6DCNN with roto-translational convolution filters for volumetric data processing

Dmitrii Zhemchuzhnikov,Ilia Igashov, Sergei Grudinin

Univ. Grenoble Alpes, CNRS, Grenoble INP, LJK
Dmitrii.Zhemchuzhnikov@univ-grenoble-alpes.fr, Ilia.Igashov@epfl.ch, Sergei.Grudinin@univ-grenoble-alpes.fr

Abstract

In this work, we introduce 6D Convolutional Neural Network (6DCNN) designed to tackle the problem of detecting relative positions and orientations of local patterns when processing three-dimensional volumetric data. 6DCNN also includes SE(3)-equivariant message-passing and nonlinear activation operations constructed in the Fourier space. Working in the Fourier space allows significantly reducing the computational complexity of our operations. We demonstrate the properties of the 6D convolution and its efficiency in the recognition of spatial patterns. We also assess the 6DCNN model on several datasets from the recent CASP protein structure prediction challenges. Here, 6DCNN improves over the baseline architecture and also outperforms the state-of-the-art.

Introduction

Methods of deep learning have recently made a great leap forward in the spatial data processing. This domain contains various tasks from different areas of industry and natural sciences, including three-dimensional (3D) data analysis. For a long time, convolutional neural networks (CNNs) remained the main tool in this domain. CNNs helped to solve many real-world challenges, especially in computer vision. However, these architectures have rather strict application restrictions. Unfortunately, real-world raw data rarely have standard orientation and size, which limits the efficiency of translational convolutions. This circumstance has created an increased interest in the topic of SE(3)-equivariant operations in recent years. While most of SE(3)-equivariant methods are focused on learning relations between rotational invariants, this paper addresses the problem of recognition of arbitrarily-positioned and -oriented volumetric patterns by reusing some theory already developed in computational physics and crystallography.

A volumetric pattern in 3D has six degrees of freedom (DOFs), three to define a rotation, and three for a translation. Thus, a classical convolution technique would require scanning through all these six DOFs and scale as $O(N^3M^6)$ if a brute-force computation is used, where N is the linear size of the volumetric data, and M is the linear size of the pattern.

This work proposes a set of novel operations with the corresponding architecture based on six-dimensional (6D) *roto-translational* convolutional filters. For the first time, thanks to the polynomial expansions in the Fourier space, we demonstrate the feasibility of the 6D roto-translational-based convolutional network with the leading complexity of $O(N^2M^4)$ operations. We tested our method on simulated data and also on protein structure prediction datasets, where the overall accuracy of our predictions is on par with the state-of-the-art methods. Proteins play a crucial role in most biological processes. Despite their seeming complexity, structures of proteins attract more and more attention from the data science community (Senior et al. 2020; Jumper et al. 2021; Laine et al. 2021). In particular, the task of protein structure prediction and analysis raises the challenge of constructing rotational and translational equivariant architectures.

State of the art / Related work

Equivariant operations. The first attempt of learning rotation-equivariant representations was made in Harmonic Networks (Worrall et al. 2017) in application to 2D images. Further, this idea was transferred to the 3D space with the corresponding architecture known as 3D Steerable CNNs (Weiler et al. 2018). In Spherical CNNs (Cohen et al. 2018), the authors introduced a correlation on the rotation group and proposed a concept of rotation-equivariant CNNs on a sphere. Spherical harmonics kernels that provide rotational invariance have also been applied to point-cloud data (Poulenard et al. 2019). A further effort on leveraging compact group representations resulted in the range of methods based on Clebsh-Gordan coefficients (Kondor 2018; Kondor, Lin, and Trivedi 2018; Anderson, Hy, and Kondor 2019). This approach was finally generalized in Tensor field networks (Thomas et al. 2018), where rotation-equivariant operations were applied to vector and tensor fields. Later, SE(3)-Transformers (Fuchs et al. 2020) were proposed to efficiently capture the distant spatial relationships. More recently, (Hutchinson et al. 2021) continued to develop the theory of equiv-

ariant convolution operations for homogeneous spaces and proposed Lie group equivariant transformers, following works on the general theory of group equivariant operations in SO(2) (Romero et al. 2020; Romero and Cordonnier 2021) and SO(3) (Cohen, Geiger, and Weiler 2018). Equivariant operations have been also applied to gauge fields (Cohen et al. 2019). We should mention that the most common data representation in this domain is a 3D point cloud, however, several approaches operate on regular 3D grids (Weiler et al. 2018; Pagès, Charmettant, and Grudinin 2019). We should also add that some of the above-mentioned methods (Cohen et al. 2018; Weiler et al. 2018; Kondor 2018; Anderson, Hy, and Kondor 2019) employ Fourier transform in order to learn rotation-equivariant representations.

Geometric learning on molecules. As graphs and point clouds are natural structures for representing molecules, it is reasonable that geometric learning methods have been actively evolving especially in application to biology, chemistry, and physics. More classical graph-learning methods for molecules include MPNNs (Gilmer et al. 2017), SchNet (Schütt et al. 2017), and MEGNet (Chen et al. 2019). Ideas for efficient capturing of spatial relations in molecules resulted in rotation-invariant message-passing methods DimeNet (Klicpera, Groß, and Günnemann 2020) and DimeNet++ (Klicpera et al. 2020). Extending message-passing mechanism with rotationally equivariant representations, polarizable atom interaction neural networks (Schütt, Unke, and Gastegger 2021) managed to efficiently predict tensorial properties of molecules. (Satorras, Hoogeboom, and Welling 2021) proposed E(n) equivariant GNNs for predicting molecular properties and later used them for developing generative models equivariant to Euclidean symmetries (Satorras et al. 2021).

