Machine Learning: Science and Technology

PAPER • OPEN ACCESS

Spherical convolutions on molecular graphs for protein model quality assessment

To cite this article: Ilia Igashov *et al* 2021 *Mach. Learn.: Sci. Technol.* **2** 045005

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 045005 https://doi.org/10.1088/2632-2153/abf856

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
30 December 2020

REVISED
24 March 2021

ACCEPTED FOR PUBLICATION
15 April 2021

PUBLISHED
15 July 2021

PAPER

Spherical convolutions on molecular graphs for protein model quality assessment

Ilia Igashov[1,2,3], **Nikita Pavlichenko**[1,3] **and Sergei Grudinin**[2,*]

[1] Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region 141700, Russia
[2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France
[3] These authors contributed equally.
[*] Author to whom any correspondence should be addressed.

E-mail: sergei.grudinin@inria.fr

Keywords: graph convolutional network, spherical convolution on a graph, spherical harmonics, protein molecular graph, protein structure prediction, deep learning, CASP

Supplementary material for this article is available online

Abstract

Processing information on three-dimensional (3D) objects requires methods stable to rigid-body transformations, in particular rotations, of the input data. In image processing tasks, convolutional neural networks achieve this property using rotation-equivariant operations. However, contrary to images, graphs generally have irregular topology. This makes it challenging to define a rotation-equivariant convolution operation on these structures. In this work, we propose spherical graph convolutional network that processes 3D models of proteins represented as molecular graphs. In a protein molecule, individual amino acids have common topological elements. This allows us to unambiguously associate each amino acid with a local coordinate system and construct rotation-equivariant spherical filters that operate on angular information between graph nodes. Within the framework of the protein model quality assessment problem, we demonstrate that the proposed spherical convolution method significantly improves the quality of model assessment compared to the standard message-passing approach. It is also comparable to state-of-the-art methods, as we demonstrate on critical assessment of structure prediction benchmarks. The proposed technique operates only on geometric features of protein 3D models. This makes it universal and applicable to any other geometric-learning task where the graph structure allows constructing local coordinate systems. The method is available at https://team.inria.fr/nano-d/software/s-gcn/.

1. Introduction

Prediction of protein three-dimensional (3D) structure is an important problem in structural biology and structural bioinformatics. Despite tremendous progress in this field (Greener *et al* 2019, Kryshtafovych *et al* 2019, Xu 2019, Senior *et al* 2020), particularly in light of the recent critical assessment of structure prediction (CASP14) results (Callaway 2020), the accuracy of the predicted structures tends to vary significantly depending on the availability of additional information, and the number of homologous structures and sequences in the databases (Abriata *et al* 2019, Hou *et al* 2019, Senior *et al* 2019, Zheng *et al* 2019). Therefore, estimation of reliability of the predicted models, and also the assessment of the local structural fragments, is crucial for the practical application of these predictions.

The problem of *protein model quality assessment* (MQA) has been recognized by the protein structure modeling community and became one of the subchallenges of CASP, the critical assessment of protein structure prediction community-wide challenge (Cheng *et al* 2019, Won *et al* 2019). All of the state-of-the-art MQA methods use, to a certain extent, supervised or unsupervised machine learning. Initially, statistical potentials (Olechnovič and Venclovas 2017), shallow neural networks (Wallner and Elofsson 2003), regression methods, and support vector machines (Ray *et al* 2012, Uziela *et al* 2016) were widely used. More recently, this problem has also got attention from the machine-learning (ML) community. This triggered the development of more advanced approaches, such as deep learning-based techniques (Derevyanko *et al* 2018, Conover *et al* 2019, Pagès *et al* 2019, Eismann *et al* 2020, Jing *et al* 2020, Hiranuma *et al* 2021) and graph convolutional networks (GCNs) (Baldassarre *et al* 2021, Sanyal *et al* 2020, Igashov *et al* 2021). The latter methods operate on a *molecular graph* representation of protein *models*.

In this work, we propose to capture the 3D structure of a molecular graph using convolution operation based on spherical harmonics. The main idea of our approach is to learn spatial filters in a reference orientation of each graph node. Indeed, proteins are chained molecules, with a repeated topology of the backbone. Thus, using local coordinate frames constructed on the protein's backbone, we can build *rotational-equivariant* spherical filters. We then incorporate these filters into a message-passing framework and design a new method called *spherical graph convolutional network (S-GCN)*, which significantly outperforms the classical GCN architecture.

Most of the protein MQA methods operate on the atom-level representation of a protein molecule (Uziela *et al* 2016, Olechnovič and Venclovas 2017, Karasikov *et al* 2019, Pagès *et al* 2019, Igashov *et al* 2021). At the same time, the state-of-the-art methods use various types of features that often include information about the evolution of molecules or other biological characteristics. On the contrary, S-GCN works with the residue-level protein representation, which significantly reduces computational costs and the number of parameters. Also, our method processes only geometric information, i.e. the input feature vector of each amino acid contains only three geometric features and a one-hot vector representing the type of the amino acid. This work demonstrates that the state-of-the-art quality of the protein model assessment task can be achieved using only geometric properties of protein models without any chemo-physical prior information, which often requires additional expensive computations.

