Clifford Group Equivariant Neural Network Layers for Protein Structure Prediction

Alberto Pepe*[1], Sven Buchholz[2], and Joan Lasenby[1]

[1]Signal Processing and Communications Lab, University of Cambridge, UK
[2]Department 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. $\mathcal{G}_{3,0,0}$, 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 (o.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.

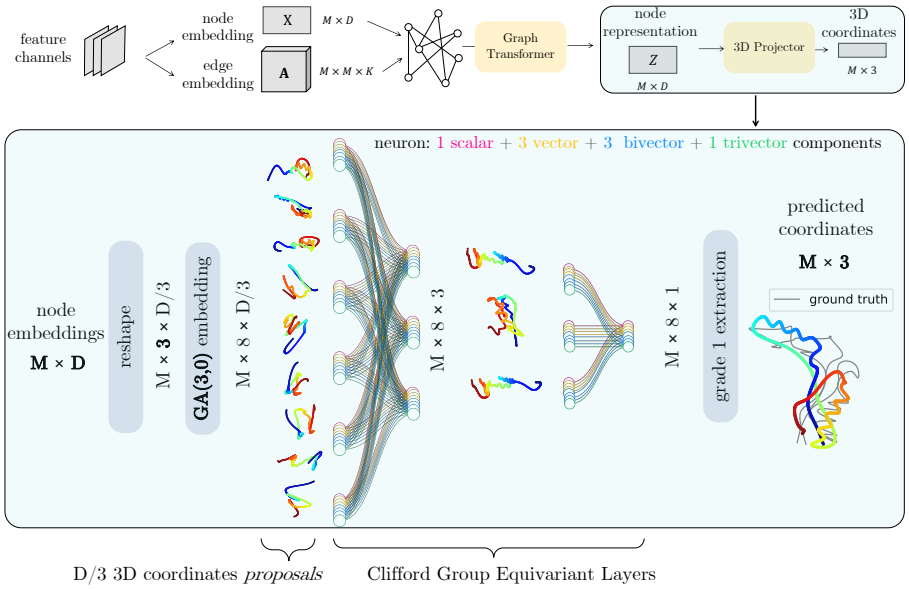


Figure 1. CGENN layers employed within the 3D Projector in a Protein Structure Prediction pipeline. A protein with M amino acids can be represented as a graph with D nodes and K edges. The Graph Transformer extracts a new node representation $M \times D$, while the 3D Projector maps the representation onto 3D space to obtain $M \times 3$ 3D coordinates. In CGENNs, neurons, weights and biases are multivectors, in our case in $\mathcal{G}_{3,0,0}$, whose vector part can be interpreted geometrically.

(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 or 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 $\mathbf{a}, \mathbf{b}$, the geometric product is defined as

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}, \tag{1}$$

in which $\cdot$ indicates the inner product and $\wedge$ indicates the Grassmann outer product, resulting in a scalar $(\mathbf{a} \cdot \mathbf{b})$ plus a bivector $(\mathbf{a} \wedge \mathbf{b})$. 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 tranformations. 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 ϕ) 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 *corotate*, i.e.

$$\phi(\rho(w)(x)) = \rho(w)(\phi(x)). \tag{2}$$

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 CGENNs is particularly desirable in PSP problems, since ground truth protein coordinates sit in an arbitrary reference frame which differs for each protein chain. Moreover, CGENNs directly transform data in a vector basis, offering a better geometric interpreta-

tion of the network's intermediate outputs in terms of folding.

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 connectected 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:

$$\langle z_j \rangle_k = \sum_{i=1}^{C} \phi_{ijk} \langle x_i \rangle_k, \tag{3}$$

in which $\langle \cdot \rangle_k$ 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

$$y_i = \sum_{p=1}^{C} \beta_{pi} x_p, \tag{4}$$

the output of the j-th channel, z_j, obeys

$$\langle z_j \rangle_k = \sum_{i=1}^{C} \sum_{p=1}^{C} \phi_{ijk} \langle x_i (\beta_{pi} x_p) \rangle_k. \tag{5}$$

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

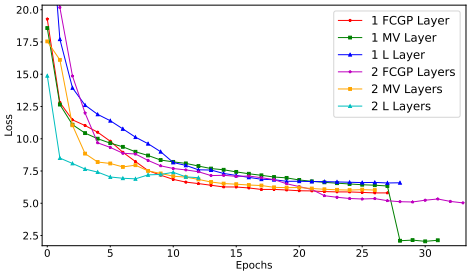
$$P = g(Z^{(L)}), \tag{6}$$

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_α atoms in the protein chain and g depends on the type of layer chosen.