Proteins are much bigger and more complex systems than small molecules but are composed of repeating blocks. Therefore, more efficient and slightly different methods are required to operate on them. A very good example is the two recent and very powerful methods AlphaFold2 (Jumper et al. 2021) and RoseTTAFold (Baek et al. 2021). Most recent geometric learning methods designed for proteins include deep convolutional networks processing either volumetric data in local coordinate frames (Pagès, Charmettant, and Grudinin 2019; Hiranuma et al. 2021), graph neural networks (Ingraham et al. 2019; Sanyal et al. 2020; Baldassarre et al. 2021; Igashov et al. 2021; Igashov, Pavlichenko, and Grudinin 2021), deep learning methods on surfaces and point clouds (Gainza et al. 2020; Sverrisson et al. 2020), and geometric vector perceptrons (Jing et al. 2021b,a). In addition, several attempts were made to scale tensor-field-like SE(3)-equivariant methods to proteins (Derevyanko and Lamoureux 2019; Eismann et al. 2020; Townshend et al. 2020; Baek et al. 2021).

Model/Method

Workflow

Here, we give a brief description of all steps of our method that are described in more detail below. Firstly, for each residue in the input protein molecule, we construct a function $\mathbf{f}(\vec{r})$ that describes its local environment. More technically, this function is a set of 3D Gaussian-shaped features centered on the location of atoms within a certain distance R_{max} from the C_α atom of the corresponding residue (see Fig. 1A).

Then, for each function $\mathbf{f}(\vec{r})$, we compute its spherical Fourier expansion coefficients $\mathbf{F}_l^k(\rho)$. The angular resolution of the expansion is determined by the maximum order of spherical harmonics L. The radial resolution of the expansion corresponds to the maximum reciprocal distance ρ_{max} and is inversely proportional to the resolution σ of the real-space Gaussian features as $\rho_{max} = \pi/\sigma$ (see Fig. 1B). Similarly, the radial spacing between the reciprocal points is inversely proportional to the linear size of the data, $\Delta\rho = \pi/(2R_{max})$. Without loss of generality, we can set the number of reciprocal radial points to be equal L, such that $\rho_{max}/\Delta\rho = L + 1 = 2R_{max}/\sigma$.

Spherical Fourier coefficients $\mathbf{F}_l^k(\rho)$ constitute the input for our network, along with the information about the transition from the coordinate system of one residue to another. We start the network with the embedding block that reduces the dimensionality of the feature space. Then, we apply a series of 6D convolution blocks that consist of 6D convolution, normalization, and activation layers, followed by a message-passing layer. After a series of operations on continuous data, we switch to the discrete representation and continue the network with graph convolutional layers (see Fig. 1C-D). In the graph representation, each node corresponds to a protein residue, and a graph edge links nodes if the distance between the corresponding C_α atoms is smaller than a certain threshold R_n. We should also mention that the backbone structure of a protein residue can be used to unambiguously define its local coordinate system (Pagès, Charmettant, and Grudinin 2019; Jumper et al. 2021) using the Gram–Schmidt orthogonalization process starting from $C_\alpha - N$ and $C_\alpha - C$ vectors.

Representation of volumetric data

Let us consider a function $\mathbf{f}(\vec{r}) : \mathcal{R}^3 \rightarrow \mathcal{R}^{d_f}$ that describes a distribution of d_f-dimensional features in the 3D space. Very often, initial data is given as a point cloud, as it is typically the case for protein structures. Let us consider a set of points located within a maximum radius R_{max} at positions $\vec{r}_1, ..., \vec{r}_n, ..., \vec{r}_N$ with the corresponding feature vectors $\mathbf{t}_1, ..., \mathbf{t}_n, ..., \mathbf{t}_N$. To convert this representation to a continuous one, we assume that each point feature has a Gaussian shape with a standard deviation σ. Then, the continuous function characterizing this area will have the form,

$$\mathbf{f}(\vec{r}) = \sum_{n=1}^{N} \mathbf{f}_n(\vec{r}) = \sum_{n=1}^{N} \mathbf{t}_n \exp(-\frac{(\vec{r} - \vec{r}_n)}{2\sigma^2}), \qquad (1)$$

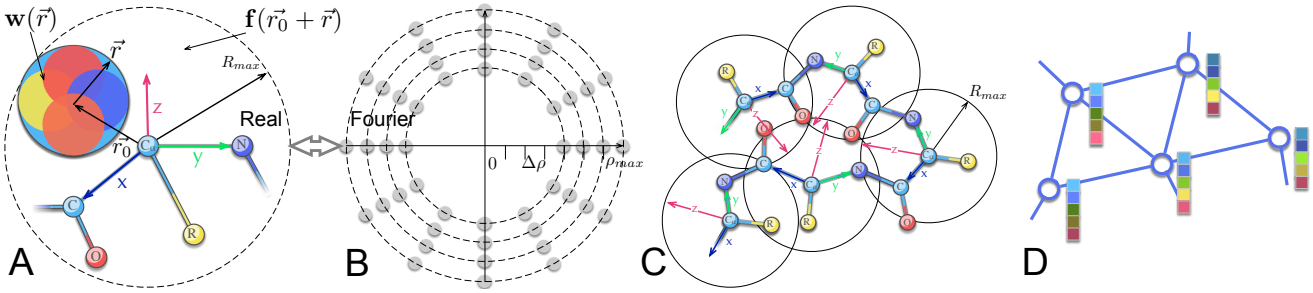

Figure 1: **A**. Six-dimensional (6D) convolution between a filter $\mathbf{w}(\vec{r})$ and a function $\mathbf{f}(\vec{r_0}+\vec{r})$. The function $\mathbf{f}(\vec{r_0}+\vec{r})$ describes the local environment of a protein residue and is defined within a certain radius R_{max} from the corresponding C_α atom. The local coordinate system xyz is built on the backbone atoms C_α, C, N of each protein residue. R denotes the location of a residue's side-chain. **B**. The spherical Fourier space with the reciprocal spacing $\Delta\rho$ and the maximum resolution of ρ_{max}. Grey dots schematically illustrate points where the Fourier image is stored. **C**. An illustration of a protein chain representation. Each protein residue has its own coordinate system xyz and the corresponding local volumetric description $\mathbf{F}_l^k(\rho)$ within a certain sphere of R_{max} radius. Spheres of different residues may overlap. Two residues are considered as neighbors in the graph representation if their C_α atoms are located within a certain threshold R_n. **D**. The graph representation of the protein structure. The node features are learned by the network and are represented with colored rectangles. The edge features are assigned based on the types of the corresponding residues and the topological distance of the protein graph.

where σ is the *spatial resolution* of the features. It is very convenient to use the Fourier description of this function in spherical coordinates. The spherical harmonics expansion of the Fourier transform of function $\mathbf{f}(\vec{r})$ will be

$$\mathbf{F}_l^k(\rho) = 4\pi(-i)^l \sum_{n=1}^{N} j_l(\rho r_n)\overline{Y_l^k(\Omega_{r_n})} \\ (\sqrt{2\pi}\sigma)^3 \exp(-\frac{\sigma^2\rho^2}{2})\mathbf{t}_n, \tag{2}$$

where ρ is the *reciprocal distance*. Please see Appendices A-E for more details.

