Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

ARTICLE TYPE

Modeling orientational features via Geometric Algebra for 3D protein coordinates prediction

Alberto Pepe* | Joan Lasenby

Signal Processing & Communications Lab, Engineering Department, University of Cambridge, UK

Correspondence
Alberto Pepe, Signal Processing & Communications Lab, CUED, Trumpington Street, Cambridge, UK. Email: ap2219@cam.ac.uk

Present Address
CUED, Trumpington Street, Cambridge, UK

Abstract

By protein structure prediction (PSP) we refer to the prediction of the 3-dimensional (3D) folding of a protein, known as tertiary structure, starting from its amino acid sequence, known as primary structure. The state-of-the-art in PSP is currently achieved by complex deep learning pipelines that require several input features extracted from amino acid sequences. It has been demonstrated that features that grasp the relative orientation of amino acids in space positively impacts the prediction accuracy of the 3D coordinates of atoms in the protein backbone. In this paper, we demonstrate the relevance of Geometric Algebra (GA) in instantiating orientational features for PSP problems. We do so by proposing two novel GA-based metrics which contain information on relative orientations of amino acid residues. We then employ these metrics as an additional input features to a Graph Transformer (GT) architecture to aid the prediction of the 3D coordinates of a protein, and compare them to classical angle-based metrics. We show how our GA features yield comparable results to angle maps in terms of accuracy of the predicted coordinates. This is despite being constructed from less initial information about the protein backbone. The features are also fewer and more informative, and can be (i) closely associated to protein secondary structures and (ii) more readily predicted compared to angle maps. We hence deduce that GA can be employed as a tool to simplify the modeling of protein structures and pack orientational information in a more natural and meaningful way.

KEYWORDS:
protein structure prediction, geometric algebra, conformal geometric algebra, 3D modelling, protein modelling, graph transformer

1 | INTRODUCTION

The structure of a protein determines its function. Knowing the structure of proteins is hence relevant to understanding their role in the human body, with a big impact on medicine and drug design. An example of this is given by HIV-1 protease, which is the enzyme responsible for the replication of HIV. Understanding the structure of HIV-1 protease allowed the design of drugs that worked as inhibitors by mimicking the shape of the enzyme substrate, i.e. the molecule that the enzyme reacts with, successfully preventing the replication of the virus [1,2].

[0] **Abbreviations:** PSP, protein structure prediction; GA, geometric algebra; CGA, conformal geometric algebra; DL, deep learning; GT, graph transformer

The first protein structure to be resolved was the structure of myoglobin in 1960[3]. The 3D shape of myoglobin was determined experimentally through X-ray diffraction. In the past decade, however, deep learning (DL) has been proven to be a more successful approach to tackle the protein folding problem[4,5]. DL techniques are generally highly accurate and significantly cheaper and faster than experimental techniques such as X-ray crystallography[6], nuclear magnetic resonance (NMR)[7] or cryoelectron microscopy[8].

Hence, we now talk about protein structure *prediction* (PSP) as the protein structure is predicted through a DL architecture. To assess the state-of-the-art in the field, an international competition is held every two years, named the Critical Assessment of Protein Structure Prediction (CASP). CASP14 was won by AlphaFold 2, reaching an unprecedented global distance test (GDT) score of above 90% in almost 70% of the proteins in the CASP dataset, demonstrating how AlphaFold 2 can resolve the 3D folding of virtually any protein with known primary structure[2,9,10].

A typical DL-based PSP pipeline is generally composed of several cascaded neural networks, whose end goal is the prediction of 3D coordinates of some of the atoms in the protein backbone[11]. In recent literature, Transformer networks have been proven to be particularly suitable for coordinate prediction. Transformer networks are sequence-to-sequence models first introduced in[12], and have found widespread application in fields including speech synthesis[13], semantic correspondence[14], speech separation[15] and trajectory forecasting[16]. In PSP, for example, two of the seven networks employed in[11] are Transformer networks to predict and refine the coordinates of the backbone atoms. Similarly, a multiple sequence alignment (MSA) Transformer followed by a GT has been employed to predict 3D coordinates starting from the protein's sequence of amino acids in[17].

The goal of this paper are to: (1) employ Geometric Algebra tools to model proteins and capture information about the pairwise orientation of amino acids in the chain and (2) obtain new machine learning-ready features derived from this modeling and to employ them in a PSP pipeline based on a GT, enabling comparison with the standard angle description in terms of accuracy of the predicted 3D coordinates.

The rest of the paper is structured as follows: in Section 2 a brief summary of the state-of-the-art is provided; in Section 3 the fundamentals notions of GA and CGA are introduced and in Section 4 different modeling approach to protein structures are presented. The experimental setup is discussed in Section 5, results are presented in Section 6 and interpreted in Section 7. Lastly, conclusions are drawn in Section 8.

2 | RELATED WORK

3D coordinates are predicted by training the network on several biological and chemical features of the protein. These features are extracted starting from its amino acid sequence (or primary structure) and include interresidue distances (e.g. distance between amino acid pairs), the secondary structures of the proteins (e.g. the local folding patterns such as helices or sheets), the potential energy and others[2,10,11,18,19,20,21].

Among these features, orientational information of atoms in the backbone has been proven to be particularly relevant. Xu proposed a structure prediction approach in which orientational information is implicitly inferred from multiple distance maps for several atoms in the backbone (e.g. $C_\beta - C_\beta, C_\alpha - C_\alpha, C_g - C_g, N - C$) in[19]. This approach outperformed RaptorX, their previous contact-based prediction pipeline in[18], which lacked such kind of information.

Yang *et al.* showed that *angle maps*, constructed explicitly starting from dihedral angles of the protein backbone, rather than multiple distance maps, increase the coordinate prediction accuracy up to 2.2% compared to approaches with no orientational information[20]. Baek *et al.* also included the same angle maps of[20] as a pairwise feature as input of a Graph Transformer network[11], and Xu *et al.* employed them in a PSP pipeline deprived of proteins' co-evolutionary information in[22], suggesting a hierarchy of features in terms of their importance in the prediction: when orientational information is available, other less relevant features could be be dropped.

In AlphaFold, orientational information is embedded in the prediction pipeline by associating a 3D rigid body in the form of a rotation and translation to each amino acid in the chain[2]. A similar approach is seen in AlphaFold 2, in which a "rigid body gas", i.e. a collection of 3D rigid bodies built on each $N - C_\alpha - C$ plane of the protein backbone, is fed into a 3D Transformer architecture which updates the rigid bodies jointly with the side chain[2].

Regardless of how the pairwise amino acid orientation is captured (interatomic distances, angle maps, rigid body frames, etc) it has been demonstrated that it represents key information when trying to resolve the overall 3D protein structure. We believe that a suitable candidate for providing information on pairwise orientations of amino acid is represented by Geometric Algebra (GA), due to its intuitive handling of geometrical objects and operations on them[23,24]. GA has already found some

applications in protein modelling: for example non-normalized rotors (see Section 3 for further details) have been employed for tackling the loop closure problem, i.e. when feasible loops must be found between two given anchor residues, and to compute atomic coordinates from internal coordinates[25,26]. More specifically, Conformal Geometric Algebra (CGA, see Section 3.2) has seen application in the molecular distance problem[27,28,29,30], which provides a suitable approach in selecting plausible protein conformation starting from NMR measurements affected by uncertainty, rather than in a learning-based approach.

To the best of our knowledge, there has not been an effort to employ GA as a modelling tool in learning problems involving proteins.

3 | FUNDAMENTALS OF GEOMETRIC ALGEBRA

3.1 | Geometric Algebra

A GA with p-basis vectors that square to $+1$, q-basis vectors that square to -1 and r vectors that square to 0 is indicated with $\mathcal{G}_{p,q,r}$, with $p+q+r=n$ being the dimensionality of the space. GA is an algebra of geometric objects, which are built up via the geometric product, which for vectors is defined as

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

in which a and b are vectors, and $\cdot$ and $\wedge$ represents the inner (or dot) and outer (or $wedge$) products, respectively. Both components of Equation 1 have geometrical meaning: the inner product between a, b is proportional to the cosine of the angle between them, while the outer product corresponds to the oriented area of the parallelogram with sides a, b, and we call $a \wedge b$ a *bivector*. It is possible to define higher grade objects by multiple outer products as $A_r = a_1 \wedge a_2 \wedge ... \wedge a_r$, in which A_r is called an r-blade with grade r. The geometric product ab is called a *multivector*, since it is a linear combination of a scalar, of grade 0, and a bivector, of grade 2. Multivectors are linear combinations of blades of different grades.

We say that a geometric product of vectors $R = a_1 a_2 ... a_r$ has a reverse given by $\tilde{R} = a_r a_{r-1} ... a_1$, as long as $r \leq n$. If we scale R so that $R\tilde{R} = 1$, it can be verified that the equation

$$v' = Rv\tilde{R} \tag{2}$$

yields a rotation of v into v', as it preserves both lengths and angles. This can be easily proved since $(Rv\tilde{R})^2 = Rv^2\tilde{R} = v^2 R\tilde{R} = v^2$ and $(Rv\tilde{R}) \cdot (Rw\tilde{R}) = v \cdot w$. We refer to the object R as a *rotor* when r is even, and it is a key quantity in GA.