The main results of our work can be summarized as follows:

- We propose a new message-passing method based on trainable rotational-equivariant spherical filters.
- The proposed method significantly improves the quality of model assessment as compared to a classical GCN approach, applied to the same input configuration.
- Despite the residue-level representation and only geometric input features, the results of the proposed method are comparable to the state of the art.

2. Related work

2.1. Structural bioinformatics

Quality assessment of protein models is a classical problem in protein structure prediction community. There have been multiple approaches developed over last 30 years. These include physics-based techniques (Randall and Baldi 2008, Faraggi and Kloczkowski 2014), statistical and unsupervised methods, such as DFIRE (Zhou and Zhou 2002), DOPE (Shen and Sali 2006), GOAP (Zhou and Skolnick 2011), RWplus (Zhang and Zhang 2010), ORDER_AVE (Liu *et al* 2014), VoroMQA (Olechnovič and Venclovas 2014) and more, classical ML-approaches ModelEvaluator (Wang *et al* 2009), ProQ2 (Ray *et al* 2012), Wang_SVM (Liu *et al* 2016), Qprob (Cao and Cheng 2016), SBROD (Karasikov *et al* 2019), a learning-to-rank technique (Jing *et al* 2016), deep learning methods (Derevyanko *et al* 2018, Conover *et al* 2019, Pagès *et al* 2019, Sato and Ishida 2019, Jing and Jinbo 2020, Hiranuma *et al* 2021), neural (Wallner and Elofsson 2003), and graph neural networks (GNNs) (Baldassarre *et al* 2021, Sanyal *et al* 2020, Igashov *et al* 2021).

2.2. GNNs for molecular graphs

In the last years, various GNNs were proposed to address the problem of learning on molecular graphs. Starting with the message-passing paradigm in the molecular graph domain (Gilmer *et al* 2017), further multiple approaches elaborated on this idea (Schütt *et al* 2017, Thomas *et al* 2018, Chen *et al* 2019, Klicpera *et al* 2020, Nachmani and Wolf 2020, Sun *et al* 2020). All of them were designed to operate on small molecules. For example, in QM9 (Blum and Reymond 2009, Rupp *et al* 2012), a popular benchmark that is used to evaluate these methods, molecules consist of up to 23 atoms. On the contrary, a protein molecule can contain thousands of atoms, and this fact requires different approaches that take into account the size of the data. Recently, GNNs have also been started to be applied to protein graphs for solving various problems such as protein design (Ingraham *et al* 2019), protein docking (Fout *et al* 2017, Cao and Shen 2020), classification (Weiler *et al* 2018, Zamora-Resendiz and Crivelli 2019), and quality assessment (Baldassarre *et al* 2021, Sanyal *et al* 2020, Igashov *et al* 2021).

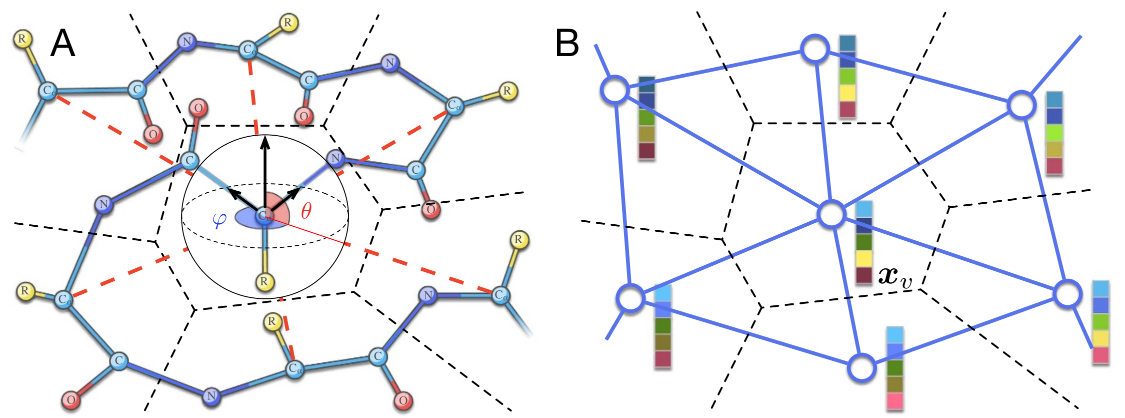

Figure 1. Schematic representation of a molecular graph. (A) 3D protein structure is partitioned into Voronoi cells, shown with the dashed lines. The central amino acid has the associated coordinate system, which is built according to the topology of its backbone (atoms C, C_α, N) with the center at the position of the C_α atom. R symbols denote amino acid residues. The spherical angles φ and θ of the neighboring residues are computed with respect to the local coordinate system of the central residue. (B) Graph corresponding to the Voronoi tessellation, v is the central node, u is its neighbor, $\boldsymbol{x}_v$ and $\boldsymbol{x}_u$ are the corresponding feature vectors, which are also shown with colored boxes.