6D convolution operation

The initial idea that prompted us to consider non-traditional types of convolution is the intention to perceive spatial patterns in whatever orientations they have. In a classical 3D convolution, when a filter has learned a pattern in a particular orientation, a different orientation of this pattern may result in a lower response to the same filter in the inference mode. Keeping this in mind, we came with the idea to extend the convolution with an integration of all possible filter rotations. Let $\mathbf{f}(\vec{r}) : \mathcal{R}^3 \to \mathcal{R}^{d_i}$ and $\mathbf{w}(\vec{r}) : \mathcal{R}^3 \to \mathcal{R}^{d_i} \times \mathcal{R}^{d_o}$ be the initial signal and a spatial filter, correspondingly. We propose to extend the classical convolution as follows,

$$\int_{\vec{r}} d\vec{r}\, \mathbf{f}(\vec{r_0}+\vec{r})\mathbf{w}(\vec{r}) \to \int_{\Lambda} d\Lambda \int_{\vec{r}} d\vec{r}\, \mathbf{f}(\vec{r_0}+\Lambda^{-1}\vec{r})\mathbf{w}(\Lambda\vec{r}), \tag{3}$$

where $\Lambda \in \mathrm{SO}(3)$ is a 3D rotation. Please see Fig. 1A for an illustration. Let the functions $\mathbf{f}(\vec{r})$ and $\mathbf{w}(\vec{r})$ be *finite-resolution* and have spherical Fourier expansion coefficients $F_l^k(\rho)$ and $W_l^k(\rho)$, correspondingly, which are nonzero for $l \leq$ than some maximum expansion coefficient L. Then, the result of the 6D convolution has the following coefficients,

$$[\mathbf{F}_{\text{out}}]_l^k(\rho) = \sum_{l_1=0}^{L} \sum_{k_1=-l_1}^{l_1} \frac{8\pi^2}{2l_1+1} \mathbf{W}_{l_1}^{-k_1}(\rho) \\ \sum_{l_2=|l-l_1|}^{l+l_1} c^{l_2}(l,k,l_1,-k_1)\mathbf{F}_{l_2}^{k+k_1}(\rho), \tag{4}$$

where c^l are the products of three spherical harmonics, see Eq. 26 in Appendix D. The proof can be found in Appendix F. For a single reciprocal distance ρ, the complexity of this operation is $O(L^5)$, where L is the maximum order of the spherical harmonics expansion.

Nonparametric message passing of continuous data

Let us assume that our spatial data can be represented with overlapping spatial fragments, each having its own local coordinate frame, as it is shown in Fig. 1C for the continuous representation of a protein molecule. Then, we can recompute the fragment representation in the neighboring coordinate frames using spatial transformation operators. For this purpose, we designed a message passing operation in a form that recomputes the spherical Fourier coefficients in a new reference frame. We should specifically note that this operation is SE(3)-equivariant by construction, because the spatial relationship between molecular fragments remain the same when rotating and shifting the global coordinate system. We decompose such a spatial transformation in a sequence of a rotation followed by a z-axis translation followed by the second rotation. Indeed, the spherical

Fourier basis provides low computational complexity for the z-translation and rotation operations. Let function $\mathbf{f}_z(\vec{r})$ be the results of translating function $\mathbf{f}(\vec{r})$ along the z-axis by an amount Δ. Then, the expansion coefficients of these two functions will have the following relation,

$$[\mathbf{F}_z]_l^k(\rho) = \sum_{l'=k}^{L} T_{l,l'}^k(\rho,\Delta)\mathbf{F}_{l'}^k(\rho) + O\left(\frac{1}{(L-l)!}(\frac{\rho\Delta}{2})^{L-l}\right), \tag{5}$$

where $T_{l,l'}^k$ is a translation tensor specified in Appendix D. The update of all the expansion coefficients costs $O(L^4)$ operations.

Similarly, let function $\mathbf{f}_\Lambda(\vec{r})$ be the rotation of function $\mathbf{f}(\vec{r})$ by an amount $\Lambda \in \mathrm{SO}(3)$, $\mathbf{f}_\Lambda(\vec{r}) = \mathbf{f}(\Lambda\vec{r})$. The corresponding expansion coefficients are then related as

$$[\mathbf{F}_\Lambda]_l^k(\rho) = \sum_{k'=-l}^{l} D_{k',k}^l(\Lambda)\mathbf{F}_l^{k'}(\rho), \tag{6}$$

where $D_{k',k}^l$ is a rotation Wigner matrix specified in Appendix C. The update of all the expansion coefficients will again cost $O(L^4)$ operations.

Normalization

Since we are working with continuous data in the Fourier space, we have to introduce our own activation and normalization functions. When developing the normalization, we proceeded from very basic premises. More precisely, we normalize the signal $\mathbf{f}(\vec{r})$ by setting its mean to zero and its variance to unity. This can be achieved if the following operations are performed on the spherical Fourier expansion coefficients of the initial function,

$$[\mathbf{F}_n]_l^k(\rho) = \begin{cases} 0, & \text{if } l = k = \rho = 0, \\ [\mathbf{F}]_l^k(\rho)/\mathbf{S_2}, & \text{otherwise,} \end{cases} \tag{7}$$

where $\mathbf{S_1} = \int_{\mathcal{R}^3} \mathbf{f}(\vec{r})d\vec{r}$, $\mathbf{S_2} = \int_{\mathcal{R}^3}(\mathbf{f}(\vec{r}) - \mathbf{S_1})^2 d\vec{r}$. We should also notice that we apply the element-wise division in Eq. 7. The proof can be found in Appendix G.