3.2 | Conformal Geometric Algebra

Conformal Geometric Algebra (CGA) extends a GA $\mathcal{G}_{p,q,r}$ of dimension n to $\mathcal{G}_{p+1,q+1,r}$ by introducing two basis vectors, e and $\bar{e}$ which satisfy $e^2 = +1$ and $\bar{e}^2 = -1$. Having introduced e and $\bar{e}$, we can construct two vectors

$$\begin{aligned} n_\infty &= e + \bar{e} \\ n_0 &= \frac{1}{2}(\bar{e} - e) \end{aligned} \tag{3}$$

n_∞ and n_0 represent the point at infinity and the point at the origin, respectively. These newly defined vectors are at the basis of the conformal mapping that extends GA onto CGA as follows

$$x \in \mathcal{G}_{p,q,r} \rightarrow F(x) \in \mathcal{G}_{p+1,q+1,r} \tag{4}$$

with

$$F(x) = \frac{1}{2}(x^2 n_\infty + x + n_0) \tag{5}$$

When dealing with a 3-dimensional Euclidean space, i.e. $\mathcal{G}_{3,0,0}$, the equivalent CGA will be $\mathcal{G}_{4,1,0}$, i.e. a five-dimensional space. One of the most interesting properties of CGA is the representation of geometrical objects. It can be shown how point pairs, lines, planes, circles and spheres are all easily represented by blades in the 5D CGA (see Table 1).

TABLE 1 Objects in CGA

Grade	Symbol	Object
1	A	point
2	$A \wedge B$	point pair
3	$A \wedge B \wedge C$	circle (C)
3	$A \wedge B \wedge n_\infty$	line (L)
4	$A \wedge B \wedge C \wedge D$	sphere (Σ)
4	$A \wedge B \wedge C \wedge n_\infty$	plane (Π)

4 | MODELLING THE ORIENTATION OF AMINO ACIDS

The protein backbone is responsible for the protein folding. A protein backbone is characterized by several atoms bonded together in a chain. The chemical group that differentiates amino acids between each other is bonded the alpha-Carbon (C_α) of the protein backbone. Given a string of M amino acids, we can associate to it M C_α atoms in the backbone, each of which is preceded by a nitrogen (N) atom and followed by a carbon atom (C). In this section we present three ways of describing the geometry of the protein backbone.

4.1 | Dihedral angles

Dihedral angles are the most common way to describe the orientation of atoms in the protein backbone. They are the angles between planes formed by different atom triplets in the backbone. Commonly, each amino acid i can be described with three dihedral angles $\{\phi_i, \psi_i, \omega_i\}$.

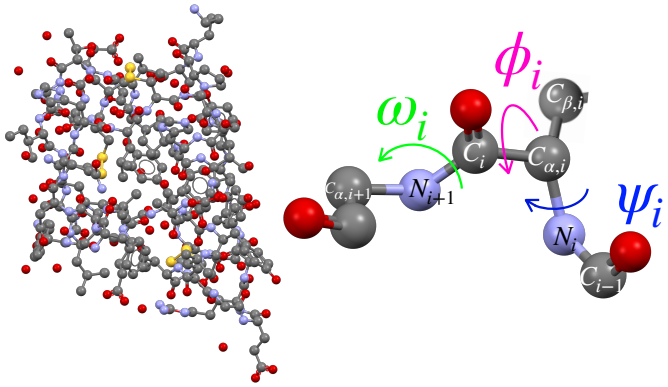
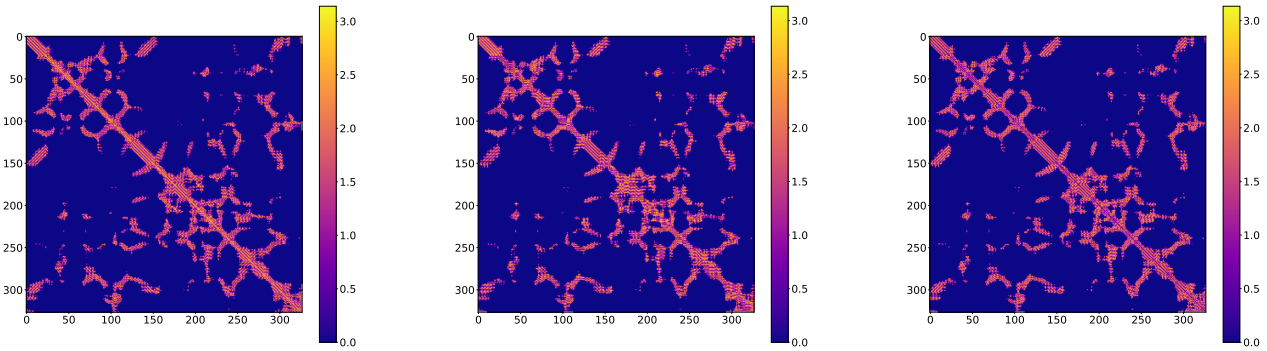


FIGURE 1 Ball-and-stick model of Insulin (ID: 3i40, left) and close up on the backbone in position i with dihedral angles ω_i, ϕ_i, ψ_i highlighted (right).

Following the convention in [31], the angle ψ_i is defined as the dihedral angle between the $C_{i-1} - N_i - C_{\alpha,i}$ and the $N_i - C_{\alpha,i} - C_i$ planes, the angle ϕ_i as the dihedral angle between the $C_{\beta,i} - C_{\alpha,i} - C_i$ and the $N_i - C_{\alpha,i} - C_i$ planes, while ω_i is the dihedral angle between the $N_{i+1} - C_i - C_{\alpha,i}$ and the $C_i - N_{i+1} - C_{\alpha,i+1}$ planes (see Fig. 1). A total of 5 planes, all relative to residue i, are involved to specify the triplet of dihedral angles $\{\phi_i, \psi_i, \omega_i\}$.

In order to build *angle maps*, we need dihedral angles between pairs of residues. We will hence define $\{\phi_{ij}, \psi_{ij}, \omega_{ij}\}$ as the dihedral angles between the same two pair planes mentioned above, but with the first plane relative to residue i and the second



(a) Φ **(b) Ψ** **(c) Ω**

FIGURE 2 Angle maps for protein 12asA.

relative to residue j. With this assumption we have that $\phi_i = \phi_{ii}, \psi_i = \psi_{ii}$ and $\omega_i = \omega_{ii}$. We can hence define the angle maps as follows:

$$\mathbf{\Phi}_{ij} = \begin{cases} \phi_{ij} & \text{if } d_{ij} < 15 \text{ Å} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$\mathbf{\Psi}_{ij} = \begin{cases} \psi_{ij} & \text{if } d_{ij} < 15 \text{ Å} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$\mathbf{\Omega}_{ij} = \begin{cases} \omega_{ij} & \text{if } d_{ij} < 15 \text{ Å} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where d_{ij} is the Euclidean distance between the C_α's of residues i, j measured in Å. Examples of angle maps are given in Figure 2 . Note how the patterns in $\mathbf{\Psi}$ are asymmetric.

4.2 | Cost function between interpolated rotors

Each triplet of atoms in the backbone lies on a plane. We can take advantage of this information and associate each triplet i with a plane Π_i in Conformal Geometric Algebra (CGA): let A_i, B_i and C_i be the CGA representations of the Euclidean coordinates of the atoms in the triplet. Π_i can be then computed as the 4-blade:

$$\Pi_i = A_i \wedge B_i \wedge C_i \wedge n_\infty \tag{9}$$

In this way, a protein is modelled as a collection of planes not too dissimilar to the gas of 3D rigid bodies of AlphaFold 2 [10] (see Figure 3).

For each pair of planes Π_i, Π_j we can then form a rotor that rotates Π_i into Π_j as presented in [32]:

$$R_{ij} = \frac{1}{\sqrt{\langle \xi \rangle_0}}(1 - \Pi_i \Pi_j) \tag{10}$$

where $\xi = 2 - (\Pi_i \Pi_j + \Pi_j \Pi_i)$ and $\langle \cdot \rangle$ is the grade projector operator. We now use a cost function $C(R)$, which is a simplified version of the cost defined in [33], that quantifies the variation of R from the identity:

$$C(R) = \langle R_\parallel \tilde{R}_\parallel \rangle_0 \tag{11}$$

in which the term $R_\parallel$ is defined as $R_\parallel = R \cdot e$. It can be shown that $R_\parallel$ contains both rotational and translational information of a pair of rotors. Since each amino acid can be associated with a plane, and each pair of planes can be associated with a rotor and