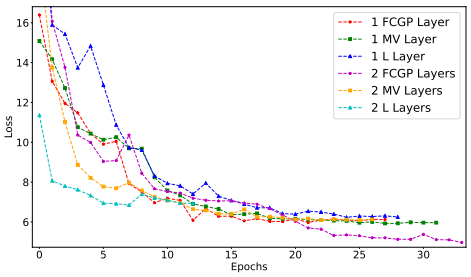
When the 3D projector is a fully connected layer, as in [13, 25] the function $g(\cdot)$ is parametrized by a weight matrix $W_P \in R^{D \times 3}$. When CGENNs are employed, on the other hand, some extra steps have to be considered:

- 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 $\mathcal{G}_{3,0,0}$ 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 $x \in \mathcal{G}_{3,0,0}$ 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 $X_{out} \in R^{M \times 1 \times 8}$,
- 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 approches: (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)

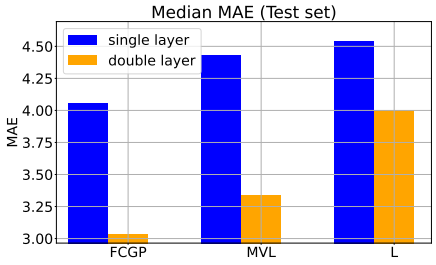


(a) Train Loss

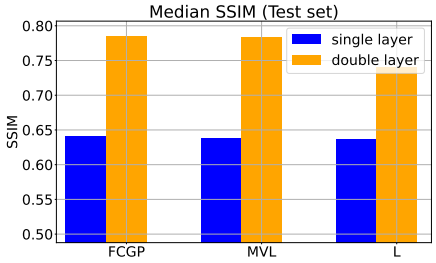


(b) Validation Loss

Figure 2. Losses for the 6 different approaches to the 3D projector.



(a) Mean Absolute Error (↓)



(b) Structural Similarity Index Measure (↑)

Figure 3. (a) MAE and (b) SSIM distributions over the PSICOV150 dataset.

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 $\mathbf{X} \in R^{M \times D=27}$, and $K = 5$ pairwise features, which can be expressed as edges $\mathbf{A} \in R^{M \times M \times K=5}$.

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.

3D projector type	max	median	min	params
(a) L layer [25]	28.69	15.15	9.07	150
(b) MVL layer	34.29	16.98	6.59	142
(c) FCGP layer	33.20	18.13	5.35	682
(d) 2 L layers	36.06	18.76	6.61	421
(e) 2 MVL layers	48.01	20.81	7.68	232
(f) 2 FCGP layers	58.04	20.56	5.45	1240

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 $\mathbf{D}, \hat{\mathbf{D}}$ built upon ground truth and predicted coordinates of the alpha-Carbon atoms of the protein chain $P, \hat{P}$, respectively; along with the (iii) global distance test (GDT) score between $P, \hat{P}$ after alignment. The GDT total score (TS) and half size (HA) are defined as:

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

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

$$\text{GDT\_TS} = \frac{p_{<1\text{Å}} + p_{<2\text{Å}} + p_{<4\text{Å}} + p_{<8\text{Å}}}{4}, \quad (7)$$

and

$$\text{GDT\_HA} = \frac{p_{<0.5\text{Å}} + p_{<1\text{Å}} + p_{<2\text{Å}} + p_{<4\text{Å}}}{4}, \quad (8)$$

respectively, where $p_{<n\text{Å}}$ indicates the percentage of amino acids in the chains $P, \hat{P}$ 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 $\mathbf{D}_{ij} = d_{ij}$, in which d_{ij} is the Euclidean distance between amino acids i and j expressed in Å.

3.4 Training details

The loss function is identical to the one employed [18], and is measured over distance maps $\mathbf{D}, \hat{\mathbf{D}}$, built from ground truth and predicted 3D coordinates $P, \hat{P}$. This is used because distances are independent of the reference frame.