Activation

The concept of our activation operation coincides with the idea of the classical activation in neural networks, i.e. to nonlinearly transform the initial signal depending on how it differs from the bias. Let the initial signal be $\mathbf{f}(\vec{r})$, and the bias signal with trainable Fourier coefficients be $\mathbf{b}(\vec{r})$. Then, we propose the following output of the normalization-activation block,

$$\mathbf{f}_a(\vec{r}) = \left(\frac{1}{4}\int_{\mathbb{R}^3}(N(\mathbf{f}(\vec{r}) + \mathbf{b}(\vec{r})) - N(\mathbf{b}(\vec{r})))^2 d^3\vec{r}\right) N(\mathbf{f}(\vec{r}) + \mathbf{b}(\vec{r})), \tag{8}$$

where $N()$ is the normalization operation defined in Eq. 7 such that the value $\frac{1}{4}\int_{\mathbb{R}^3}(N(\mathbf{f}(\vec{r}) + \mathbf{b}(\vec{r})) - N(\mathbf{b}(\vec{r})))^2 d^3\vec{r}$ lies in the interval $[0, 1]$. If the signal $\mathbf{f}(\vec{r})$ is 'coherent' to $\mathbf{b}(\vec{r})$, $\mathbf{f}(\vec{r}) = K\mathbf{b}(\vec{r}), K > 0$, then it does not pass the block. The amplification factor reaches its maximum value when the two signals are anti-coherent. Parseval's theorem allows us to translate these formulas into operations on their expansion coefficients,

$$[\mathbf{F}_a]_l^k(\rho) = \frac{1}{4}\left(\sum_{l'=0}^{L}\sum_{k'=-l'}^{l'}\int_0^\infty (N(\mathbf{F}_{l'}^{k'}(\rho) + \mathbf{B}_{l'}^{k'}(\rho)) - N(\mathbf{B}_{l'}^{k'}(\rho)))^2\rho^2 d\rho\right) N(\mathbf{F}_l^k(\rho) + \mathbf{B}_l^k(\rho)), \tag{9}$$

where $[\mathbf{F}_a]_l^k(\rho_p)$ and $\mathbf{B}_l^k(\rho_p)$ are the expansion coefficients of functions $\mathbf{f}(\vec{r})$ and $\mathbf{b}(\vec{r})$, correspondingly.

Switching the representations

For most of the real-world tasks that may require the proposed architecture, a transition from a continuous functional representation $\mathbf{f}(\vec{r}) : \mathcal{R}^3 \to \mathcal{R}^{d_f}$ to a discrete vector representation $\mathbf{h} \in \mathcal{R}^{d_f}$ is necessary. There can be several ways to achieve it. In the simplest case, when the input function $\mathbf{f}(\vec{r})$ can be unambiguously associated with some reference coordinate system, as in the case of protein's peptide chain, we may use the following operation,

$$\mathbf{h} = \int_{\mathcal{R}^3} \mathbf{f}(\vec{r})\mathbf{w}(\vec{r})d\vec{r}, \tag{10}$$

where $\mathbf{w}(\vec{r}) : \mathcal{R}^3 \to \mathcal{R}^{d_f}$ is a filter function element-wise multiplied with the input function. If the functions $\mathbf{f}(\vec{r})$ and $\mathbf{w}(\vec{r})$ have corresponding expansion coefficients $\mathbf{F}_l^k(\rho_p)$ and $\mathbf{W}_l^k(\rho_p)$, then this operation will have the following form (please see more details in Appendix H),

$$\mathbf{h} = \sum_{l=0}^{L}\sum_{k=-l}^{l}\int_0^\infty \mathbf{F}_l^k(\rho)\overline{\mathbf{W}}_l^k(\rho)\rho^2 d\rho. \tag{11}$$

Graph convolutions layers

In our model, the continuous representation is followed by the classical graph convolutional layers (see Fig. 1C-D). Indeed, protein structures, on which we assess our model, allow us to use the graph representation. In such a graph, each node corresponds to an amino acid residue and characterizes the 3D structure of its neighborhood, and each edge between two nodes indicates their spatial proximity, i.e. the distance between the corresponding C-alpha atoms within a certain threshold R_n (see Fig. 1D).

Let us consider a graph $\mathcal{G}$ that is described by the feature matrix $\mathbf{H} \in \mathcal{R}^{N\times d_v}$, where N is the number of graph nodes and d_v is the dimensionality of the node feature space, and the adjacency matrix $\mathbf{A} \in \mathcal{R}^{N\times N\times d_e}$, where d_e is the dimensionality of the edge feature space. We decided to use one-hot edge features that would encode the types of amino acids of the associated nodes. To reduce the dimensionality of the edge feature space

to d'_e, we use the following trainable embedding $\mathbf{R}_e \in \mathcal{R}^{d_e \times d'_e}$, and a reduced adjacency matrix $\mathbf{A}_r = \mathbf{A}\mathbf{R}_e$. Finally, the graph convolution step is defined as

$$\mathbf{H}_{k+1} = \sigma_a(\mathbf{A}_r\mathbf{H}_k\mathbf{W}_k + \mathbf{H}_k\mathbf{W}^s_k + \mathbf{b}_k), \qquad (12)$$

where $\mathbf{W}_k \in \mathcal{R}^{d_k \times d_{k+1} \times d'_e}$ and $\mathbf{W}^s_k \in \mathcal{R}^{d_k \times d_{k+1}}$ are trainable matrices.