2.3. Equivariance

Processing information in 3D must be stable against rigid-body transformations of the input data. This stability can be achieved using *equivariant operations*, which is a very active research topic, especially regarding rotational equivariance. For example, rotation-equivariant CNNs were proposed for spherical images using correlations on a sphere (Cohen *et al* 2018) and then extended to fully Fourier-space architectures (Kondor *et al* 2018, Anderson *et al* 2019). Similar architectures can be constructed for rigid-body motions using tensor field rotation- and translation-equivariant networks (Thomas *et al* 2018, Weiler *et al* 2018). Spherical harmonics kernels have also been applied to point-cloud data (Poulenard *et al* 2019). Alternatively, for some types of volumetric data, rotation-equivariant representation can be constructed with oriented local coordinate frames (Pagès *et al* 2019). The same idea can be applied to the protein graph representation, where the spatial relation between local frames can be encoded using spatial edge features (Ingraham *et al* 2019) or additional edge descriptors (Sanyal *et al* 2020). For general molecular graphs, the problem is more difficult. Still, there has been significant progress using, e.g. the message-passing formalism with messages containing radial and directional information about neighboring graph nodes (Klicpera *et al* 2020).

3. Proposed method

3.1. Protein graph

A protein molecule is a chain of amino acids, or *residues*, folded in a 3D space. We construct a graph $\mathcal{G}$ of the protein molecule by splitting the surrounding space into cells using the Voronoi tessellation method Voronota (Olechnovič and Venclovas 2014). Nodes of the resulting graph correspond to the protein residues and edges are associated with the pairs of residues whose Voronoi cells have a non-zero contact surface. Figure 1 schematically shows the graph construction.

Each node v of the graph $\mathcal{G}$ contains a feature vector $\boldsymbol{x}_v$ associated with the corresponding protein residue. These features include one of 20 amino-acid types encoded with the one-hot representation, the solvent-accessible surface area for each residue, the volume of residue's Voronoi cell, and the 'buriedness' of the residue, which is a topological distance in the graph $\mathcal{G}$ to the nearest solvent-accessible node. We represent the whole set of nodes as a feature matrix $\mathbf{X} \in \mathbb{R}^{N\times d}$ where N is a number of residues and $d = 23$ is the size of the feature vector.

To describe the edges of the graph $\mathcal{G}$, we will use the following notations. Let $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ be the symmetric binary adjacency matrix of the graph. For any pair of residues v and u, the two corresponding entries of the matrix $\boldsymbol{A}$ equal 1 if v and u have an edge, and zero otherwise. In our settings, the graph $\mathcal{G}$ does not have *self-loops*, hence the main diagonal elements of the matrix $\boldsymbol{A}$ are zeros. In order to refer to neighbors of a node v in the graph $\mathcal{G}$, i.e. those nodes that have a common edge with v, we will use notation $\mathcal{N}(v)$.

3.2. Spherical harmonics

Let us consider a complex square-integrable function $f(\theta,\varphi)$ defined on a unit sphere S_1. This function can be expanded in a polynomial basis using *spherical harmonics* as the basis functions,

$$f(\theta,\varphi)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l} w_l^m Y_l^m(\theta,\varphi), \tag{1}$$

where w_l^m are the expansion coefficients, and $Y_l^m(\theta,\varphi)$ are the spherical harmonics (Hobson 1955),

$$Y_l^m(\theta,\varphi)=\sqrt{\frac{(2l+1)}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos\theta)e^{im\varphi}. \tag{2}$$

Here, $P_l^m(\cos\theta)$ are the associated Legendre polynomials (Hobson 1955). We should also note that a real function on a unit sphere can be decomposed in a polynomial basis more compactly using *real spherical harmonics* as the basis functions. Below we will be using this real basis, which is specified in supplementary materials (is available online at stacks.iop.org/MLST/2/045005/mmedia).

3.3. Local coordinate system

The protein backbone consists of atom repetitions C, C_α, N, O. This allows us to unambiguously associate each residue with a local coordinate system. Indeed, for each residue we can define the normalized C_α–N vector as the x-axis, the unit vector lying in the C–C_α–N plane, orthogonal to x, and having positive dot product with C_α–C as the y-axis, and the vector product of x with y as the z-axis. Then, given a node v, we can associate each neighbor $u\in\mathcal{N}(v)$ with a pair of *spherical angles* $\Omega_v^u=(\theta_v^u,\varphi_v^u)$. They specify the angular position of the projection of the node u onto a unit sphere in the local coordinate system of v. An example of a local coordinates system is schematically shown in figure 1(A). Now, having an unambiguous orientation for each node in the graph, we can construct a *rotation-equivariant* convolution operation.

