Clifford Group Equivariant Neural Network Layers for Protein Structure Prediction

Alberto Pepe∗, Sven Buchholz2, and Joan Lasenby1
1Signal Processing and Communications Lab, University of Cambridge, UK
2Department of Computer Science & Media, Technical University Brandenburg, Germany

Abstract

We employ Clifford Group Equivariant Neural Network (CGENN) layers to predict protein coordinates in a Protein Structure Prediction (PSP) pipeline. PSP is the estimation of the 3D structure of a protein, generally through deep learning architectures. Information about the geometry of the protein chain has been proven to be crucial for accurate predictions of 3D structures. However, this information is usually flattened as machine learning features that are not representative of the geometric nature of the problem. Leveraging recent advances in geometric deep learning, we redesign the 3D projector part of a PSP architecture with the addition of CGENN layers. CGENNs can achieve better generalization and robustness when dealing with data that show rotational or translational invariance such as protein coordinates, which are independent of the chosen reference frame. CGENNs inputs, outputs, weights and biases are objects in the Geometric Algebra of 3D Euclidean space, i.e., $\mathbb{G}_{3,0,0}$, and hence are interpretable from a geometrical perspective. We test 6 approaches to PSP and show that CGENN layers increase the accuracy in term of GDT scores by up to 2.1%, with fewer trainable parameters compared to linear layers and give a clear geometric interpretation of their outputs.

1 Introduction

Geometric deep learning (GDL) focuses on developing models capable of handling data with an underlying geometric structure, including 3D point clouds, graphs, manifolds and molecules [1–3]. Graph Neural Networks (GNNs) are one example of GDL architecture [4], but many other types exist and have been applied in fields such as computer vision, natural language processing, bioinformatics and social network analysis [5–8].

In this paper, we focus on Clifford Group Equivariant Neural Networks (CGENNs) as introduced in [9]. CGENNs allow us to work with objects in 3D space without scalarizing them, thereby preserving their geometrical meaning. CGENNs have been demonstrated to be equivariant maps with respect to the Clifford group. CGENNs have revived the interest in Clifford networks, i.e., networks whose neurons, inputs and outputs are objects in the Clifford Algebra, and have reached state of the art performance in several inherently geometric problems [9–11].

The aim of the paper is to understand the impact of CGENNs on protein structure prediction (PSP) and how they compare to non-geometric machine learning layers. GDL has been widely employed in PSP [12–14]. Moreover, several Clifford and Geometric algebra approaches to protein modelling exist in the literature [15–18]. To the best of our knowledge, this is the first example of layers working in Clifford algebra for a PSP problem.

2 Related Work

2.1 Protein Structure Prediction

Protein structure prediction (PSP) is the estimation, via one or more deep learning (DL) architectures, of the 3D structure of a protein. Input features are generally biochemical quantities related to the protein chain, which are themselves extracted from the amino acid sequence, i.e. the protein’s primary structure. Commonly employed features include the Euclidean distance between amino acids, secondary structure predictions (i.e. the chain’s local folding), coevolutionary information (i.e. the reciprocal evolutionary change in a set of interacting populations) and others [12, 19–24].

Among these features, information about the geometry of the chain has also proven to be particularly relevant. In the literature, geometric information has been encoded in several ways, including through multiple distance maps [20], dihedral angles [22], 3D rigid bodies [12, 23] or geometric algebra (GA)-instantiated features [18, 25].

A major shortcoming of the cited approaches, however, is that this geometric information always needs to be flattened (i.e. scalarized) in order to be fed into and interpreted by DL architectures. GDL can be employed to overcome this issue and preserve the geometric nature of the data. Several examples are given in [13, 14, 25–27], in which proteins are represented as graphs, where nodes correspond to amino acids, and edges represent interactions between them.

Proceedings of the 5th Northern Lights Deep Learning Conference (NLDL), PMLR 233, 2024. © 2024 Alberto Pepe, Sven Buchholz, & Joan Lasenby. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).
(e.g., bonds or contacts). In this paper we aim to implement a GDL pipeline built upon CGENN layers, which explicitly work with objects in Clifford algebra and can model equivariant transformation on them.

2.2 Clifford Algebra

Basics. We will use the term Geometric Algebra (GA) for a real Clifford algebra and follow the geometry-inspired approach pioneered by Hestenes [28]. GA has found many applications in physics, engineering, graphics and more [29, 30].