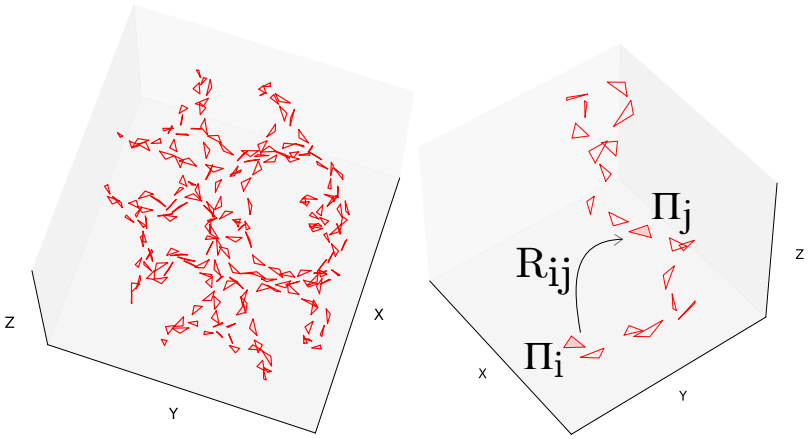
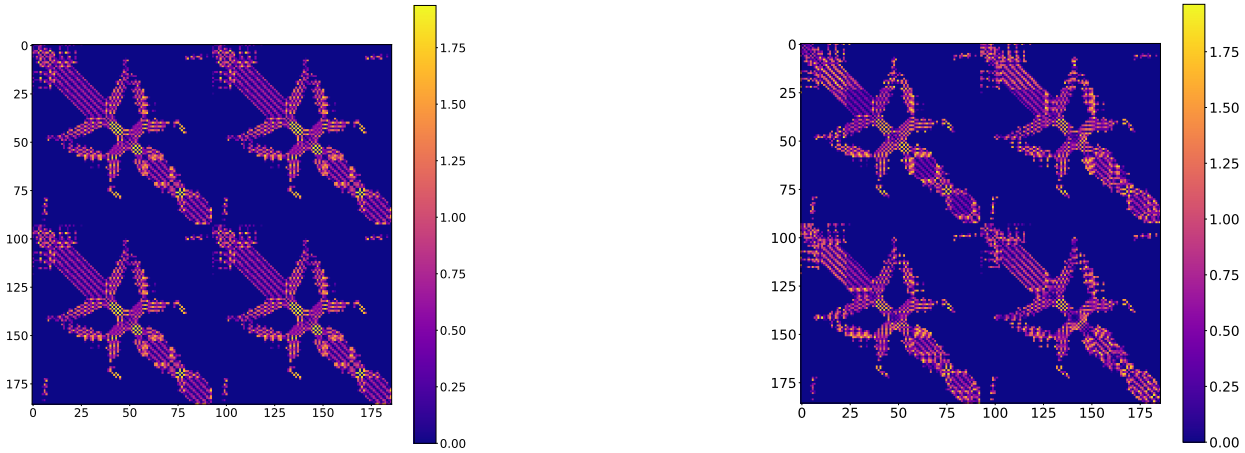


FIGURE 3 HIV-1 protease (ID: 1dmp) modelled as a collection of $N - C_\alpha - C$ planes (left) and close up on the first 20 residues (right). Note $\Pi_j = R_{ij}\Pi_i\tilde{R}_{ij}$.

eventually to a cost function, we can then build an $M \times M$ matrix $\mathbf{M}$ (in which M is the length of the amino acid sequence) as follows:

$$\mathbf{M}_{X,ij} = \begin{cases} \mathrm{C}(\mathrm{R}_{X,ij}) & \text{if } d_{ij} < 15 \text{ Å} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

We call $\mathbf{M}_X$ a "cost map" centered at atom X. We worked with two types of cost maps: M_{C_α}, constructed from the rotor between planes specified by $N - C_\alpha - C$ triplets (as in Fig. 3), and M_{C_β}, constructed from the rotor between planes specified by $C_\alpha - C_\beta - C$ triplets. We chose these two planes as the amino acid side chain is bonded to the C_α of the backbone via C_β. Examples of cost maps are given in Figure 4 .



(a) $\mathbf{M}_{C_\alpha}$ **(b)** $\mathbf{M}_{C_\beta}$

FIGURE 4 Cost maps for protein 1n6jA.

4.3 | Dot product between oriented points

Oriented points were first introduced in [34]. An oriented point Q in CGA is a *trivector*, i.e. a circle, with radius $r = 0$, defined as

$$Q = I_q \wedge q + \left[\frac{1}{2}q^2 I_q - q(q \cdot I_q)\right] n_\infty + I_q n_0 + (I_q \cdot q)E \tag{13}$$

In which $q \in \mathbb{R}^3$ is the 3D position vector defining the centre of Q, I_q is the (unit) bivector corresponding to the oriented plane in which Q lies (orthogonal to the normal n_q of the plane) and $E = n_\infty \wedge n_0$ is the origin-infinity bivector.

Given a pair of oriented points P, Q, the dot product between them was first studied in [35], and it was derived to be equal to

$$P \cdot Q = d^2 \left[-\frac{1}{2}\cos\alpha_{pq} + \cos\Theta_q \cos\Theta_p\right] \tag{14}$$

in which d is the Euclidean distance between $p, q \in \mathbb{R}^3$, α_{pq} the dihedral angle between the two planes $\cos\alpha_{pq} = n_q \cdot n_q$, $\cos\Theta_p = d \cdot n_p$ and $\cos\Theta_q = d \cdot n_q$.

Equation 14 tells us that, up to a scale factor, the dot product between two oriented points is a function of three angles, hence it encodes the orientation between them. Moreover, similarly to the cost function of Section 4.2, Equation 14 allows us obtain a scalar measure from an initial GA description.

As each atom in the backbone can be associated to a set of 3D coordinates and it lies on a plane specified by the bonds to the previous and following atoms in the backbone, it is easy to associate an atom with an oriented point. We will call Q_{C_α} the oriented point centered on C_α in the $N - C_\alpha - C$ plane (see Fig. 5), Q_{C_β} the oriented point centered on C_β in the $C_\alpha - C_\beta - C$ plane and so on.

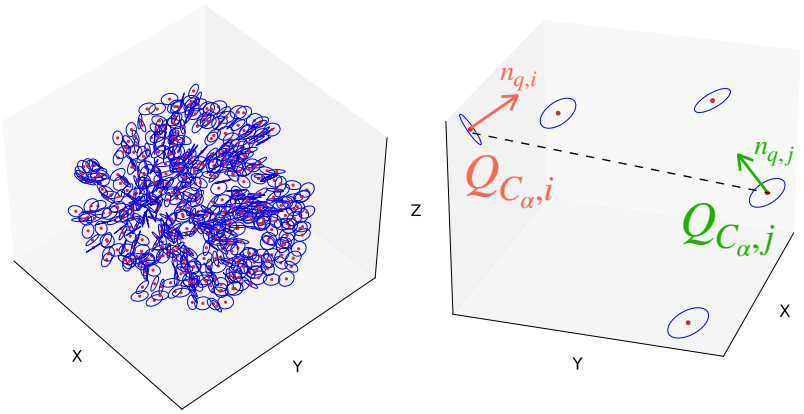


FIGURE 5 Haemoglobin (ID: 1a3n) modelled as a collection of oriented points centered at C_α (left) and close up on 5 residues with labelled oriented point (right).

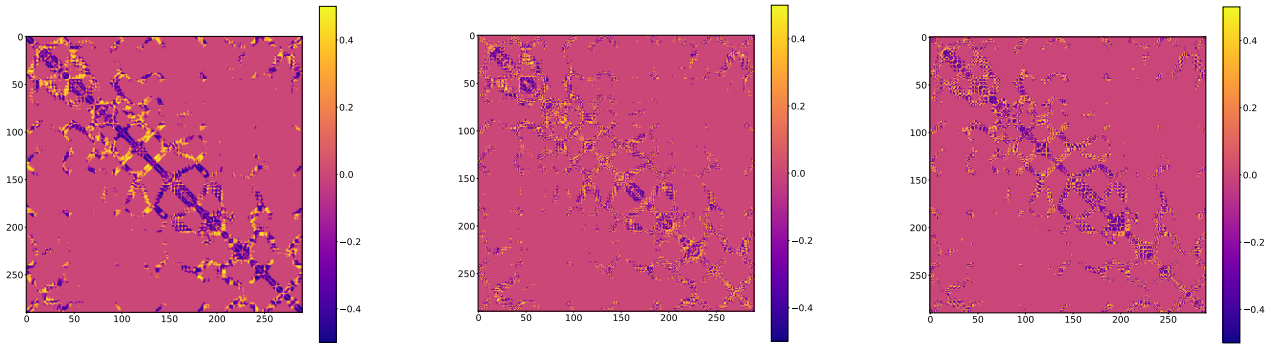
We can again build an $M \times M$ map for each protein chain of length M, which we will call *dot product maps*, defined as:

$$\mathbf{N}_{X,ij} = \begin{cases} \frac{1}{d_{ij}^2}(\mathrm{Q}_{X,i} \cdot Q_{X,j}) & \text{if } d_{ij} < 15 \text{ Å} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

In which X is any atom among C_α, C_β, C or N, and d_{ij} is the Euclidean distance between X_i and X_j. Note the normalization factor as we are only interested in the orientation between atoms. Examples of dot product maps maps are given in Figure 6 .

5 | EXPERIMENTS

Having defined these orientational maps, we now proceed to study how they impact the prediction of protein coordinates.



(a) $\mathbf{N}_{C_\alpha}$ **(b)** $\mathbf{N}_{C_\beta}$ **(c)** $\mathbf{N}_N$

FIGURE 6 Dot product maps for protein 3i41A.

5.1 | Dataset

We employed an expanded version of PDNET to train our models[21]. The original PDNET includes 1000 proteins for training and 150 proteins for testing, of which the top seven features in contact and distance prediction problems are picked. These features are sequence profiles, secondary structures, solvent accessibility, coevolutionary signals, FreeContact[36], contact potentials alignments and Shannon entropy, all predicted starting from the amino acid sequence. The seven features are encoded in PDNET as a stack of 57 $M \times M$ channels for each of its protein chains of length M. Of the 57 channels, 3 of them correspond to pairwise features, i.e. FreeContact, coevolutionary signals and contact potentials, while the remaining 54 are 27 individual features repeated twice as a matrix and their corresponding transpose (i.e. Y and Y^T).

We expanded PDNET by adding distance maps (defined as $\mathbf{D}_{ij} = d_{ij}$, where $d_{ij} = \|T_i - T_j\|_2$, with $T \in \mathbb{R}^{N\times 3}$ being the ground truth coordinates of the C_α atoms of the protein) and eight possible combinations of the cost maps described in Section 4 (namely $\mathbf{M}_{C_\alpha}, \mathbf{M}_{C_\beta}, \mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}, \mathbf{N}_N, \mathbf{\Omega}, \mathbf{\Phi}$ and $\mathbf{\Psi}$), for a total of nine different cases (see Table 2).