Experiments

6D filters

Our first step was to study the properties of the 6D roto-translational convolution operation. To do so, we generated a small 3D pattern, $\mathbf{f}(\vec{r})$, composed of six 3D Gaussians with $\sigma = 0.4$ Å shown in Fig. 2A-B. We then created a function $\mathbf{h}(\vec{r})$, rotated and translated $\mathbf{f}(\vec{r})$ to a new location. Figures 2C-D show the result of the 6D convolution between $\mathbf{h}(\vec{r})$ and $\mathbf{f}(\vec{r})$ given by Eq. 4. As we can expect, the maximum of this convolution corresponds to the position of the center of mass of function $\mathbf{h}(\vec{r})$, and the value of the convolution reduces as we go further from this point. We used the following parameters for this experiment, $\sigma = 0.4$ Å, $L = 4$, $\rho_{\max} = 0.6\pi$ Å^{-1}, $\Delta\rho = 0.2\pi$ Å^{-1}.

We then compared the result of the classical 3D convolution (see Appendix B) with the proposed 6D convolution. Using the same initial pattern, randomly rotated multiple times, we recorded the maximum positional error in determining the center of mass of the translated pattern with respect to the maximum expansion order L. As we can see in Fig. 2E, the 6D convolution detects the position of the shifted pattern more accurately compared to its classical 3D counterpart, and the accuracy increases with the maximum expansion order L.

Message passing and translation operator

Our next experiment was the assessment of the message-passing step. Our main goal was to study the conditions of validity for the translation operator 28 given in Appendix D, which is also implicitly used in the 6D convolution part. Specifically, we were interested in whether the result of the 6D convolution will be preserved after changing the coordinate systems. For this experiment, we used the same volumetric pattern $\mathbf{f}(\vec{r})$ described above and shown in Fig. 2A-B. We then shifted this pattern to a new location and recorded the value of the 6D convolution, as described above, shown in Fig. 2C-D. For the comparison, we computed the same 6D convolution from a different coordinated system, shifted by $3\sqrt{3}$ Å from the original one. We used parameters from the previous experiment. The result is shown in Fig. 2G-H. Both convolution functions have their maximums near the location of the center of mass of the shifted pattern, however, their volumetric shapes are slightly different.

For a more rigorous experiment, we examined the relative error of the translation operator 28 (Appendix D) as a function of the displacement of the coordinate system and the expansion order. Here, we fixed parameters to $\sigma = 0.4$ Å, $\rho_{\max} = 0.6\pi$ Å^{-1}, $\Delta\rho = 0.2\pi$ Å^{-1}, and varied the value of L. From the results shown in Fig. 2F we can see that for the displacements within about $\sigma L/2$, the error is negligibly small, which is the consequence of the Nyquist–Shannon theorem.

Technical details

Amino-acid residues can contain atoms of 167 distinguishable types. We have also included one additional type for the solvent molecules. Overall, the dimensionality of the volumetric function characterizing the protein model $N_a = 168$. We choose some maximum value of the radius R_{max}, which limits the set of atoms that fall in the amino-residue neighborhood. We also choose some parameter R_n, which is the maximum distance between amino residues that considered as neighbors in the graph.

We use 20 amino acids types. The edge between two amino acids can be determined by a vector of the size $20 \times 20 + d_t$, where the first 20×20 elements are a one-hot representation of the amino acids pair, with the distinguishable order of residues in a pair. The last d_t elements in this vector is a one-hot representation of the topological distance between the residues in the protein graph. The values of R_{max}, R_n, d_t, and L are the hyperparameters of the network that were optimized with a grid search.

Baseline architecture

For the comparison, we introduced a baseline architecture that would help us to assess the novel layers. It begins with trainable embedding in the feature space. We then applied the transition from the continuous to the discrete representation using operation 11 that is followed by three graph convolutional layers described in Eq. 12. For the activation, we used the tanh function in the last layer and the LReLU function with the 'leak' parameter of 0.05 in all other layers. We also introduced two trainable parameters μ_t and σ_t for the mean and the standard deviation of the local rates in the training sample. The relationship between the output of the last layer l_N and the output of the network o is $o = \sigma_t l_N + \mu_t$. Overall, the baseline architecture had 21,026 trainable parameters.

Training. This network was trained on CASP 8-11 datasets (see Appendix J for more detail) in $1,280$ iterations. At each iteration, the network was fed with 16 input protein models. One training iteration took $\approx$ 5 minutes on Intel ©Xeon(R) CPU E5-2630 v4 @ 2.20GHz. We used a composite loss function that is described in Appendix L.

Hyperparameters. The network has the following hyperparameters, $\sigma = 2$ Å, $R_n = 12$ Å, $d_t = 10$, $L = 4$, and $R_{max} = 8$ Å.

6DCNN networks

The main difference between the baseline architecture and the 6DCNN networks is the presence of 6D convolution layers in the latter. Our first architecture (6DCNN-1) contains only one 6D convolution layer. The second