A GA $\mathcal{G}_{p,q,r}$ has $n = p + q + r$ basis vectors, with $p$ basis vectors that square to $1$, $q$ basis vectors that square to $-1$ and $r$ basis vectors that square to $0$. In this paper, we work with $\mathcal{G}_{3,0,0}$, which is the 3D Euclidean GA. $\mathcal{G}_{3,0,0}$ is fully described by $\{1, e_1, e_2, e_3, e_{12}, e_{13}, e_{23}, e_{123}\}$, in which $\{1\}$ is a scalar and $a$ a 0-blade with grade 0, $\{e_1, e_2, e_3\}$ are vectors or 1-blades, with grade 1, $\{e_{12}, e_{13}, e_{23}\}$ are bivectors or 2-blades, with grade 2 and $\{e_{123}\}$ is a trivector or 3-blade, with grade 3.

Given two GA vectors $a, b$, the geometric product is defined as

$$ab = a \cdot b + a \wedge b,$$

in which $\cdot$ indicates the inner product and $\wedge$ indicates the Grassmann outer product, resulting in a scalar $(a \cdot b)$ plus a bivector $(a \wedge b)$. Multiplication of objects in GA results in multivectors, which are linear combinations of objects of different grades. CGENN layers operate through geometric products on multivectors. A more comprehensive introduction to GA can be found in [31, 32].

**Clifford Algebra Neural Networks.** GA is a framework that allows us to deal elegantly with geometrical objects and transformations. It is hence not surprising that several GDL approaches built upon Clifford and Geometric Algebra exist in the literature. The earliest examples of neural networks working in Clifford Algebra are found in [33–35]. Clifford Algebra neural networks have been recently rediscovered, and have been demonstrated to reach state-of-the-art accuracy in several physics problems of an intrinsically geometric nature [9–11]. In this paper we employ CGENN layers as firstly introduced in [9]. CGENNs (whose mapping is denoted by $\phi$) are networks built upon equivariant layers and operate on multivectors of a Clifford Algebra in any dimension in an $E(n)$-equivariant way, i.e. equivariant over the $n$-dimensional Euclidean space. This means that when an orthogonal transformation $\rho(w)$ is applied to the input data, $x$, the model’s representations correlate, i.e.

$$\phi(\rho(w)(x)) = \rho(w)(\phi(x)).$$

(2)

Operating on the transformed data is the same as operating on the data and then transforming: for physical transformations this would be termed as covariance. The equivariance of CGENN is particularly desirable in PSP problems, since ground truth protein coordinates sit in an arbitrary reference frame which differs for each protein chain. Moreover, CGENN directly transform data in a vector basis, offering a better geometric interpreta-
3 Method

3.1 Architecture

The architecture employed, shown in Figure 1, has been derived from [13]. It is composed of two parts, (i) a Graph Transformer (GrT) and (ii) a 3D projector. The GrT is responsible for encoding biochemical features into graph form for each protein, extracting information from the graph connectivity and obtaining a new node representation. For further details we refer the reader to [13, 25] The 3D projector, on the other hand, is responsible for transforming, or projecting the new nodes in output of the GrT onto 3D Euclidean space.

In this paper we employ two types of CGENN layers, namely (i) multivector linear (MVL) layers and (ii) fully connected geometric product (FCGP) layers, within the 3D projector, and compare them to fully connected linear (L) layers.

Given a set of multivectors \( \{ x_i \}_{i=1}^{C} \), with \( C \) input channels, the output \( z_j \) of the \( j \)-th channel of a MVL layer is given by:

\[
\langle z_j \rangle_k = \sum_{i=1}^{C} \phi_{ijk} \langle x_i \rangle_k,
\]

where \( \langle \cdot \rangle_k \) is the extraction of the grade \( k \) elements in multivectors \( x, z \) and \( \phi_{ijk} \in R \) is a learnable weight.

The FCGP layer, on the other hand, models interaction terms between pairs of multivectors. Given a learnable linear combination of the inputs

\[
y_i = \sum_{p=1}^{C} \beta_{pi} x_p,
\]

the output of the \( j \)-th channel, \( z_j \), obeys

\[
\langle z_j \rangle_k = \sum_{p=1}^{C} \sum_{i=1}^{C} \phi_{ijk} \langle x_i (\beta_{pi} x_p) \rangle_k.
\]

Both \( \phi_{ijk} \in R \) and \( \beta_{pi} \in R \) are learnt, explaining the higher number of parameters for the FCGP layers compared to the MVL layers in Tables 1-2.