We then recast the dataset in the form of heterogeneous graphs so that it could be input into the GT (see Section 5.2.1). By heterogeneous graph $\mathcal{G}(V, E)$ with V and E being its set of nodes and edges, we refer to a graph with different types of nodes and edges. If $|V| = M$ is the total number of nodes, the graph can be described as a set of adjacency matrices for each of the K edge types, i.e. $\{A_k\}_{k=1}^K$, where $A_k \in \mathbb{R}^{M\times M}$, or in tensor form $\mathbf{A} \in \mathbb{R}^{M\times M\times K}$: we say that $A_{k;i,j}$ is non-zero when there exists an edge of type k between nodes i, j. Along with $\mathbf{A}$, we can also define a feature matrix $X \in \mathbb{R}^{M\times D}$, where D is the dimensionality of the features, or equivalently we can say there are D node types.

Ignoring the transposed matrices, we have $D = 27$ channels which correspond to features relative to individual amino acids which can be manipulated and arranged in a feature matrix $X \in \mathbb{R}^{M\times D=27}$. On the other hand, the pairwise features range from a minimum of $K = 4$ in case (a) up to a maximum of $K = 7$ in cases (g)-(i) which correspond to the edges of the protein graph, i.e. the adjacency matrices $\mathbf{A} \in \mathbb{R}^{M\times M\times K}$ (see Table 2).

The input to the architecture is then given by the pair of tensors $\{\mathbf{A}, X\}^{(i)}$ for each protein i in the dataset. Training details are given in Appendix B.1.

5.2 | Architecture

The end-to-end architecture, derived from[17], is composed of two parts: (1) a Graph Transformer and (2) a 3D projector. A summary of the architecture is shown in Fig. 7 . We omitted the MSA Transformer of[17] as the employed dataset allows us to directly perform node and edge embedding on its features.

5.2.1 | The Graph Transformer

The GT has been implemented as described in[37]. The goal of a GT is to learn informative meta-paths within the graph, i.e. an ordered sequence of node types and edge types. The GT also implements an attention mechanism, which is a function of the

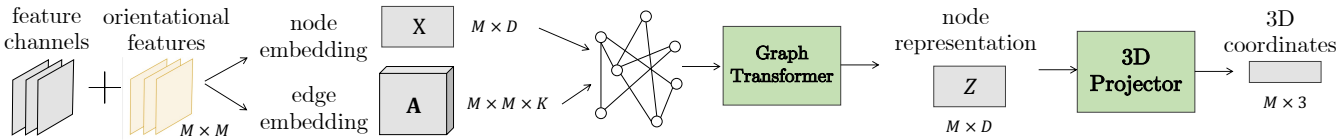


FIGURE 7 The employed architecture. Features are recast in graph form to predict 3D backbone coordinates via a GT + 3D projector architecture

neighbourhood connectivity for each node in the graph[38]. The output of the l-th layer of a GT with C attention heads is a node representation with the same dimensionality as X, i.e. $Z \in \mathbb{R}^{M\times D}$ which can be written as

$$Z^{(l)} = \bigoplus_{i=1}^{C} \sigma(\tilde{\Delta}_i^{-1} \tilde{A}_i^{(l)} X W) \tag{16}$$

where $\bigoplus$ is the concatenation operator, $\sigma(\cdot)$ is the sigmoid function, $\tilde{\Delta}_i$ is the degree matrix of $\tilde{A}_i^{(l)}$ (defined as $\Delta_m = \sum_n A_{mn}$), X is the feature matrix, $W \in \mathbb{R}^{D\times D}$ is a trainable weight matrix and $\tilde{A}_i^{(l)} = A_i^{(l)} + I$, in which $A_i^{(l)}$ is the adjacency matrix from the i-th channel of the metapath tensor $\mathbf{A}^{(l)} \in \mathbb{R}^{M\times M\times C}$. $\mathbf{A}^{(l)}$ is evaluated as $\mathbf{A}^{(l)} = \Delta^{-1}\mathbf{Q}_1\mathbf{Q}_2$. $\mathbf{Q}_1$ and $\mathbf{Q}_2$, both $\in \mathbb{R}^{M\times M\times C}$, are two adjacency tensors selected according to:

$$\mathbf{Q} = \varphi[\mathbf{A}; \zeta(\mathbf{W}_\varphi)] \tag{17}$$

where $\mathbf{A} \in \mathbb{R}^{M\times M\times K}$ is the adjacency tensor, $\varphi(\cdot)$ is the convolution operator, $\zeta(\cdot)$ is the softmax function and $\mathbf{W}_\varphi \in \mathbb{R}^{C\times C\times K}$ are the weights of φ. Z contains the node representations from C different meta-path graphs.

5.2.2 | The 3D Projector

The 3D projector is a simple fully connected layer obeying

$$P = Z^{(L)} W_P \tag{18}$$

where $Z^{(L)}$ is the output of the L-th layer of the GT, $W_P \in \mathbb{R}^{D\times 3}$ is the weight matrix of the projector and $P \in \mathbb{R}^{M\times 3}$ are the 3D coordinates of the M C_α atoms in the protein chain. To train the model, a distance map is evaluated for each protein from the predicted coordinates P as $\tilde{\mathbf{D}}_{ij} = d_{ij}$, where $d_{ij} = \|P_i - P_j\|_2$ is the Euclidean distance between the 3D coordinates of the i-th and j-th amino acid in P.

TABLE 2 Combinations of orientational features. The column " Planes" specifies how many planes are required to build the corresponding set of features. The column "K" indicates the total number of adjacency matrices of the graph.

Case	Additional Features	Planes	K
(a)	none	-	4
(b)	$\mathbf{M}_{C_\alpha}$	1	5
(c)	$\mathbf{M}_{C_\alpha}, \mathbf{M}_{C_\beta}$	2	6
(d)	$\mathbf{N}_{C_\alpha}$	1	5
(e)	$\mathbf{M}_{C_\alpha}, \mathbf{N}_{C_\alpha}$	2	6
(f)	$\mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}$	2	6
(g)	$\mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}, \mathbf{N}_N$	3	7
(h)	$\mathbf{M}_{C_\alpha}, \mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}$	3	7
(i)	$\mathbf{\Omega}, \mathbf{\Phi}, \mathbf{\Psi}$	5	7

6 | RESULTS

For each protein of length M we obtained an $M \times 3$ point cloud P of predicted 3D coordinates. We then aligned the predicted coordinates to the ground truth coordinates T (obtained from the Protein Data Bank (PDB)[39]) via singular value decomposition (SVD) (see Appendix A) and performed the GDT, and evaluated the GDT_TS (total score) and GDT_HA (half size) between P and T as follows:

$$\text{GDT\_TS} = \frac{p_{<1\text{Å}} + p_{<2\text{Å}} + p_{<4\text{Å}} + p_{<8\text{Å}}}{4} \tag{19}$$

$$\text{GDT\_HA} = \frac{p_{<0.5\text{Å}} + p_{<1\text{Å}} + p_{<2\text{Å}} + p_{<4\text{Å}}}{4} \tag{20}$$

where $p_{<n\text{Å}}$ indicates the percentage of an amino acid's coordinates in P whose distance from the corresponding amino acid's coordinates in T is below n Å. Results for 5 test sets D1 to D5 are summarized in Tables 3 -4 . The highest GDT score has been highlighted in bold and the second highest has been underlined.

TABLE 3 GDT_TS scores over the five datasets.

	Case	D1			D2			D3			D4			D5		
		max	**med.**	**min**	**max**	**med.**	**min**	**max**	**med.**	**min**	**max**	**med.**	**min**	**max**	**med.**	**min**
(a)	no orientation	23.99	7.48	0.73	22.46	7.36	0.87	21.32	7.66	0.81	21.32	7.66	0.81	25.98	7.60	1.13
(b)	$\mathbf{M}_{C_\alpha}$	30.00	11.61	1.71	28.73	11.42	1.87	36.34	12.50	2.25	32.18	12.05	1.55	32.79	12.50	1.58
(c)	$\mathbf{M}_{C_\alpha}, \mathbf{M}_{C_\beta}$	38.51	11.72	1.70	35.23	11.73	1.77	38.67	12.21	1.89	34.48	11.53	2.33	38.64	12.90	1.42
(d)	$\mathbf{N}_\alpha$	35.82	11.21	2.29	31.73	11.89	1.01	37.07	12.02	2.72	33.04	11.02	2.47	37.09	11.75	2.91
(e)	$\mathbf{M}_{C_\alpha}, \mathbf{N}_{C_\alpha}$	39.58	11.51	2.58	36.05	12.34	2.00	29.72	12.11	2.21	28.97	11.86	1.09	38.31	12.11	0.13
(f)	$\mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}$	32.20	11.56	2.35	34.09	11.75	1.20	32.59	12.33	1.03	33.62	11.90	2.12	31.25	12.66	3.05
(g)	$\mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}, \mathbf{N}_N$	29.74	12.00	1.83	39.39	12.39	1.41	41.37	12.71	1.49	38.99	11.65	1.62	37.75	13.03	2.03
(h)	$\mathbf{M}_{C_\alpha}, \mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}$	33.85	11.73	2.71	31.85	11.97	0.89	29.76	**12.92**	1.79	30.74	**12.42**	2.03	43.13	11.24	0.66
(i)	$\mathbf{\Omega}, \mathbf{\Phi}, \mathbf{\Psi}$	33.21	**12.29**	2.29	33.04	**13.10**	1.42	32.33	12.69	1.58	29.61	12.05	1.82	32.81	**13.75**	1.82

TABLE 4 GDT_HA scores over the five datasets.