3.4. Spherical convolution

We can approximate the expansion (3) of the function $f(\theta,\varphi)$ by cutting the series at the maximum expansion order L,

$$f(\theta,\varphi)\approx\hat{f}(\theta,\varphi)=\sum_{l=0}^{L}\sum_{m=-l}^{l} w_l^m Y_l^m(\theta,\varphi). \tag{3}$$

The same approximation can be obtained for a matrix function $\boldsymbol{F}: S_1\to\mathbb{R}^{d_1\times d_2}$, $d_1,d_2\in\mathbb{N}$,

$$\boldsymbol{F}(\theta,\varphi)\approx\hat{\boldsymbol{F}}(\theta,\varphi)=\sum_{l=0}^{L}\sum_{m=-l}^{l}\boldsymbol{W}_l^m Y_l^m(\theta,\varphi), \tag{4}$$

where matrices $\boldsymbol{W}_l^m$ denote expansion coefficients of the function $\boldsymbol{F}$ in the Y_l^m basis. Finally, we can introduce the spherical convolution operation for the vertex v in the following way,

$$\boldsymbol{F}\circ v=\sum_{u\in\mathcal{N}(v)}\hat{\boldsymbol{F}}(\theta_v^u,\varphi_v^u)\boldsymbol{x}_v. \tag{5}$$

Considering matrices $\boldsymbol{W}_l^m$ to be optimized parameters, we will thus learn a spherical filter. We should specifically emphasize that matrices $\boldsymbol{W}_l^m$ are rotation-equivariant by construction.

3.5. Neural network

The distinctive feature of convolutional networks built on spatial graphs is the way the graph nodes exchange information by passing messages to each other. On each layer of the network, nodes' feature vectors are combined and updated using the information from the neighboring nodes (Scarselli *et al* 2009, Kipf and Welling 2017). In our implementation, for the information exchange, we use the proposed spherical convolution operation (5).

Let $\boldsymbol{A}_\Omega\in\mathbb{R}^{N\times N}$ be a matrix of local angular coordinates for each node's neighbor in the adjacency matrix $\boldsymbol{A}$. This means, for any pair of graph nodes v and u connected with an edge, the corresponding entry of matrix $\boldsymbol{A}_\Omega$ is a pair $\Omega_v^u=(\theta_v^u,\varphi_v^u)$ of angular coordinates of u with respect to the local coordinate system of v. We also denote $Y_l^m(\boldsymbol{A}_\Omega)\in\mathbb{R}^{N\times N}$ as a result of the elementwise application of the spherical harmonics Y_l^m to the matrix $\boldsymbol{A}_\Omega$. We should note that the main diagonal elements of matrices $\boldsymbol{A}_\Omega$ and $Y_l^m(\boldsymbol{A}_\Omega)$ are zeros, and, opposed to the adjacency matrix $\boldsymbol{A}$, matrices $\boldsymbol{A}_\Omega$ and $Y_l^m(\boldsymbol{A}_\Omega)$ are not symmetric. Then, the kth layer of the S-GCN can be expressed as follows,

$$\boldsymbol{H}^k = \sigma\left(\sum_{l,m}^{L} Y_l^m(\boldsymbol{A}_\Omega)\boldsymbol{H}^{k-1}\boldsymbol{W}_l^m + \boldsymbol{H}^{k-1}\boldsymbol{W} + \boldsymbol{b}\right), \tag{6}$$

where $\boldsymbol{H}^{k-1} \in \mathbb{R}^{N\times d_{k-1}}$ and $\boldsymbol{H}^k \in \mathbb{R}^{N\times d_k}$ are nodes' feature matrices before and after applying the layer, $\boldsymbol{H}^0 \equiv \boldsymbol{X}$ is the input feature matrix, $\boldsymbol{W}_l^m \in \mathbb{R}^{d_{k-1}\times d_k}$ and $\boldsymbol{W} \in \mathbb{R}^{d_{k-1}\times d_k}$ are trainable parameters, $\boldsymbol{b} \in \mathbb{R}^{d_k}$ is a trainable bias vector, and σ is a nonlinear activation function. If we let the maximum expansion order $L = 0$, we can see that the operation (6) reduces to a standard message-passing form,

$$\boldsymbol{H}^k = \sigma\left(\boldsymbol{A}\boldsymbol{H}^{k-1}\boldsymbol{W}_0^0 + \boldsymbol{H}^{k-1}\boldsymbol{W} + \boldsymbol{b}\right), \tag{7}$$

where for each node v, the first term transforms features from v's neighbors $\mathcal{N}(v)$, the second term transforms and aggregates features from v itself, and the last terms provides a bias for the activation function σ.

In supplementary materials, we also provide a modification of the proposed spherical convolution layer (6), which explicitly uses information about contact surface areas between Voronoi cells, and discuss the corresponding network architecture.