The 3D projector models a function \( g \) such that

\[
P = g(Z^{(L)}),
\]

where \( Z^{(L)} \in R^{M \times D} \) is the output of the \( L \)-th layer of the GrT, with \( D \) being the number of nodes, \( P \in R^{M \times 3} \) are the 3D coordinates of the \( M \) \( C_a \) atoms in the protein chain and \( g \) depends on the type of layer chosen.

When the 3D projector is a fully connected layer, as in [13, 25] the function \( g(\cdot) \) is parametrized by a weight matrix \( W_p \in R^{D \times 3} \). When CGENN are employed, on the other hand, some extra steps have to be considered:

- Reshape \( Z^{(L)} \in R^{M \times D} \) into \( Z^{(L)} \in R^{M \times (D/3) \times 3} \), so that we can geometrically interpret the output of the GrT as \( D/3 \) proposals of 3D Euclidean coordinates,
- Embed the reshaped GrT output into the \( G_{3,0,0} \) algebra (i.e. assign 3D coordinates to a basis vector \( \{e_1, e_2, e_3\} \)) so to obtain an input tensor \( X_{in} \in R^{M \times (D/3) \times 8} \) representing a multivector \( x \in G_{3,0,0} \) with 8 real coefficients (1 scalar, 3 vectors, 3 bivectors, 1 trivector) and only the vector part non-zero,
- Downsample the multivector proposals with one or more CGENN layers, operating according to Eq. 3 for the MVL layer or Eq. 5 for the FCGP layer, until obtaining an output tensor \( X_{out} \in R^{M \times 1 \times 3} \),
- Extract grade 1 elements from the obtained multivectors, corresponding to the vector part, i.e. the protein coordinates in 3D Euclidean space \( P \in R^{M \times 1 \times 3} \).

We tested a total of 6 approaches: (a) 1 linear layer (27 nodes to 3); (b) 1 MVL layer (9 3D structures to 1); (c) 1 FCGP layer (9 3D structures to 1); (d) 2 linear layers (27 nodes to 9 to 3); (e) 2 MVL layers (9 3D structures to 3 to 1); (f) 2 FCGP layers (9 3D structures to 3 to 1)
TS scores over the PSICOV150 dataset

HA scores over the PSICOV150 dataset

<table>
<thead>
<tr>
<th>3D projector type</th>
<th>max</th>
<th>median</th>
<th>min</th>
<th>params</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) L layer [25]</td>
<td>11.06</td>
<td>4.27</td>
<td>0.74</td>
<td>150</td>
</tr>
<tr>
<td>(b) MVL layer</td>
<td>13.52</td>
<td>4.49</td>
<td>0.58</td>
<td>142</td>
</tr>
<tr>
<td>(c) FCGP layer</td>
<td>14.14</td>
<td>5.38</td>
<td>0.78</td>
<td>682</td>
</tr>
<tr>
<td>(d) 2 L layers</td>
<td>15.98</td>
<td>5.17</td>
<td>0.74</td>
<td>421</td>
</tr>
<tr>
<td>(e) 2 MVL layers</td>
<td>23.01</td>
<td>6.77</td>
<td>1.42</td>
<td>232</td>
</tr>
<tr>
<td>(f) 2 FCGP layers</td>
<td>33.62</td>
<td>6.79</td>
<td>0.61</td>
<td>1240</td>
</tr>
</tbody>
</table>

Table 2. GDT_HA scores over the PSICOV150 dataset for different 3D projection strategies.

<table>
<thead>
<tr>
<th>3D projector type</th>
<th>max</th>
<th>median</th>
<th>min</th>
<th>params</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) L layer [25]</td>
<td>11.06</td>
<td>4.27</td>
<td>0.74</td>
<td>150</td>
</tr>
<tr>
<td>(b) MVL layer</td>
<td>13.52</td>
<td>4.49</td>
<td>0.58</td>
<td>142</td>
</tr>
<tr>
<td>(c) FCGP layer</td>
<td>14.14</td>
<td>5.38</td>
<td>0.78</td>
<td>682</td>
</tr>
<tr>
<td>(d) 2 L layers</td>
<td>15.98</td>
<td>5.17</td>
<td>0.74</td>
<td>421</td>
</tr>
<tr>
<td>(e) 2 MVL layers</td>
<td>23.01</td>
<td>6.77</td>
<td>1.42</td>
<td>232</td>
</tr>
<tr>
<td>(f) 2 FCGP layers</td>
<td>33.62</td>
<td>6.79</td>
<td>0.61</td>
<td>1240</td>
</tr>
</tbody>
</table>

Table 3. GDT_HA scores over the PSICOV150 dataset for different 3D projection strategies.

