}) \to ob(\mathcal{B})$, written as $A \mapsto F(A)$
- For each $A, A' \in ob(\mathcal{A})$, a function $\mathcal{A}(A, A') \to \mathcal{B}(F(A), F(A'))$ written as $f \mapsto F(f)$

satisfying the following axioms:

- $F(f' \circ f) = F(f') \circ F(f)$ whenever f, f' are composable in $\mathcal{A}$.
- $F(1_A) = 1_{F(A)}$ whenever $A \in ob(\mathcal{A})$.

Examples of functors To make our definition more concrete, we present some examples of functors.

1. The *forgetful* functors 'disregard' some properties of the objects they are mapping. For example, we can construct a forgetful functor from **Grp** to **Set** by mapping a group to its underlying set, and a group homomorphism to the corresponding map between sets.

2. The *free* functors are 'adjoint' to forgetful functors. They construct objects and maps in the target category with elements that, a priori, do not have those properties. For example, one can construct a free functor from **Set** to **Mon**, the category of monoids: given a set S, its image would be the monoid consisting of formal expressions of words such as xyz^3 (with $x, y, z \in S$). Given a function $f : S \to S'$, its image will be the monoid homomorphism which maps words in the alphabet given by S to words in the corresponding alphabet given by $f(S)$.

3. The fundamental group π is a functor from **Top*** (pointed topological spaces) to **Grp**, mapping topological spaces with a base-point to the corresponding fundamental group at that point, and a continuous map to a group homomorphism as described in any topology textbook.

4. T, the tangent space, is a functor from the category of smooth manifolds with base-point **Man*** to **Vect**, which maps a smooth manifold with base-point to its corresponding tangent space, and a smooth map between manifolds to the corresponding tangent (linear) map: the differential. Therefore, the *chain rule* for differentiable functions is just an instance of the functoriality of the tangent map.

Definition B.3 (Isomorphism)**.** A map $f : A \to B$ in a category $\mathcal{C}$ is an *isomorphism* if there exists a map $g : B \to A$ in $\mathcal{C}$ such that $gf = 1_A$ and $fg = 1_B$. g is called the *inverse* of f and A and B are said to be *isomorphic*, denoted $A \cong B$.

Intuitively, two objects are isomorphic if they are essentially the same, i.e. if they share the same fundamental properties in the context of the category they are in.

If $F : \mathcal{A} \to \mathcal{B}$ is a functor and $A \in ob(\mathcal{A}), B \in ob(\mathcal{B})$ with $A \cong B$, then it is easy to see that $F(A) \cong F(B)$. Further, $F(f^{-1}) = F(f)^{-1}$