4. Experiment

The main purpose of the MQA task is to evaluate the deviation between a generated model of a protein molecule and its native, or *target*, structure. If the target structure is known, the quality of the model can be calculated by computing one of specifically designed metrics, e.g. CAD-score (Olechnovič *et al* 2012), lDDT (Mariani *et al* 2013), or GDT-TS (Zemla *et al* 1999). Most often, however, experimental protein structures are unknown, and thus there is a need for protein structure prediction and MQA. In this section, we report the results of the protein MQA task obtained by our spherical architectures and the GCN baseline. We trained all of our networks using *local* per-residue CAD-scores as the ground truth. They have been shown to be a more informative and stable metric compared to other MQA measures with respect to local structural perturbations of a protein molecule (Olechnovič *et al* 2019). To retrieve the *global* per-model scores, we averaged the predicted local scores.

We provide results of S-GCN along with several state-of-the-art MQA methods that are described in detail below. We would also like to emphasize that for several reasons, in this work, we do not compare S-GCN with recent GNNs designed for small molecules. First of all, we attempted to train tensor field networks (Thomas *et al* 2018) and DimeNet (Klicpera *et al* 2020) on protein molecules but did not get any adequate results. Secondly, we are unable to test S-GCN on established ML benchmarks such as QM9 (Blum and Reymond 2009, Rupp *et al* 2012) due to the specificity of the graph representation in our method.

4.1. Datasets

For our experiments, we collected data from the CASP benchmarks (Moult *et al* 1995, Kryshtafovych *et al* 2019). They contain experimentally obtained native protein structures and the corresponding 3D models predicted by the CASP challenge participants.

For training, we used data from CASP[8–11] stage2 submissions. For each target, we additionally generated 50 near-native models (Hoffmann and Grudinin 2017) in order to enrich the training dataset with high-quality examples. For validation and selection of hyperparameters, we used data from CASP12 stage2 submissions. All models that we used for training and validation were initially filtered and preprocessed. More precisely, we excluded targets that had only models with low CAD-scores and preprocessed all models by removing residues that were not present in the target structure. More details and the list of all targets are available in supplementary materials. In total, we had 333 target structures and 73 418 models from CASP[8–11] for training and 39 targets and 5411 models from CASP12 for validation. Finally, to test our architectures, we used unrefined data from CASP13 (73 target structures, 10 882 models) and unrefined data from CASP12 (38 target structures, 5471 models). The overall homology between the targets is very low, as it is shown in supplementary materials.

For each model, we precomputed matrices $Y_l^m(A_\Omega)$ up to the 10th expansion order. These matrices were the most space-consuming part of our dataset, as a spherical harmonic expansion of order L requires the storage of L^2 coefficients for each pair of adjacent nodes in a graph.

4.2. Metrics

For the evaluation of the methods, we chose z-scores, mean squared error (MSE), determination coefficient R^2, Pearson, and Spearman correlations, as it is described in more detail below. We used global CAD-scores

Table 1. Architectures of our baseline and S-GCNs. SCL and GCL are the spherical and graph convolution layers, correspondingly. FC is a fully-connected layer with ELU activation. The parameters in the parentheses are the sizes of the input and the output feature vectors, correspondingly. BN is the batch normalization layer.

Network	Architecture
Baseline	***Encoder:*** FC(23, 32) → Dropout → FC(32, 64) → Dropout → FC(64, 128) → Dropout → ***Message-passing:*** GCL(128, 113) → Dropout → GCL(113, 98) → Dropout → GCL(98, 83) → Dropout → GCL(83, 68) → Dropout → GCL(68, 53) → Dropout → GCL(53, 38) → Dropout → GCL(38, 23) → Dropout →GCL(23, 8) → Dropout ***Scorer:*** FC(8, 16) → Dropout → FC(16, 32) → Dropout → FC(32, 64) → Dropout → FC(64, 32) → Dropout → FC(32, 16) → Dropout → FC(16, 1) → Sigmoid
S-GCN	SCL(23, 20) → Dropout → SCL(20, 16) → BN → Dropout → SCL(16, 8) → Dropout → SCL(8, 4) → BN → Dropout → SCL(4, 1) → Sigmoid
S-GCN$_s$	***Spherical part:*** SCL(23, 20) → Dropout → SCL(20, 16) → BN → Dropout → SCL(16, 14) → Dropout → SCL(14, 12) → BN → Dropout → SCL(12, 8)→ ***Scorer:*** FC(8, 128) → D → FC(128, 64) → D → FC(64, 1) → Sigmoid

as the ground truth for the assessment[5]. We computed z-scores for the top-predicted protein models for each target and then averaged them over all targets, as explained in more detail in supplementary materials. For the MSE, R^2, and correlations, we used two different ways of calculation: *per-target* and *global*. In the per-target approach, we computed the metrics separately within each protein target and then averaged results over all targets (we averaged correlations using the Fisher transformation (Fisher 1915)). In the global approach, we stacked scores of all protein models into one vector and calculated the metrics on this vector. For each metric, we also computed bootstrapped means and confidence intervals. For the global metrics, a bootstrapped sample is chosen from the whole set of models. For the per-target metrics, a bootstrapped sample is a sample of targets and their models, respectively. These results are available in supplementary materials.