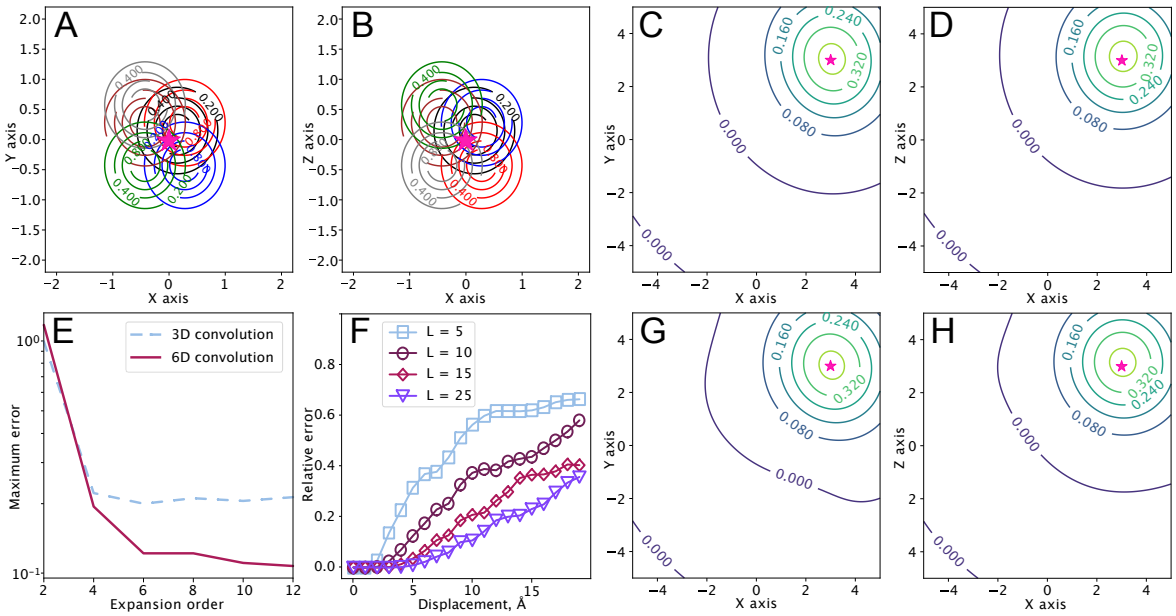

Figure 2: **A-B.** Six Gaussian volumetric features of $\sigma = 0.4$ Å shown in the xy-plane (A) and xz-plane (B). The center of mass of the whole pattern is shown with the pink star. **C-D.** The density map of the resultant 6D convolution between an original pattern and its translated and rotated copy to the pink star position, shown in the xy-plane (C) and xz-plane (D). **E.** The maximum error in determining the center of the translated and rotated pattern as a function of the expansion order L divided by the Gaussian feature σ for 3D and 6D convolutions. **F.** Relative error of the recovering volumetric patterns as a function of the translation amplitude and the expansion order. **G-H.** Results of the 6D convolution in the translated coordinate system, by $3\sqrt{3}$ Å, shown in the xy-plane (G) and xz-plane (H).

architecture (6DCNN-2) has two 6DCNN layers. The 6DCNN layer is composed of the following operations: 6D convolution, followed by normalization and activation. Consecutive 6D convolution layers are linked with the message passing step. Table 3 lists the architectures of the networks. Overall, the 6DCNN-1 and 6DCNN-2 architectures had 187,326 and 283,426 trainable parameters, correspondingly. Figure 3A in Appendix N shows real-space projections of two 6D convolution filters learned by 6DCNN-1.

Training. The two networks were trained on CASP 8-11 datasets (see Appendix J for more detail) in $1,280$ iterations. At each iteration, the networks were fed with 16 protein models. One training iteration took ≈ 6 minutes for 6DCNN-1 and ≈ 15 minutes for 6DCNN-2 on Intel ©Xeon(R) CPU E5-2630 v4 @ 2.20GHz. Figure 3D in Appendix N demonstrates the learning curves on the validation dataset of three architectures, baseline, 6DCNN-1, and 6DCNN-2. We used the same loss function as for the baseline architecture.

Hyperparameters. The networks have the following hyperparameters, $\sigma = 2$ Å, $R_n = 12$ Å, $d_t = 10$, $L = 4$, and $R_{max} = 8$ Å.

CASP results

In order to assess the 6DCNN architectures, we compared their performance on the CASP12 (Table 1) and CASP13 (Table 2) datasets, described in Appendix J, with the baseline model and also with the state-of-the-art single-model quality assessment methods SBROD, SVMQA, VoroCNN, Ornate, ProQ3, and VoroMQA (Cheng et al. 2019). SBROD is a linear regression model that uses geometric features of the protein backbone (Karasikov, Pagès, and Grudinin 2019). SVMQA is a support-vector-machine-based method that also uses structural features (Manavalan and Lee 2017). VoroMQA engages statistical features of 3D Voronoi tessellation of the protein structure (Olechnovič and Venclovas 2017). VoroCNN is a graph neural network built on the 3D Voronoi tessellation of protein structures (Igashov et al. 2021). Ornate is a convolutional neural network that uses 3D volumetric representation of protein residues in their local reference frames (Pagès, Charmettant, and Grudinin 2019). ProQ3 is a fully connected neural network operating on the precomputed descriptors (Uziela et al. 2016). We computed the ground-truth lDDT values ourselves. Therefore, we were forced to limit the datasets to only those models that had publicly available target structures. As a result, the CASP12 dataset turned out to be significantly bigger than CASP13, with more demonstrative and representative results.

On the CASP12 test set, we achieved a noticeable improvement in comparison with the state-of-the-art meth-

Method	z-score	Global			Per-target		
		R^2	Pearson, r	Spearman, ρ	R^2	Pearson, r	Spearman, ρ
SBROD	1,29	-33,66	0,55	0,54	-325,18	0,76	0,67
SVMQA	**1, 48**	0,32	0,82	0,80	-2,44	0,76	**0, 73**
VoroCNN	1,39	0,61	0,80	0,80	-2,79	0,73	0,69
Ornate	1,42	0,36	0,78	0,77	-5,14	0,73	0,69
ProQ3	1,18	0,26	0,74	0,77	-4,85	0,73	0,68
VoroMQA	1,18	-0,25	0,59	0,62	-6,23	0,74	0,70
Baseline	1,34	0,55	0,82	0,82	-2,35	0,78	0,71
6DCNN-1	1,26	0,57	0,81	0,81	-2,29	**0, 80**	0, 73
6DCNN-2	1,19	**0, 63**	**0, 85**	**0, 84**	**−1, 95**	0,79	0,71

Table 1: Comparison of the 6DCNN networks with the baseline architecture and the state-of-the-art methods on the unrefined CASP12 stage2 dataset. The best value for each metric (see Appendix K) is highlighted in bold.