The total loss to minimize is equal to

$$\mathcal{L} = \mathcal{L}_{MAE} + \mathcal{L}_{SSIM}, \quad (9)$$

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

$$\mathcal{L}_{MAE} = \text{MAE}(\mathbf{D}, \hat{\mathbf{D}}). \quad (10)$$

The second term *maximizes* the SSIM between $\mathbf{D}$ and $\hat{\mathbf{D}}$

$$\mathcal{L}_{SSIM} = \alpha \left(1 - SSIM(\mathbf{D}, \hat{\mathbf{D}})\right), \quad (11)$$

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

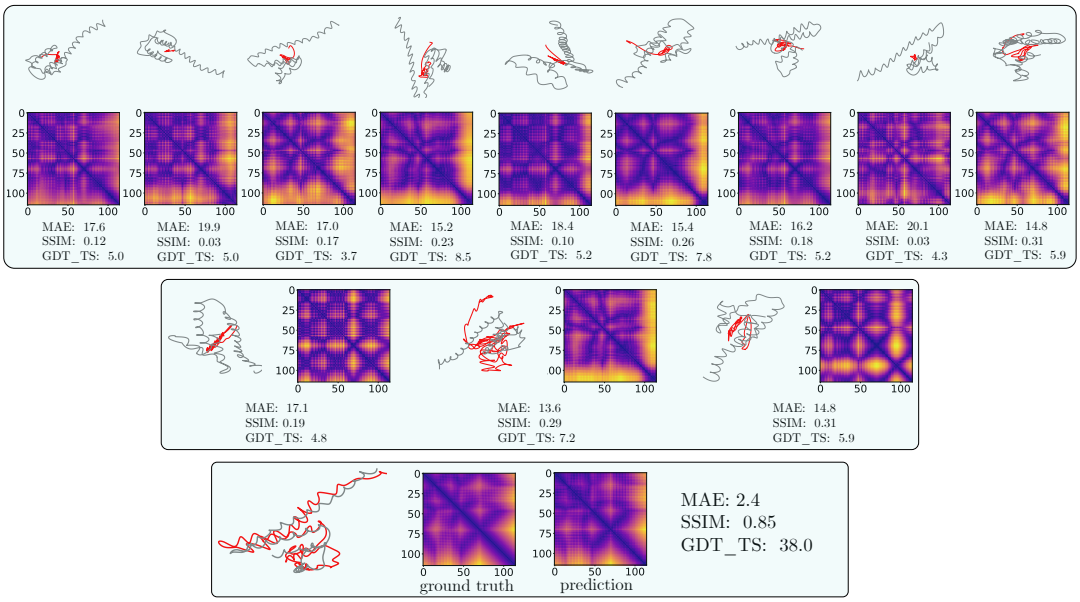


Figure 4. Input and outputs of the CGENN layers for the prediction of the 3D structure of protein 2ehwA for approach (e). The outputs (in red) are *also* 3D structures, with sensible distance maps close to ground truth (in grey). Note how the same ground truth appears rotated due to the alignment procedure.

script has been written as a notebook on Google Colaboratory, run on an NVIDIA Tesla A100 GPU. The operations in GA have been implemented via [37] protein data were handled via Biopython [38]. The GrT was derived from [13] and the CGENN layers from [10]. The code is available at https://github.com/albertomariapepe/CGENN-PSP.

4 Results

Results are summarized in Tables 1-2. Compared to linear layers, CGENNs increase the median GDT_TS by up to 3% and 2.1% with one and two layers, respectively. Similarly, the median GDT_HA shows increases of 1.1% and of 1.6% with one and two layers, respectively. Note how for the MVL layers, the improvement is given despite a reduction in the number of trainable parameters, which is quite significant when the 3D projector is composed by 2 layers. This trend is mirrored in the loss profiles: train and validation losses for the 6 different approaches to the 3D projector are presented in Figures 2(a)-2(b), respectively. In both figures, CGENN layers loss reaches lower minima than linear layers, despite a slower convergence. The distributions over the test set of MAE and SSIM measured between $\mathbf{D}, \hat{\mathbf{D}}$ are shown in Figures 3(a)-3(b), respectively. The lowest MAE is obtained via approach (e), followed by (f) and (c), all CGENN layers. The highest median MAE, on the other hand, is achieved by approach (a), i.e. the single linear layer of [13, 25]. Similarly, the highest median SSIM over the test set is obtained via approaches (e) and (f), while the lowest via approach (a).