	Case	D1			D2			D3			D4			D5		
		max	**med.**	**min**	**max**	**med.**	**min**	**max**	**med.**	**min**	**max**	**med.**	**min**	**max**	**med.**	**min**
(a)	no orientation	9.12	1.61	0.00	7.41	1.81	0.00	8.02	1.74	0.10	8.02	1.74	0.10	8.33	1.80	0.00
(b)	$\mathbf{M}_{C_\alpha}$	10.27	2.55	0.08	10.06	2.42	0.35	15.09	2.68	0.31	11.62	2.05	0.13	13.11	2.78	0.00
(c)	$\mathbf{M}_{C_\alpha}, \mathbf{M}_{C_\beta}$	16.53	2.50	0.22	12.89	2.46	0.00	16.41	2.42	0.51	11.78	2.43	0.29	15.26	2.81	0.26
(d)	$\mathbf{N}_\alpha$	15.30	2.68	0.22	11.06	**3.00**	0.14	16.05	2.82	0.13	12.07	2.60	0.51	16.01	2.88	0.59
(e)	$\mathbf{M}_{C_\alpha}, \mathbf{N}_{C_\alpha}$	15.83	**2.96**	0.20	12.46	2.97	0.21	11.79	2.72	0.28	9.20	**2.85**	0.00	15.90	2.12	0.00
(f)	$\mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}$	11.36	2.20	0.18	10.23	2.07	0.10	10.21	2.22	0.34	11.21	2.27	0.00	12.59	2.62	3.05
(g)	$\mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}, \mathbf{N}_N$	10.38	2.56	0.28	17.04	2.57	0.20	17.67	3.03	0.11	16.37	2.69	0.17	15.46	**2.94**	0.34
(h)	$\mathbf{M}_{C_\alpha}, \mathbf{N}_{C_\alpha}, \mathbf{N}_{C_\beta}$	12.69	2.59	0.33	11.36	3.06	0.20	10.12	2.85	0.28	9.41	2.64	0.13	21.08	2.71	0.00
(i)	$\mathbf{\Omega}, \mathbf{\Phi}, \mathbf{\Psi}$	13.81	2.93	0.23	11.56	2.51	0.00	12.35	**3.20**	0.28	10.00	2.83	0.30	13.38	2.91	0.30

There are several things to note here: firstly, by adding orientational information, an improvement on the protein coordinates quality corresponding to at least $\sim 4\%$ compared to the approach without orientational information can be measured, as found in[20].

Secondly, for all the analyzed cases (b)-(i) the relative improvement is generally $< 2\%$, which implies that no clear superior approach to modeling the amino acid orientation exists. This means that, as long as orientational information is added, an improvement is going to be seen regardless of how this information is encoded. This is also mirrored in the training loss, which is significantly larger only for case (a) (see Figure 8).

However, the orientational information encoded through our GA-based metrics generally require much less information about the protein backbone than encoding orientation as angle maps. Compare, for example, case (d) with (i) in Table 4 : in case

TABLE 5 GDT_TS score for 10 example proteins (GDT_HA score in parenthesis)

Protein ID	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
1mk0A	15.46 (6.44)	23.45 (5.15)	14.43 (2.58)	14.94 (4.90)	16.49 (3.35)	22.68 (**8.25**)	**30.93** (7.99)	21.13 (7.99)	18.81 (6.70)
1yqhA	9.37 (3.12)	16.34 (3.85)	16.10 (4.08)	12.25 (1.68)	19.47 (6.97)	**24.04** (6.73)	21.87 (4.57)	21.15 (6.94)	20.91 (**8.65**)
1zv1A	22.89 (5.93)	21.61 (6.35)	**28.81** (8.90)	22.88 (**9.74**)	20.34 (8.89)	20.34 (3.81)	25.84 (7.63)	**28.81** (5.93)	19.49 (6.36)
2d0oB	12.96 (4.17)	11.80 (2.31)	13.66 (3.24)	17.13 (5.55)	15.74 (5.55)	19.91 (5.78)	**25.69** (**8.10**)	25.46 (7.87)	20.37 (7.41)
2dgbA	9.04 (2.71)	13.85 (2.71)	16.87 (3.01)	18.37 (6.32)	**23.49 6.93**)	20.18 (6.02)	17.49 (5.42)	13.25 (4.22)	19.91 (6.48)
2dm9A	12.29 (2.97)	15.67 (4.24)	18.64 (4.45)	16.73 (4.66)	**18.86** (**7.20**)	13.13 (3.60)	14.19 (2.33)	15.04 (4.24)	10.59 (2.12)
2ehwA	8.69 (1.09)	10.87 (2.83)	10.65 (2.83)	12.39 (3.26)	11.74 (3.48)	10.22 (2.82)	11.09 (2.61)	15.00 (3.26)	**22.17** (**7.17**)
2fyuK	10.38 (1.89)	15.10 (5.66)	16.98 (5.66)	**29.72** (**9.43**)	18.87 (5.19)	16.51 (2.83)	26.89 (6.13)	14.62 (4.72)	24.06 (7.55)
2fztA	14.74 (2.24)	25.96 (7.37)	10.58 (2.24)	**27.88** (**8.33**)	19.23 (6.09)	15.06 (4.49)	25.00 (6.73)	17.63 (4.17)	21.47 (3.52)
2gomA	19.67 (7.79)	26.64 (9.43)	27.46 (6.56)	25.00 (9.84)	**28.28** (9.43)	25.00 (6.56)	27.87 (8.20)	**27.46** (9.02)	25.00 (**11.07**)

(d) we have a single dot product map $\mathbf{N}_{C_\alpha}$ ($K = 5$), constructed over oriented points at position C_α in the plane specified by the $N - C_\alpha - C$ triplet, that yields comparable results to case (i), in which angle maps are three additional features ($K = 7$) constructed over a total of 5 different planes.

This tells us two things: GA-based metrics can (1) distill more information about the relative orientation of amino acids from of fewer geometrical objects, and (2) that this information can be condensed in fewer features, which are equivalent to the 3 angle maps approach.

Examples of GDT scores for 10 proteins are given in Table 5 . Again, it can be seen that no approach is consistently superior for randomly selected proteins. Examples of predicted coordinates for selected cases are given in Figures 9 -10 -11 .

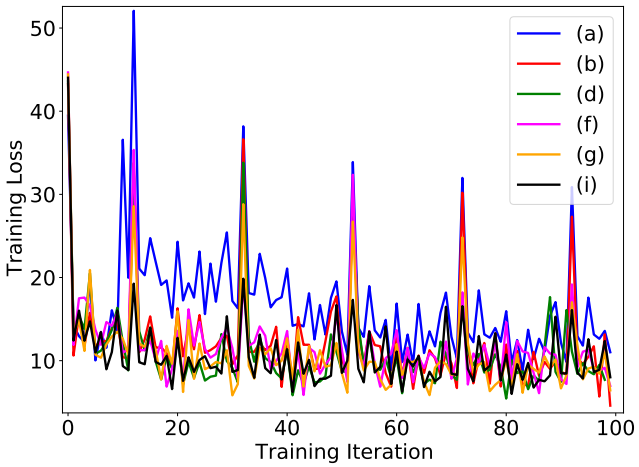


FIGURE 8 Training loss sampled every 10 learning steps for selected approaches. With the exception of case (a), the training loss profiles are generally undistinguishable.

7 | DISCUSSION

7.1 | Geometrical meaning of features

The fact that our GA approach yields comparable results to angle maps by employing fewer features obtained from less geometrical information can be explained with how the information is packed in GA based features: it is possible to establish a relationship between the secondary structure and the patterns in the cost maps. By secondary structure we refer to local folding patterns of a protein, most commonly α-helices and β-sheets.

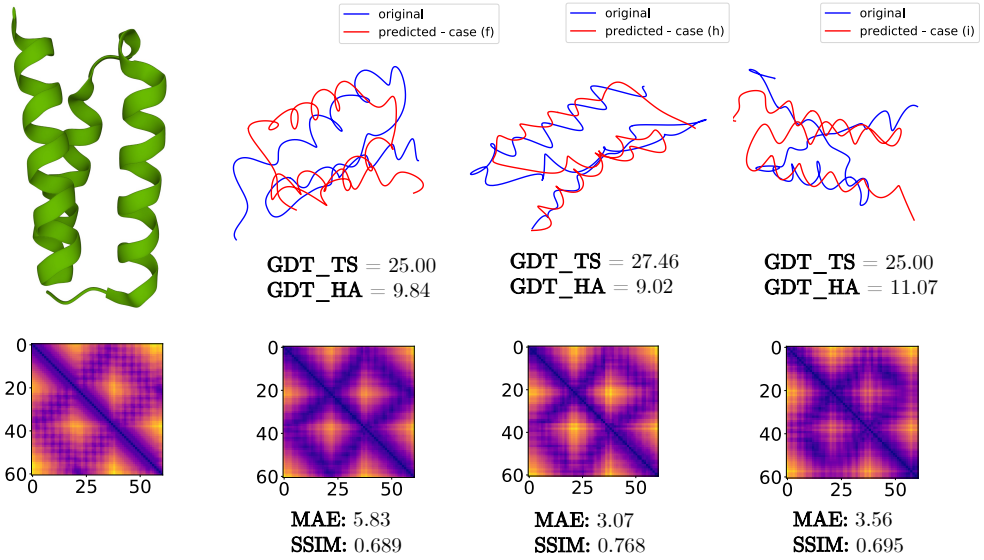


FIGURE 9 Results for protein 2gomA. The original 3D protein model is shown in green with the original distance map **D** below. The ground truth and predicted coordinates T, P are given on the top row for selected cases in red and blue, respectively, with their corresponding GDT scores. Below, the distance map $\hat{\mathbf{D}}$ built from P and the MAE and SSIM measured with respect to **D**.