Method	z-score	Global			Per-target		
		R^2	Pearson, r	Spearman, ρ	R^2	Pearson, r	Spearman, ρ
SBROD	1,23	0,07	0,72	0,69	-1,44	0,81	0,74
VoroCNN	1,15	**0, 67**	**0, 84**	**0, 82**	0,03	0,79	0,77
VoroMQA	1,32	0,20	0,77	0,79	-1,10	0,79	0,75
ProQ3D	**1, 39**	-0,03	0,75	0,75	-2,07	0,76	0,72
Ornate	0.93	0,26	0.62	0.67	-2,10	0,78	0,77
Baseline	1,35	0,44	0,78	0,77	-0,30	0,82	0,77
6DCNN-1	1,03	0,59	0,82	0,80	-0,02	**0, 83**	**0, 79**
6DCNN-2	1,30	0,56	0,79	0,78	-0,12	0,82	0,77

Table 2: Comparison of the 6DCNN networks with the baseline architecture and the state-of-the-art methods on the unrefined CASP13 stage2 dataset. The best value for each metric (see Appendix K) is highlighted in bold.

ods. Even though the difference between the 6DCNN networks and the baseline model performance is not big, one of the 6DCNN architectures outperforms the baseline in every metric except for the z-score. We can also notice that the 6DCNN-2 method gives significantly higher global correlations and R^2 metrics on the CASP12 dataset than 6DCNN-1 and all other methods. However, 6DCNN-1 demonstrates better per-target correlations on CASP12 data than 6DCNN-2. Both of the networks have higher per-target correlations than most of the state-of-the-art methods. Unfortunately, we did not manage to achieve satisfying performance on the z-score metric. However, z-scores are rather noisy compared to correlations, and not directly linked to the optimized loss function. The fact that 6DCNN-2 has better global correlation scores confirms the importance of the additional 6D correlation block. Figures 3 (B-C) in Appendix N show correlations between the ground-truth global scores from the CASP12 dataset and the corresponding predictions by the two 6DCNN models. The 6DCNN-2 map has a higher density near the diagonal, indicating a better absolute predictions of global scores and a better R^2 metric.

On the CASP13 dataset, we did not greatly outperform the state-of-the-art methods (see Table 2). However, we reached a performance that is on par with the state of the art. Moreover, we should notice that 6DCNN-2 did not outperform 6DCNN-1. This can be explained by the fact that we trained our models on CASP[8-11] datasets, which are rather different from CASP13, and also that the CASP13 dataset is less representative than CASP12.

Table 4 in Appendix M lists Spearman rank correlations of local quality predictions with the corresponding ground-truth values of our networks and the state-of-the-art methods on model structures of 11 targets from CASP13. For the comparison, we chose only those models that had both publicly available target structures and local score predictions by all other methods. As we did not have these predictions for the CASP12 dataset, we limited local score evaluation by CASP13 data only. Here, we did not achieve the best results that could be explained by the small size of the dataset.

Conclusion

This work presents a theoretical foundation for 6D roto-translational spatial patterns detection and the construction of neural network architectures for learning on spatial continuous data in 3D. We built several networks that consisted of 6DCNN blocks followed by GCNN layers specifically designed for 3D models of protein structures. We then tested them on the CASP datasets from the community-wide protein structure prediction challenge. Our results demonstrate that 6DCNN blocks are able to accurately learn local spatial patterns and improve the quality prediction of protein models. The current network architecture can be extended in multiple directions, for example, including the attention mechanism.

Acknowledgements

This work has been partly supported by MIAI Grenoble Alpes (ANR-19-P3IA-0003). We thank Kliment Olechnovic from Vilnius University Life Sciences Center for his valuable comments during the development of the method.

References

Anderson, B.; Hy, T.-S.; and Kondor, R. 2019. Cormorant: Covariant molecular neural networks. *arXiv preprint arXiv:1906.04015*.

Baddour, N. 2010. Operational and convolution properties of three-dimensional Fourier transforms in spherical polar coordinates. *J. Opt. Soc. Am. A*, 27(10): 2144–2155.

Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G. R.; Wang, J.; Cong, Q.; Kinch, L. N.; Schaeffer, R. D.; et al. 2021. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*.

Baldassarre, F.; Menéndez Hurtado, D.; Elofsson, A.; and Azizpour, H. 2021. GraphQA: protein model quality assessment using graph convolutional networks. *Bioinformatics*, 37(3): 360–366.

Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; and Ong, S. P. 2019. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9): 3564–3572.

Cheng, J.; Choe, M.-H.; Elofsson, A.; Han, K.-S.; Hou, J.; Maghrabi, A. H. A.; McGuffin, L. J.; Menéndez-Hurtado, D.; Olechnovič, K.; Schwede, T.; Studer, G.; Uziela, K.; Venclovas, Č.; and Wallner, B. 2019. Estimation of model accuracy in CASP13. *Proteins*, 87(12): 1361–1377.

Cohen, T.; Geiger, M.; and Weiler, M. 2018. A General Theory of Equivariant CNNs on Homogeneous Spaces. *CoRR*, abs/1811.02017.

Cohen, T. S.; Geiger, M.; Köhler, J.; and Welling, M. 2018. Spherical CNNs. *arXiv preprint arXiv:1801.10130*.

Cohen, T. S.; Weiler, M.; Kicanaoglu, B.; and Welling, M. 2019. Gauge Equivariant Convolutional Networks and the Icosahedral CNN. *CoRR*, abs/1902.04615.

Derevyanko, G.; and Lamoureux, G. 2019. Protein-protein docking using learned three-dimensional representations. *bioRxiv*, 738690.

Eismann, S.; Suriana, P.; Jing, B.; Townshend, R. J.; and Dror, R. O. 2020. Protein model quality assessment using rotation-equivariant, hierarchical neural networks. *arXiv preprint arXiv:2011.13557*.

Fuchs, F. B.; Worrall, D. E.; Fischer, V.; and Welling, M. 2020. SE(3)-transformers: 3D roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*.

Gainza, P.; Sverrisson, F.; Monti, F.; Rodola, E.; Boscaini, D.; Bronstein, M.; and Correia, B. 2020. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2): 184–192.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, 1263–1272. PMLR.

Hiranuma, N.; Park, H.; Baek, M.; Anishchenko, I.; Dauparas, J.; and Baker, D. 2021. Improved protein structure refinement guided by deep learning based accuracy estimation. *Nature communications*, 12(1): 1–11.