We believe that the main advantage of our approach lies in its geometric interpretability. In Figure 4 an example of input and outputs of the CGENN layers is given for protein 2ehwA. The $D/3$ coordinate proposals for case (e), in our case 9, are fed as input to the MVL layer (top). Note how the 3D coordinates, despite being still far from the correct folding, can be used to construct distance maps which are already close to ground truth, and hence fully interpretable as protein chains. The 3 outputs from the 1st MVL layer are shown in the middle section of Figure 4: the MAE decreases, while SSIM and GDT scores increase. The 3D structures "unfold". Finally, the output of the last layer is shown in the bottom section of Figure 4. It is not possible to have the same interpretability with linear layers.

5 Conclusion

We applied CGENN layers to predict 3D structures of proteins. We showed how CGENN layers, which are E(3)-equivariant and explicitly work in 3D Euclidean space via $\mathcal{G}_{3,0,0}$, improve the quality of the predictions, reaching lower losses during training, boosting the accuracy measured as GDT scores of 2% compared to linear layers and offering a geometric interpretation of intermediate layers' ouputs as 3D protein structures.

References

[1] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. "Geometric

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.

[2] K. Atz, F. Grisoni, and G. Schneider. "Geometric deep learning on molecular representations". In: *Nature Machine Intelligence* 3.12 (2021), pp. 1023–1032. DOI: 10.1038/s42256-021-00418-8.

[3] R. Wiersma, A. Nasikun, E. Eisemann, and K. Hildebrandt. "Deltaconv: anisotropic operators for geometric deep learning on point clouds". In: *ACM Transactions on Graphics (TOG)* 41.4 (2022), pp. 1–10. DOI: 10.1145/3528223.3530166.

[4] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "The graph neural network model". In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[5] T. S. Cohen and M. Welling. "Steerable cnns". In: *arXiv preprint arXiv:1612.08498* (2016).

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[7] W. Cao, Z. Yan, Z. He, and Z. He. "A comprehensive survey on geometric deep learning". In: *IEEE Access* 8 (2020), pp. 35929–35949. DOI: 10.1109/ACCESS.2020.2975067.

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

[9] D. Ruhe, J. Brandstetter, and P. Forré. "Clifford group equivariant neural networks". In: *arXiv preprint arXiv:2305.11141* (2023).

[10] J. Brandstetter, R. v. d. Berg, M. Welling, and J. K. Gupta. "Clifford neural layers for PDE modeling". In: *arXiv preprint arXiv:2209.04934* (2022).

[11] D. Ruhe, J. K. Gupta, S. de Keninck, M. Welling, and J. Brandstetter. "Geometric clifford algebra networks". In: *arXiv preprint arXiv:2302.06594* (2023).

[12] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, K. Tunyasuvunakool, O. Ronneberger, R. Bates, A. Žídek, A. Bridgland, et al. "AlphaFold 2". In: *In Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book* (2020).

[13] A. Costa, M. Ponnapati, J. M. Jacobson, and P. Chatterjee. "Distillation of MSA embeddings to folded protein structures with graph transformers". In: *bioRxiv* (2021), pp. 2021–06.

[14] V. Gligorijević, P. D. Renfrew, T. Kosciolek, J. K. Leman, D. Berenberg, T. Vatanen, C. Chandler, B. C. Taylor, I. M. Fisk, H. Vlamakis, et al. "Structure-based protein function prediction using graph convolutional networks". In: *Nature communications* 12.1 (2021), p. 3168. DOI: 10.1038/s41467-021-23303-9.

[15] P. Chys and P. Chacón. "Spinor product computations for protein conformations". In: *Journal of Computational Chemistry* 33.21 (2012), pp. 1717–1729. DOI: 10.1002/jcc.23002.

[16] P. Chys and P. Chacón. "Random coordinate descent with spinor-matrices and geometric filters for efficient loop closure". In: *Journal of chemical theory and computation* 9.3 (2013), pp. 1821–1829. DOI: 10.1021/ct300977f.