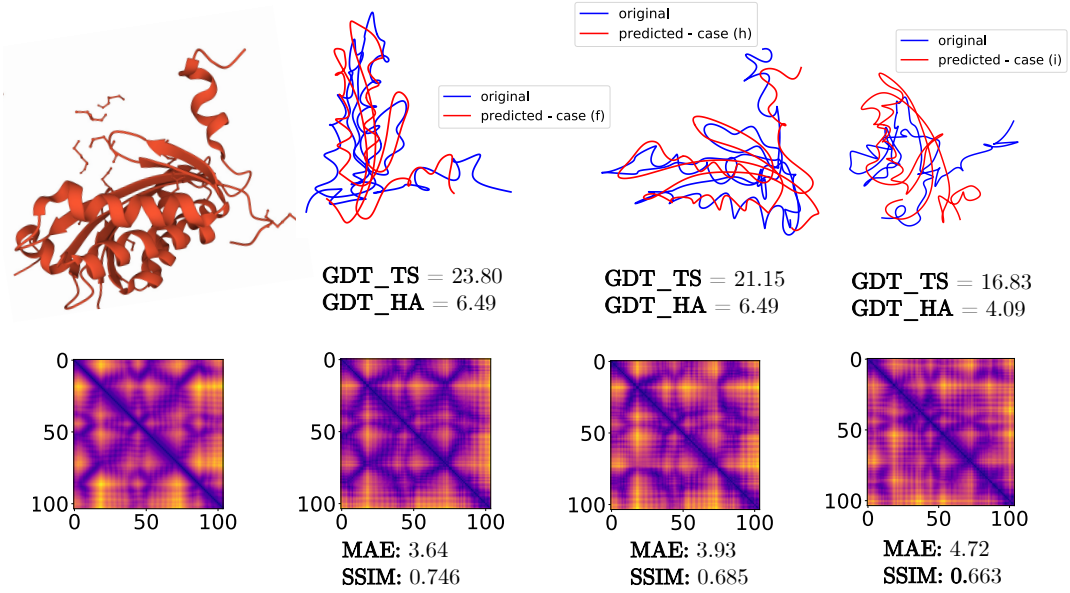


FIGURE 10 Results for protein 1yqhA. The original 3D protein model is shown in red with the original distance map **D** below. The ground truth and predicted coordinates T, P are given on the top row for selected cases in red and blue, respectively, with their corresponding GDT scores. Below, the distance map $\hat{\mathbf{D}}$ built from P and the MAE and SSIM measured with respect to **D**.

We illustrate this relationship by assigning an arbitrary colour to each secondary structure: red to α-helices, green to β-sheets, blue to turns and white to all the others. In Fig. 12 we see how there is almost one-to-one correspondence between colour patches of secondary structures with patterns in cost maps. This shows how cost maps, despite being built starting from a single plane in the backbone ($N - C\alpha - C$), contain information about the protein folding that is among the most relevant in PSP pipelines, namely secondary structures.

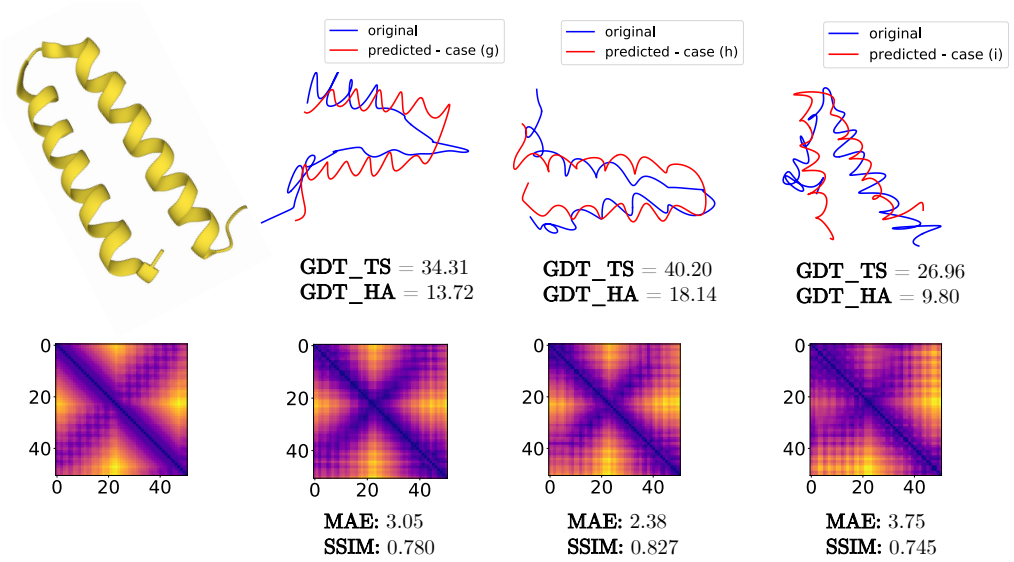


FIGURE 11 Results for protein 1z0jB. The original 3D protein model is shown in yellow with the original distance map **D** below. The ground truth and predicted coordinates T, P are given on the top row for selected cases in red and blue, respectively, with their corresponding GDT scores. Below, the distance map $\hat{\mathbf{D}}$ built from P and the MAE and SSIM measured with respect to **D**.

To the best of our knowledge, this is the first example of a single orientational map that clearly matches the secondary structures.

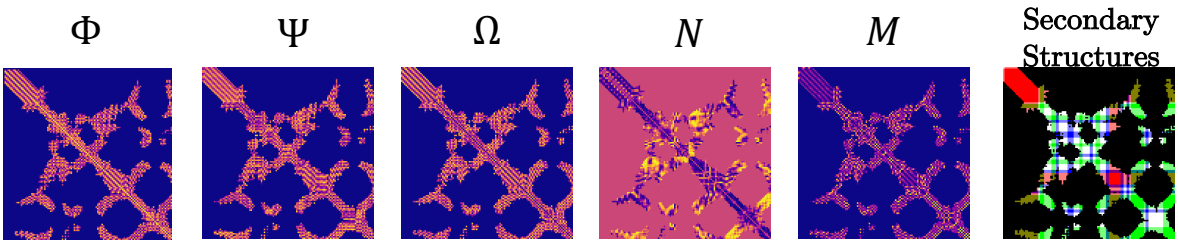


FIGURE 12 Orientational maps and secondary structures. Helices (red) and sheets (green) can be easily spotted from the patterns in the cost map **M**.

7.2 | Predictability of features

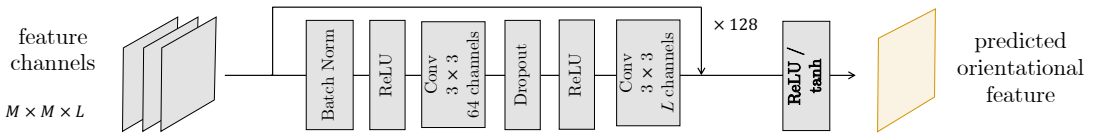


FIGURE 13 Predicting orientational features. We employed the residual neural network of[21] to predict (a) cost maps $\mathbf{M}_\alpha$, (b) dot product maps $\mathbf{N}_\alpha$ and (c) angle maps $\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Omega}$. While all (true) orientational features provide similar improvements when predicting coordinates, not all of them are as readily predicted.

Results presented so far assume that the orientational features are built starting from ground truth coordinates of atoms in the protein. In a realistic scenario, however, the coordinates are the end goal of the pipeline, not the starting point. We would then have to employ *predicted* orientational features, which are less clear than the ground truth features shown so far and must themselves be estimated from other features or MSA alignments of the amino acid sequence.

We hence employed the PDNET pipeline in[21], originally designed to predict distance maps, to compare how easily different orientational maps are predicted starting from same set of features and the same pipeline. We attempted predicting cost maps $\mathbf{M}_{C_\alpha}$, dot product maps $\mathbf{N}_{C_\alpha}$ and the three angle maps. The prediction pipeline is shown in Figure 13 , while training details are outlined in Appendix B.2.

Example of predictions are given in Figures 14 -15 : cost maps $\mathbf{M}$ were consistently more accurately predictable. This can be explained with the fact that they are closely related to secondary structure information, which is implicitly included in other input features. Less accurate are the predictions of angle maps, which don't present easily recognizable patterns or are asymmetric as in $\mathbf{\Psi}$. A similar consideration can be made for dot product maps.

While all *true* orientational maps increase the coordinate prediction accuracy in a similar way, not all orientational maps are readily predicted, meaning that in a realistic PSP pipeline we will prefer features that are easier to estimate for the same added value in terms of GDT scores.