Hutchinson, M. J.; Lan, C. L.; Zaidi, S.; Dupont, E.; Teh, Y. W.; and Kim, H. 2021. LieTransformer: Equivariant Self-Attention for Lie Groups. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 4533–4543. PMLR.

Igashov, I.; Olechnovič, L.; Kadukova, M.; Venclovas, Č.; and Grudinin, S. 2021. VoroCNN: Deep convolutional neural network built on 3D Voronoi tessellation of protein structures. *Bioinformatics*.

Igashov, I.; Pavlichenko, N.; and Grudinin, S. 2021. Spherical convolutions on molecular graphs for protein model quality assessment. *Machine Learning: Science and Technology*, 2: 045005.

Ingraham, J.; Garg, V. K.; Barzilay, R.; and Jaakkola, T. S. 2019. Generative Models for Graph-Based Protein Design. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net.

Jing, B.; Eismann, S.; Soni, P. N.; and Dror, R. O. 2021a. Equivariant Graph Neural Networks for 3D Macromolecular Structure. *CoRR*, abs/2106.03843.

Jing, B.; Eismann, S.; Suriana, P.; Townshend, R. J. L.; and Dror, R. O. 2021b. Learning from Protein Structure with Geometric Vector Perceptrons. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; and Hassabis, D. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*.

Karasikov, M.; Pagès, G.; and Grudinin, S. 2019. Smooth orientation-dependent scoring function for coarse-grained protein quality assessment. *Bioinformatics*, 35(16): 2801–2808.

Klicpera, J.; Giri, S.; Margraf, J. T.; and Günnemann, S. 2020. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. *arXiv preprint arXiv:2011.14115*.

Klicpera, J.; Groß, J.; and Günnemann, S. 2020. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*.

Kondor, R. 2018. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv preprint arXiv:1803.01588*.

Kondor, R.; Lin, Z.; and Trivedi, S. 2018. Clebsch-gordan nets: a fully fourier space spherical convolutional neural network. *arXiv preprint arXiv:1806.09231*.

Kryshtafovych, A.; Schwede, T.; Topf, M.; Fidelis, K.; and Moult, J. 2019. Critical assessment of methods of protein structure prediction (CASP)-Round XIII. *Proteins*, 87(12): 1011–1020.

Laine, E.; Eismann, S.; Elofsson, A.; and Grudinin, S. 2021. Protein sequence-to-structure learning: Is this the end(-to-end revolution)? *CoRR*, abs/2105.07407.

Manavalan, B.; and Lee, J. 2017. SVMQA: support-vector-machine-based protein single-model quality assessment. *Bioinformatics*, 33(16): 2496–2503.

Mariani, V.; Biasini, M.; Barbato, A.; and Schwede, T. 2013. lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21): 2722–8.

Moult, J.; Pedersen, J. T.; Judson, R.; and Fidelis, K. 1995. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Genetics*, 23(3): ii–v.

Olechnovič, K.; Kulberkytė, E.; and Venclovas, C. 2013. CAD-score: a new contact area difference-based function for evaluation of protein structural models. *Proteins*, 81(1): 149–62.

Olechnovič, K.; and Venclovas, Č. 2017. VoroMQA: Assessment of protein structure quality using interatomic contact areas. *Proteins*, 85(6): 1131–1145.

Pagès, G.; Charmettant, B.; and Grudinin, S. 2019. Protein model quality assessment using 3D oriented convolutional neural networks. *Bioinformatics*, 35(18): 3313–3319.

Poulenard, A.; Rakotosaona, M.-J.; Ponty, Y.; and Ovsjanikov, M. 2019. Effective rotation-invariant point CNN with spherical harmonics kernels. In *2019 International Conference on 3D Vision (3DV)*, 47–56. IEEE.

Romero, D.; Bekkers, E.; Tomczak, J.; and Hoogendoorn, M. 2020. Attentive Group Equivariant Convolutional Networks. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 8188–8199. PMLR.

Romero, D. W.; and Cordonnier, J.-B. 2021. Group Equivariant Stand-Alone Self-Attention For Vision. In *International Conference on Learning Representations*.

Sanyal, S.; Anishchenko, I.; Dagar, A.; Baker, D.; and Talukdar, P. 2020. ProteinGCN: Protein model quality assessment using graph convolutional networks. *BioRxiv*.

Satorras, V. G.; Hoogeboom, E.; Fuchs, F. B.; Posner, I.; and Welling, M. 2021. E (n) Equivariant Normalizing Flows for Molecule Generation in 3D. *arXiv preprint arXiv:2105.09016*.

Satorras, V. G.; Hoogeboom, E.; and Welling, M. 2021. E (n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*.

Schütt, K. T.; Kindermans, P.-J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; and Müller, K.-R. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv preprint arXiv:1706.08566*.

Schütt, K. T.; Unke, O. T.; and Gastegger, M. 2021. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*.

Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A. W.; Bridgland, A.; et al. 2020. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792): 706–710.

Sverrisson, F.; Feydy, J.; Correia, B.; and Bronstein, M. 2020. Fast end-to-end learning on protein surfaces. *bioRxiv*.

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

Townshend, R. J.; Townshend, B.; Eismann, S.; and Dror, R. O. 2020. Geometric Prediction: Moving Beyond Scalars. *arXiv preprint arXiv:2006.14163*.

Uziela, K.; Shu, N.; Wallner, B.; and Elofsson, A. 2016. ProQ3: Improved model quality assessments using Rosetta energy terms. *Sci Rep*, 6: 33509.

Weiler, M.; Geiger, M.; Welling, M.; Boomsma, W.; and Cohen, T. 2018. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. *arXiv preprint arXiv:1807.02547*.

Worrall, D. E.; Garbin, S. J.; Turmukhambetov, D.; and Brostow, G. J. 2017. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5028–5037.

Zemla, A.; Venclovas, Č.; Moult, J.; and Fidelis, K. 1999. Processing and analysis of CASP3 protein structure predictions. *Proteins: Structure, Function, and Bioinformatics*, 37(S3): 22–29.