[17] C. Lavor and R. Alves. "Oriented conformal geometric algebra and the molecular distance geometry problem". In: *Advances in Applied Clifford Algebras* 29 (2019), pp. 1–15. DOI: 10.1007/s00006-018-0925-0.

[18] A. Pepe, J. Lasenby, and P. Chacon. *Geometric Algebra Models of Proteins for Three-Dimensional Structure Prediction*. Tech. rep. EasyChair, 2022.

[19] J. Peng and J. Xu. "RaptorX: exploiting structure information for protein alignment by statistical inference". In: *Proteins: Structure, Function, and Bioinformatics* 79.S10 (2011), pp. 161–171. DOI: 10.1002/prot.23175.

[20] J. Xu. "Distance-based protein folding powered by deep learning". In: *Proceedings of the National Academy of Sciences* 116.34 (2019), pp. 16856–16865. DOI: 10.1073/pnas.1821309116.

[21] B. Adhikari. "A fully open-source framework for deep learning protein real-valued distances". In: *Scientific reports* 10.1 (2020), p. 13374. DOI: 10.1038/s41598-020-70181-0.

[22] J. Yang, I. Anishchenko, H. Park, Z. Peng, S. Ovchinnikov, and D. Baker. "Improved protein structure prediction using predicted interresidue orientations". In: *Proceedings of the National Academy of Sciences* 117.3 (2020), pp. 1496–1503. DOI: 10.1073/pnas.1914677117.

[23] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, et al. "Accurate prediction of protein structures and interactions using a three-track neural network". In: *Science* 373.6557 (2021), pp. 871–876. DOI: 10.1126/science.abj8754.

[24] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2.

[25] A. Pepe and J. Lasenby. "Modeling orientational features via geometric algebra for 3D protein coordinates prediction". In: *Mathematical Methods in the Applied Sciences* n/a.n/a (). DOI: 10.1002/mma.9608.

[26] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur. "Protein interface prediction using graph convolutional networks". In: *Advances in neural information processing systems* 30 (2017).

[27] G. S.-P. Lemieux, E. Paquet, H. L. Viktor, and W. Michalowski. "Geometric deep learning for protein–protein interaction predictions". In: *IEEE Access* 10 (2022), pp. 90045–90055. DOI: 10.1109/ACCESS.2022.3201543.

[28] D. Hestenes. *Space-time algebra*. Springer, 2015. DOI: 10.1007/978-3-319-18413-5.

[29] A. Lasenby. "Recent applications of conformal geometric algebra". In: *Computer Algebra and Geometric Algebra with Applications*. Springer, 2004, pp. 298–328.

[30] C. G. Gunn and S. De Keninck. "Geometric algebra and computer graphics". In: *ACM SIGGRAPH 2019 Courses*. 2019, pp. 1–140.

[31] C. Doran and A. Lasenby. *Geometric algebra for physicists*. Cambridge University Press, 2003. DOI: 10.1017/CBO9780511807497.

[32] L. Dorst and J. Lasenby. *Guide to geometric algebra in practice*. Springer, 2011.

[33] J. Pearson and D. Bisset. "Neural networks in the Clifford domain". In: *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. Vol. 3. IEEE. 1994, pp. 1465–1469. DOI: 10.1109/ICNN.1994.374502.

[34] E. B. Corrochano, S. Buchholz, and G. Sommer. "Selforganizing Clifford neural network". In: *Proceedings of International Conference on Neural Networks (ICNN'96)*. Vol. 1. IEEE. 1996, pp. 120–125. DOI: 10.1109/ICNN.1996.548877.

[35] S. Buchholz and G. Sommer. "Learning geometric transformations with Clifford neurons". In: *International Workshop on Algebraic Frames for the Perception-Action Cycle*. Springer. 2000, pp. 144–153.

[36] D. T. Jones and S. M. Kandathil. "High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features". In: *Bioinformatics* 34.19 (2018), pp. 3308–3315. DOI: 10.1093/bioinformatics/bty341.

[37] H. Hadfield, E. Wieser, A. Arsenovic, and R. Kern. "The Pygae Team: pygae/clifford: v1.3.1 (2020)". In: 1453978 (). DOI: 10.5281/zenodo.

[38] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, et al. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". In: *Bioinformatics* 25.11 (2009), p. 1422. DOI: 10.1093/bioinformatics/btp163.