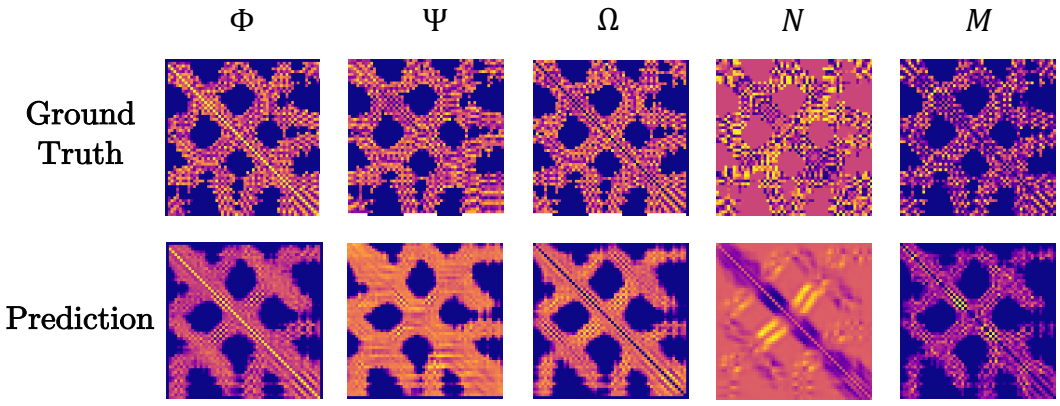


FIGURE 14 Ground truth and predicted orientational maps for protein 1a3aA.

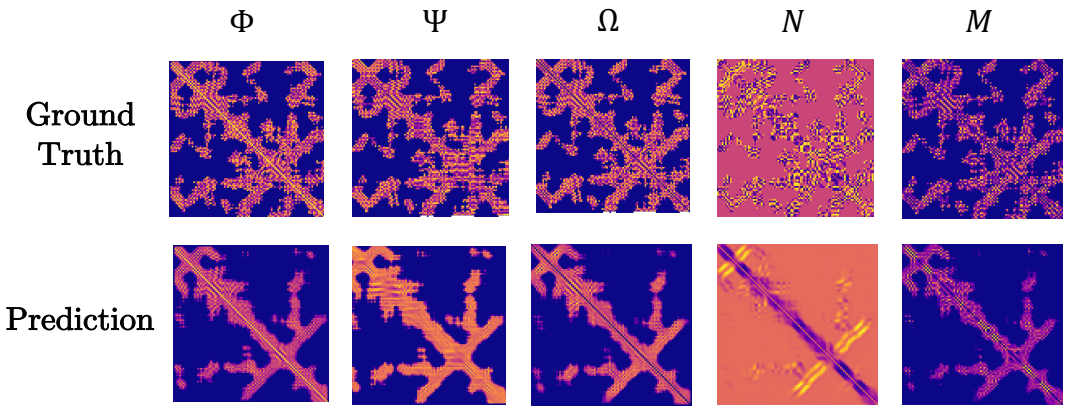


FIGURE 15 Ground truth and predicted orientational maps for protein 1a70A.

8 | CONCLUSIONS

This paper dealt with the issue of orientational features in PSP pipelines. We employed GA as a tool to model the protein backbone as a rigid body and built two novel features from it, which we named cost maps and oriented point maps, and compared their impact on PSP accuracy side by side with traditional angle maps. We verified that adding GA-based orientational features improves the accuracy in terms of GDT scores in a similar way to angle maps, but requiring less geometrical information about the protein backbone, generally from 1 up to 3 planes compared to 5 planes required by angle maps. This shows that GA condenses orientational information of proteins in fewer, more informative features. Moreover, we also showed: (i) how patterns in cost maps can be immediately associated to the protein secondary structures, which is what determines the overall folding and is one of the most relevant feature in PSP problems, and (ii) how cost maps are easily predictable compared to other features.

We hence believe that employing GA-based orientational features, which require little prior knowledge of the protein structure and are closely related to secondary structures, is going to be especially relevant in realistic PSP scenarios in which geometrical information about the protein chain is limited and all the input features have to be predicted as accurately as possible.

8.1 | Limitations

This paper aims to be a proof-of-concept for different measures of interresidue orientation for PSP problems and how they impact the predicted protein coordinates compared to more standard angle maps. We did not focus on boosting the training accuracy: we are positive that adding more proteins to the training set, employing a larger dataset which includes longer, more complex protein chains, or training the network for longer are all elements that can significantly improve the GDT scores. Moreover, we provided only a subset of possible combinations of the proposed maps.

References

1. Huff JR. HIV protease: a novel chemotherapeutic target for AIDS. *Journal of medicinal chemistry* 1991; 34(8): 2305–2314.
2. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold. *Nature* 2021; 596(7873): 583–589.
3. Phillips D, Shore V. Structure of myoglobin: A three-dimensional Fourier synthesis at 2 A. resolution. *Nature* 1960; 185: 422–7.
4. Torrisi M, Pollastri G, Le Q. Deep learning methods in protein structure prediction. *Computational and Structural Biotechnology Journal* 2020; 18: 1301–1310.
5. Kandathil SM, Greener JG, Jones DT. Recent developments in deep learning applied to protein structure prediction. *Proteins: Structure, Function, and Bioinformatics* 2019; 87(12): 1179–1189.
6. Smyth M, Martin J. x Ray crystallography. *Molecular Pathology* 2000; 53(1): 8.
7. Hall L. Nuclear magnetic resonance. *Advances in carbohydrate chemistry* 1964; 19: 51–93.
8. Fernandez-Leiro R, Scheres SH. Unravelling biological macromolecules with cryo-electron microscopy. *Nature* 2016; 537(7620): 339–346.
9. Perrakis A, Sixma TK. AI revolutions in biology: The joys and perils of AlphaFold. *EMBO reports* 2021; 22(11): e54046.
10. Jumper J, Evans R, Pritzel A, et al. AlphaFold 2. *In Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book* 2020.
11. Baek M, DiMaio F, Anishchenko I, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* 2021; 373(6557): 871–876.
12. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K. Spatial transformer networks. Advances in Neural Information Processing Systems 28. 2015.
13. Li N, Liu S, Liu Y, Zhao S, Liu M. Neural speech synthesis with transformer network. In: . 33 of *Proceedings of the AAAI Conference on Artificial Intelligence*. ; 2019: 6706–6713.
14. Kim S, Lin S, Jeon SR, Min D, Sohn K. Recurrent transformer networks for semantic correspondence. *Advances in neural information processing systems* 2018; 31.
15. Chen J, Mao Q, Liu D. Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation. *arXiv preprint arXiv:2007.13975* 2020.
16. Giuliari F, Hasan I, Cristani M, Galasso F. Transformer networks for trajectory forecasting. In: 2020 25th international conference on pattern recognition (ICPR). IEEE. ; 2021: 10335–10342.
17. Costa A, Ponnapati M, Jacobson JM, Chatterjee P. Distillation of MSA Embeddings to Folded Protein Structures with Graph Transformers. *bioRxiv* 2021.
18. Peng J, Xu J. RaptorX: exploiting structure information for protein alignment by statistical inference. *Proteins: Structure, Function, and Bioinformatics* 2011; 79(S10): 161–171.
19. Xu J. Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences* 2019; 116(34): 16856–16865.
20. Yang J, Anishchenko I, Park H, Peng Z, Ovchinnikov S, Baker D. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences* 2020; 117(3): 1496–1503.
21. Adhikari B. A fully open-source framework for deep learning protein real-valued distances. *Scientific reports* 2020; 10(1): 1–10.

22. Xu J, Mcpartlon M, Li J. Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nature Machine Intelligence* 2021; 3(7): 601–609.

23. Doran C, Lasenby A. *Geometric algebra for physicists*. Cambridge University Press . 2003.

24. Dorst L, Doran C, Lasenby J. *Applications of geometric algebra in computer science and engineering*. Springer Science & Business Media . 2012.

25. Chys P, Chacón P. Spinor product computations for protein conformations. *Journal of Computational Chemistry* 2012; 33(21): 1717–1729.

26. Chys P, Chacón P. Random coordinate descent with spinor-matrices and geometric filters for efficient loop closure. *Journal of chemical theory and computation* 2013; 9(3): 1821–1829.

27. Lavor C, Alves R. Oriented conformal geometric algebra and the molecular distance geometry problem. *Advances in Applied Clifford Algebras* 2019; 29(1): 1–15.

28. Alves R, Lavor C. Geometric algebra to model uncertainties in the discretizable molecular distance geometry problem. *Advances in Applied Clifford Algebras* 2017; 27(1): 439–452.

29. Dorst L. Boolean combination of circular arcs using orthogonal spheres. *Advances in Applied Clifford Algebras* 2019; 29(3): 1–21.

30. Hitzer E, Lavor C, Hildenbrand D. Current survey of Clifford geometric algebra applications. *Mathematical Methods in the Applied Sciences* 2022.

31. Ceci G, Mucherino A, D'Apuzzo M, et al. Computational methods for protein fold prediction: an ab-initio topological approach. In: Data Mining in Biomedicine. Springer. 2007 (pp. 391–429).

32. Lasenby J, Hadfield H, Lasenby A. Calculating the rotor between conformal objects. *Advances in Applied Clifford Algebras* 2019; 29(5): 1–9.

33. Eide E. Master's Degree Thesis. *University of Cambridge, Camera Calibration using Conformal Geometric Algebra* 2018.

34. Hildenbrand D, Charrier P. Conformal geometric objects with focus on oriented points. In: ICCA9, 7th International Conference on Clifford Algebras and Their Applications. ; 2011.

35. Hitzer E. Inner product of two oriented points in conformal geometric algebra. In: ICACGA 2022, 1st International Conference on Advanced Computational applications of Geometric Algebra. ; 2022.

36. Kaján L, Hopf TA, Kalaš M, Marks DS, Rost B. FreeContact: fast and free software for protein contact prediction from residue co-evolution. *BMC bioinformatics* 2014; 15(1): 1–6.