4.3. Baseline architecture

For the baseline, we built a standard GNN based on the message-passing operation described in equation (7). The structure of the proposed architecture can be split into three main parts. The *encoder* is a set of fully-connected layers that transform the residues' features into a high-dimensional space. The *message-passing part* is a set of graph convolution layers (7) that capture the structure of a protein graph and work as a feature extractor. Finally, the *scorer* is a set of fully-connected layers with a sigmoid at the end. They form a multilayer perceptron and use the obtained features to predict the scores of each protein residue. As a result, we obtain three main design parameters—the number of encoder, message-passing, and scoring layers. We performed a grid search on the values of these parameters and found out that the optimal architecture had three encoder layers, eight message passing layers, and three scoring layers. For each layer, we used the ELU activation function and the dropout rate set to 0.3, as we detected it to be optimal. In total, our baseline network contains 339 053 trainable parameters. Table 1 briefly lists the final architecture.

4.3.1. Training

We trained this network on CASP[8–11] datasets for 40 iterations. We tuned hyperparameters on CASP12 (preprocessed) dataset. For training, we used the Adam optimizer (Kingma and Ba 2015) and the MSE of local scores as the loss function. On each iteration, we trained the network in four parallel processes feeding 512 models to each process. One training iteration took ≈10 min on Intel® Xeon(R) W-2123 CPU @ 3.60 GHz and ≈1 min on NVIDIA Quadro P5000/PCIe/SSE2 GPU.

4.3.2. Hyperparameters

The learning rate was 0.001, the batch size 1, the dropout rate 0.3, and the L_2-regularization coefficient of 10^{-5}.

4.4. S-GCN architectures

While constructing a S-GCN, we considered multiple expansion orders in the range from 3 to 10 and finally chose orders of 5 and 10. We also experimented with the number of layers, the batch normalization layers (Ioffe and Szegedy 2015) and the batch size, dropouts, the regularization parameters, and the output

[5] Although we mainly focus on experiments with CAD-score as the ground truth, we also evaluated the quality of our method on the same data with lDDT and GDT-TS as the ground truth. The results are available in supplementary materials.

dimensionality of the spherical convolution layers. The details of these experiments are available in supplementary materials.

Finally, we settled upon two architectures. The first architecture, S-GCN, represents a sequence of spherical convolution layers (6) combined with dropout and batch normalization layers. In the second architecture, S-GCN$_s$, we added three fully-connected layers to the end of the network following the idea used in the baseline. S-GCN with the expansion order of 5 contains 24 675 trainable parameters, and S-GCN with the order of 10 contains 95 475 trainable parameters. Respectively, S-GCN$_s$ with the order of 5 contains 42 625 trainable parameters, and S-GCN$_s$ with the order of 10 contains 137 725 trainable parameters. Table 1 briefly describes these configurations.

4.4.1. Training

For training, we used the Adam optimizer and the MSE of local scores as the loss function. We trained the networks on the shuffled data and split the whole training process into equal iterations. Within each iteration, we trained each network in four parallel processes feeding 2048 models to each of them. We stored and processed the adjacency matrices in a sparse format. One training iteration takes on average ≈18 min for S-GCN(5) and ≈26 min for S-GCN(10) on Intel® Xeon(R) W-2123 CPU @ 3.60 GHz, and ≈9 min for S-GCN(5) and ≈34 min for S-GCN(10) on NVIDIA Quadro P5000/PCIe/SSE2 GPU. Such a significant difference between the CPU and GPU timings compared to the baseline experiment is explained by loading of precomputed spherical harmonics from the hard drive. We trained 5th-order networks for 40 iterations and 10th-order networks for 60 iterations.

4.4.2. Hyperparameters

The learning rate was set to 0.001, the batch size 64, the dropout rate 0.2 and we used L_2-regularization with the coefficient of 0.003 for the network of order 5. We used the dropout rate of 0.1 and L_2-regularization with the coefficient of 0.001 for the network of order 10.

4.5. Results

We compared S-GCN and S-GCN$_s$ with our baseline network architecture, and also with the state-of-the-art single-model (Cheng *et al* 2019) quality assessment methods SBROD (Karasikov *et al* 2019), VoroMQA (Olechnovič and Venclovas 2017), ProQ3 (Uziela *et al* 2016), Ornate (Pagès *et al* 2019), and VoroCNN (Igashov *et al* 2021). SBROD is a regression-based method operating on 4D geometric descriptors, VoroMQA uses statistics from Voronoi 3D tessellation, ProQ3 is a neural-network-based method with precomputed descriptors of various origin, Ornate uses deep convolutional networks to process volumetric data in local coordinate frames, and, finally, VoroCNN is a GCN built on an atom-level molecular graph. We downloaded the results of VoroMQA and ProQ3 for CASP[12–13] and the results of SBROD for CASP13 from the official CASP archive at predictioncenter.org. To obtain the results of SBROD on CASP12 and Ornate and VoroCNN on CASP[12–13], we ran these methods locally. We should emphasize that the main results we report in this work were obtained on the CASP13 dataset, which was not used during training and validation. However, to give a complete picture, we also provide the results obtained on the unrefined CASP12 dataset. Table 2 lists the results for CASP12 and table 3 lists the results for CASP13.