3.2 Dataset

We employ the PDNET dataset as presented in [21]. PDNET has \( D = 27 \) features relative to individual amino acids, which can be arranged into nodes \( X \in \mathbb{R}^{M \times D = 27} \), and \( K = 5 \) pairwise features, which can be expressed as edges \( A \in \mathbb{R}^{M \times M \times K = 5} \).

PDNET includes a training set, DEEPCOV, with 3456 protein chains, and a test set, PSICOV150, with 150 chains. PDNET is a subset of the dataset employed in [36]. The train and test sets do not present domain homology, i.e. the proteins in the two sets are not alike due to shared ancestry. The sequence lengths of the protein chains range from 50 to 500 and 50 to 266 amino acids in the training set and test set, respectively. 20% of the training set has been reserved for validation.

Table 1. GDT_TS scores over the PSICOV150 dataset for different 3D projection strategies.

<table>
<thead>
<tr>
<th>3D projector type</th>
<th>max</th>
<th>median</th>
<th>min</th>
<th>params</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) L layer [25]</td>
<td>28.89</td>
<td>15.15</td>
<td>9.07</td>
<td>150</td>
</tr>
<tr>
<td>(b) MVL layer</td>
<td>34.29</td>
<td>18.98</td>
<td>6.59</td>
<td>142</td>
</tr>
<tr>
<td>(c) FCGP layer</td>
<td>33.20</td>
<td>18.13</td>
<td>5.35</td>
<td>682</td>
</tr>
<tr>
<td>(d) 2 L layers</td>
<td>30.06</td>
<td>18.76</td>
<td>6.61</td>
<td>421</td>
</tr>
<tr>
<td>(e) 2 MVL layers</td>
<td>48.01</td>
<td>20.81</td>
<td>7.68</td>
<td>232</td>
</tr>
<tr>
<td>(f) 2 FCGP layers</td>
<td>58.04</td>
<td>20.56</td>
<td>5.45</td>
<td>1240</td>
</tr>
</tbody>
</table>

3.3 Metrics

We assessed the quality of the predictions by measuring three quantities: (i) the mean absolute error (MAE) and (ii) the structural similarity index (SSIM, bounded between 0 and 1) between distance maps \( \hat{D}, \hat{D} \) built upon ground truth and predicted coordinates of the alpha-Carbon atoms of the protein chain \( P, \bar{P} \), respectively; along with the (iii) global distance test (GDT) score between \( P, \bar{P} \) after alignment. The GDT total score (TS) and half size (HA) are defined as:

\[
GDT_{TS} = \frac{p_{<1\AA} + p_{<2\AA} + p_{<4\AA} + p_{<8\AA}}{4},
\]

and

\[
GDT_{HA} = \frac{p_{<0.5\AA} + p_{<1\AA} + p_{<2\AA} + p_{<4\AA}}{4},
\]

respectively, where \( p_{<n\AA} \) indicates the percentage of amino acids in the chains \( P, \bar{P} \) whose Euclidean distance is below \( n \) Å. The distance map for a protein chain of length \( M \) is an \( M \times M \) matrix defined as

\[
D_{ij} = d_{ij}, \text{ in which } d_{ij} \text{ is the Euclidean distance between amino acids } i \text{ and } j \text{ expressed in Å}.
\]

3.4 Training details

The loss function is identical to the one employed [36], and is measured over distance maps \( \hat{D}, \hat{D} \), built from ground truth and predicted 3D coordinates \( P, \bar{P} \). This is used because distances are independent of the reference frame.

The total loss to minimize is equal to

\[
\mathcal{L} = \mathcal{L}_{MAE} + \mathcal{L}_{SSIM},
\]

in which the first term minimizes the MAE between \( D \) (the ground truth distance map) and \( \hat{D} \), as

\[
\mathcal{L}_{MAE} = MAE(D, \hat{D}).
\]

The second term maximizes the SSIM between \( D \) and \( \hat{D} \)

\[
\mathcal{L}_{SSIM} = \alpha \left( 1 - SSIM(D, \hat{D}) \right),
\]

where \( \alpha = 20 \) is a scalar found empirically to give them an equal weight in the total loss.

We chose Adam as optimizer, with starting learning rate \( \eta_0 = 1 \times 10^{-2} \) and decay rate every 2 epochs of \( \gamma = 0.9 \). We chose a batch size of \( B = 32 \) and implemented early stopping with patience \( P = 4 \) and a tolerance of \( \Delta = 0.1 \). The
footnote reference import as citation
deep learning on graphs and manifolds using mixture model cnns”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5115–5124.