37. Yun S, Jeong M, Kim R, Kang J, Kim HJ. Graph transformer networks. *Advances in neural information processing systems* 2019; 32.

38. Dwivedi VP, Bresson X. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699* 2020.

39. Burley SK, Berman HM, Kleywegt GJ, Markley JL, Nakamura H, Velankar S. Protein Data Bank (PDB): the single global macromolecular structure archive. *Protein Crystallography* 2017: 627–641.

40. Lasenby A, Lasenby J, Matsantonis C. Reconstructing a rotor from initial and final frames using characteristic multivectors: With applications in orthogonal transformations. *Mathematical Methods in the Applied Sciences* 2022.

41. Hadfield H, Wieser E, Arsenovic A, Kern R. The Pygae Team: pygae/clifford: v1. 3.1 (2020). *DOI: https://doi.org/10.5281/zenodo*; 1453978.

How to cite this article: and (2022),

APPENDIX

A 3D COORDINATES ALIGNMENT

The predicted coordinates P are relative to a different reference frame compared to T, the ground truth coordinates from the PDB database. P and T must be aligned before evaluating the GDT scores. The goal of alignment is to find $\mathbf{R} \in \mathbb{R}^{3\times3}, t \in \mathbb{R}^3$ such that $P = \mathbf{R}T + t$. In the GA case, we aim at finding the rotor R such that $P = RT\tilde{R}$. We do so by initially centering P, T at the origin (i.e by placing their centre of mass at the origin) and align the two point clouds via characteristic multivector technique [40].

Results in Table 5 are obtained after aligning P, T. We picked the best GDT scores out of two different alignments procedures, namely (i) singular value decomposition (SVG) and (ii) GA-based alignment via characteristic multivectors. Results in Tables 3 -4 , as they refer to 750 proteins, have been obtained only through (i) as it is faster computationally. The two algorithms are summarized in Algorithms 1-2.

Algorithm 1 SVD Alignment

$i \leftarrow 0$
$M \leftarrow 10^3$
while $i \leq M$ **do**

$C_P \leftarrow \frac{1}{N}\sum_i^N P^i$ ▷ compute centroids
$C_T \leftarrow \frac{1}{N}\sum_i^N T^i$

$H \leftarrow (P - C_P)(T - C_T)^T$ ▷ $H \in \mathbb{R}^{3\times3}$
$U, \Sigma, V^T \leftarrow \text{SVD}(H)$
▷ perform SVD

$\mathbf{R} \leftarrow VU^T$
$t \leftarrow C_P - \mathbf{R}C_T$

$T \leftarrow \mathbf{R}T + t$ ▷ rotate and translate T
$i \leftarrow i + 1$
end while

B TRAINING DETAILS

B.1 Graph Transformer + 3D Projector

The total loss to minimize is equal to

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \tag{B1}$$

In which the first term *minimizes* the L_1 norm between $\mathbf{D}$ (the ground truth distance map) and $\tilde{\mathbf{D}}$, as

$$\mathcal{L}_1 = \frac{1}{N^2}\sum_i^N\sum_j^N \|\tilde{\mathbf{D}}_{ij} - \mathbf{D}_{ij}\|_1 \tag{B2}$$

The second term *maximizes* the structural similarity index (SSIM) between $\mathbf{D}$ and $\tilde{\mathbf{D}}$ weighted by an arbitrary coefficient $\alpha = 10$ to make $\mathcal{L}_2$ of the same order of magnitude as $\mathcal{L}_1$:

$$\mathcal{L}_2 = \alpha\left(1 - SSIM\{\mathbf{D}, \tilde{\mathbf{D}}\}\right) = \alpha\left(1 - \frac{(2\mu_{\tilde{\mathbf{D}}}\mu_{\mathbf{D}} + c_1)(2\sigma_{\tilde{\mathbf{D}}\mathbf{D}} + c_2)}{(\mu_{\tilde{\mathbf{D}}}^2 + \mu_{\mathbf{D}}^2 + c_1)(\sigma_{\tilde{\mathbf{D}}}^2 + \sigma_{\mathbf{D}}^2 + c_2)}\right) \tag{B3}$$

where $\mu_{\mathbf{D}}$ is the mean of $\mathbf{D}$, $\mu_{\tilde{\mathbf{D}}}$ the mean of $\tilde{\mathbf{D}}$, $\sigma_{\tilde{\mathbf{D}}\mathbf{D}}$ the covariance of $\tilde{\mathbf{D}}$ and $\mathbf{D}$, $\sigma_{\tilde{\mathbf{D}}}^2$ the variance of $\tilde{\mathbf{D}}$, $\sigma_{\mathbf{D}}^2$ the variance of $\mathbf{D}$, $c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$ with $k_1 = 0.01, k_2 = 0.03$ and L the dynamic range is set to 255.

Algorithm 2 GA Alignment

$i \leftarrow 0$
$M \leftarrow 10^3$
while $i \leq M$ **do**
  $C_P \leftarrow \frac{1}{N}\sum_i^N P^i$ ▷ compute centroids
  $C_T \leftarrow \frac{1}{N}\sum_i^N T^i$

  $F \leftarrow (P - C_P)(T - C_T)^T$
  $G \leftarrow (T - C_T)(T - C_T)^T$ ▷ $F, G \in \mathbb{R}^{3\times3}$

  $f_1, f_2, f_3 \leftarrow F_{:,1}, F_{:,2}, F_{:,3}$ ▷ extract columns of F, G
  $g_1, g_2, g_3 \leftarrow G_{:,1}, G_{:,2}, G_{:,3}$

  $f^1 \leftarrow (f_2 \wedge f_3)/(f_1 \wedge f_2 \wedge f_3)$ ▷ reciprocal frames
  $f^2 \leftarrow (f_1 \wedge f_3)/(f_1 \wedge f_2 \wedge f_3)$
  $f^3 \leftarrow (f_1 \wedge f_2)/(f_1 \wedge f_2 \wedge f_3)$

  $X \leftarrow 1+[(f^1 g_1+f^2 g_2+f^3 g_3)]+[(f^2\wedge f^1)(g_1\wedge g_2)+(f^3\wedge f^2)(g_2\wedge g_3)+(f^3\wedge f^1)(g_1\wedge g_3)+(f^3\wedge f^2\wedge f^1)(g_1\wedge g_2\wedge g_3)]$
  $\alpha \leftarrow X\tilde{X}$
  $\tilde{R} \leftarrow X/\sqrt{\alpha}$

  $T \leftarrow RT\tilde{R}$
  $i \leftarrow i + 1$
end while

Note how the loss is measured over distance maps and not over 3D coordinates as 3D coordinates depend on a reference frame, while distances are rotationally and translationally invariant. It is possible to include an orientational term in the loss (e.g. MAE between original and predicted cost maps or angle maps), but this would require more coordinates to be predicted rather than just the C_α coordinates, which are enough for distance maps.

The model consists of 108813 trainable parameters, of which 108648 are from the GT and 165 are from the projector. The training and testing sets are subsets of PDNET. The first is composed of 200 proteins, while for testing the accuracy of the model we tested it on 5 test sets of 150 proteins each (labelled D1, D2, etc). The optimizer has been set to Adam with exponentially decaying learning rate, with initial learning rate $\eta_0 = 1 \times 10^{-2}$ and decay rate per epoch $\gamma = 0.9$. The GT has $C = 4$ attention heads and $L = 3$ layers. The batch size has been fixed to $B = 1$ and the network has been trained for $E = 5$ epochs, for a total of 1000 training iterations.

Combinations of $\eta \in \{1\times10^{-1}, 1\times10^{-2}, 1\times10^{-3}, 3\times10^{-4}\}$, $E \in \{3, 5, 10\}$, $B \in \{1, 50, 100\}$, $L \in \{3, 6, 10\}$, $C \in \{1, 4, 5\}$, as well as $\mathcal{L} = \mathcal{L}_1$ and additional layers in the 3D projector have also been implemented and tested, but the hyperparameters above were found to be optimal for our problem.

The code has been written as a Jupyter Notebook on Google Colaboratory, run on an NVIDIA Tesla K80 GPU and it uses PyTorch for the DL architecture, the Clifford library for GA operations[41] and the PDB Module of Biopython for handling protein data. The GT was derived from[17]. Scripts and datasets are available upon request to the authors.

B.2 PDNET Pipeline

The loss of the PDNET pipeline shown in Section 7.2 has been chosen to be

$$\mathcal{L}_{\mathbf{X}} = \log\left(\cosh\left(\mathbf{X}_P - \mathbf{X}_T\right)\right) \tag{B4}$$

where $\mathbf{X}_P, \mathbf{X}_T$ are the predicted and true orientational maps in the training set, respectively, with $\mathbf{X} = \{\mathbf{M}, \mathbf{N}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Omega}\}$. The only difference between the approaches is the activation function of the last layer: when predicting $\mathbf{N}$, which ranges between $[-0.5, 0.5]$, we employed a *tanh* activation instead of a *ReLU*.

The training features are identical to those of PDNET, namely a stack of images of the type $\{\mathbf{Y}^{(i)}\}_{i=1}^{N}$, with $L = 57$ and $\mathbf{Y}^{(i)} \in \mathbb{R}^{M\times M}$, in which M is the length of the protein sequence. The loss is evaluated per pixel. The training set has been kept to 1000 proteins from the DEEPCOV dataset, and the testing set to 150 proteins from the PSICOV dataset, as in the original PDNET pipeline. The code has been implemented similarly to B.1.

Trained models, datasets and results are available upon request to the authors.