First of all, we can see a huge performance gap between the baseline network and the other methods. This can be explained by the fact that the baseline approach uses neither the 3D structure of the graph nor additional chemo-physical or biological features that are widely accepted by the state-of-the-art methods. At the same time, we would like to emphasize that our S-GCNs, which explicitly use the 3D structure of the data, managed to achieve a similar or better quality of predictions compared to the state-of-the-art methods. Figure 2(A) shows some of the spherical filters learned by the 5th and the 10th order S-GCNs. We can see their rather complex shape, which is difficult to interpret solely from physico-chemical considerations.

Tables 2 and 3 also demonstrate that using a higher order of the spherical harmonic expansion improves the MSE and R^2 metrics. At the same time, the order 5 seems to outperform the 10th order in correlation and z-score metrics. This behavior becomes clearer if we look closer at the absolute values of the predictions. Indeed, figure 2(B) illustrates that the predictions of the 10th order network are closer to the diagonal, thus improving MSE and R^2. It also explains that even though some methods can have high correlation metrics, their predictions are shifted with respect to the main diagonal, which results in negative R^2. Thus, we can conclude that a higher polynomial order of the network allows us to better predict the absolute values of protein scores.

Regarding the correlation metrics, the 5th order S-GCN performs better than the others. We can conclude that it should be the method of choice for ranking protein models and selecting the best model from a given set. Also, taking into account the fact that the 10th order S-GCN has considerably more

Table 2. Comparison of S-GCN and S-GCN$_s$ with the baseline network and the state-of-the-art MQA methods on the unrefined CASP12 stage2 dataset. All metrics are provided for CAD-score. Parameters in parentheses correspond to the order of the spherical harmonic expansion.

		Global metrics				Per-target metrics			
Method	z-score	MSE	R^2	Pearson, r	Spearman, ρ	MSE	R^2	Pearson, r	Spearman, ρ
SBROD	1.282	0.961	−81.899	0.552	0.531	0.961	−427.838	0.762	0.685
VoroMQA	1.410	0.051	−3.426	0.675	0.700	0.051	−19.762	0.803	0.766
ProQ3	1.670	0.035	−2.036	0.795	0.806	0.035	−16.572	0.801	0.750
Ornate	1.780	0.007	0.424	0.813	0.805	0.007	−1.101	**0.828**	**0.781**
VoroCNN	**1.871**	0.007	0.370	0.818	0.803	0.007	−1.380	0.817	0.774
Baseline	1.025	0.011	0.065	0.658	0.666	0.011	−2.641	0.677	0.604
S-GCN(5)	1.704	0.010	0.157	0.854	0.831	0.010	−1.890	0.797	0.738
S-GCN(10)	1.665	**0.005**	**0.573**	0.812	0.789	**0.005**	**−0.831**	0.710	0.680
S-GCN$_s$(5)	1.609	0.015	−0.272	**0.872**	**0.853**	0.015	−3.866	0.816	0.762
S-GCN$_s$(10)	1.303	0.006	0.492	0.803	0.790	0.006	−0.917	0.738	0.683

Note: Best column values are highlighted in bold.

Table 3. Comparison of S-GCN and S-GCN$_s$ with the baseline network and the state-of-the-art MQA methods on the unrefined CASP13 stage2 dataset. All metrics are provided for CAD-score. Parameters in parentheses correspond to the order of the spherical harmonic expansion.

		Global metrics				Per-target metrics			
Method	z-score	MSE	R^2	Pearson, r	Spearman, ρ	MSE	R^2	Pearson, r	Spearman, ρ
SBROD	1.453	0.050	−3.234	0.417	0.433	0.051	−22.455	0.805	0.761
VoroMQA	1.369	0.038	−2.197	0.659	0.688	0.038	−15.930	0.804	0.768
ProQ3	1.459	0.035	−1.969	0.726	0.728	0.035	−17.519	0.775	0.737
Ornate	1.403	0.009	0.193	0.786	0.799	0.009	−2.326	0.814	**0.786**
VoroCNN	1.516	**0.007**	0.368	0.764	0.767	**0.007**	−1.962	0.811	0.771
Baseline	0.865	0.017	−0.424	0.465	0.491	0.017	−6.375	0.648	0.619
S-GCN(5)	1.362	0.013	−0.118	**0.806**	**0.808**	0.013	−3.459	0.789	0.744
S-GCN(10)	1.247	**0.007**	**0.422**	0.774	0.783	**0.007**	**−1.348**	0.722	0.694
S-GCN$_s$(5)	**1.582**	0.020	−0.668	0.801	0.799	0.020	−6.415	**0.820**	0.773
S-GCN$_s$(10)	1.281	0.008	0.336	0.779	0.785	0.008	−1.760	0.742	0.702

Note: Best column values are highlighted in bold.

trainable parameters, and takes four times more disk space than the 5th order S-GCN, it makes more sense to use the latter for practical tasks. We can also see that the last scoring layer improves the correlation metrics. S-GCNs also demonstrate a better prediction quality on z-scores, but, as we show in supplementary materials, these metrics are not stable and we can not confidently say that one method outperforms another because they all have intersecting confident intervals.

One final remark that we can make after comparing the 5th and the 10th order S-GCNs is that the 10th order architecture may require significantly more training data. Therefore, the current CASP training set may not be very well suited for higher-order architectures. As an alternative, one can consider training on Rosetta-generated decoys (Hiranuma *et al* 2021) or using other methods for protein structure prediction.

Let us now demonstrate some practical examples of local per-residue score estimations. Figures 3(A) and (B) shows S-GCN(5) predictions for an obligatory complex of bacteriophage RNA-binding protein and its individual subunits. We can clearly see that the binding interface has lower scores compared to the rest of the structure, and is very visually distinguishable. This can be explained by the specificity of the interface, as it lacks energetically favorable partner contacts in the unbound state. Another example of a structural transition between the open and the closed states of a D-ribose-binding protein is shown in figures 3(D)–(F). As in the previous case, we can see that the interface between the two protein domains becomes more energetically favorable in the closed form of the protein.

An interesting general question is how well S-GCN can distinguish target (native) structures from the generated models. Supplementary figure S5 shows distributions of the global scores predicted for targets and models from CASP13 by S-GCN(5) and S-GCN(10). In both cases, we can see a clear separation between the two distributions.

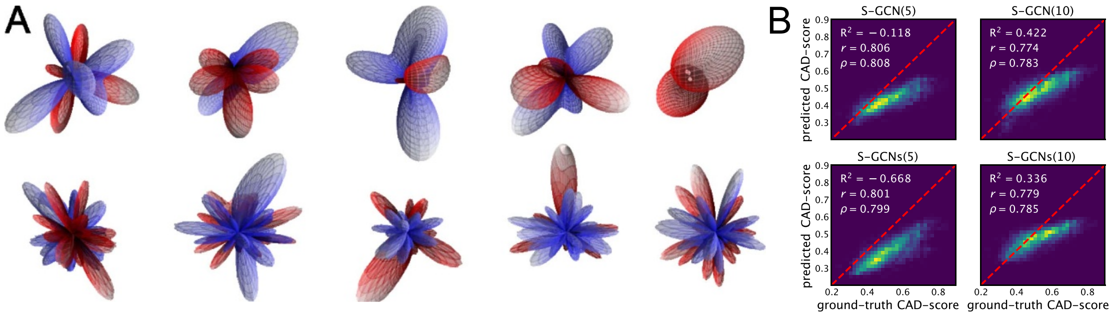

Figure 2. (A) Examples of spherical filters learned by S-GCN of order 5 (top row) and S-GCN of order 10 (bottom row). The distance to the center is proportional to the absolute function value. The red color corresponds to the positive values, the blue color—to the negative ones. (B) Histograms comparing the ground-truth scores and the predictions of S-GCNs on the CASP13 dataset.

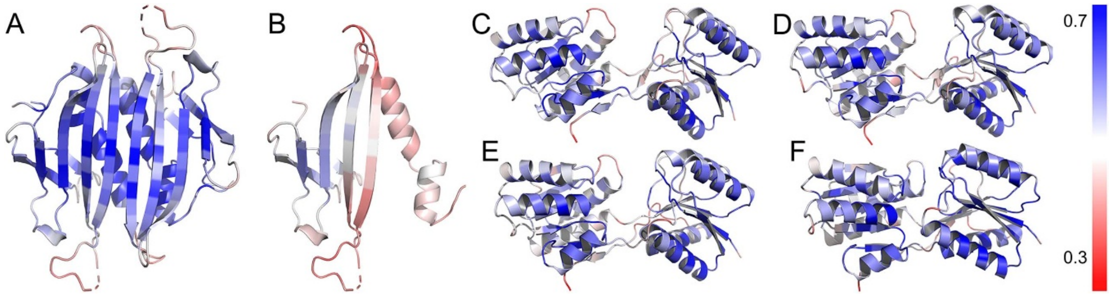

Figure 3. (A)–(B) S-GCN(5) predictions for the obligatory complex of bacteriophage RNA-binding protein (pdb code 1una, (A) and its individual subunit (B). (C)–(F) S-GCN(5) predictions for the open-to-close transition of the D-ribose-binding protein, the open state (pdb code 1ba2_A, (C), two intermediates (pdb codes 1ba2_B, D, and 1urp, (E), and the closed state (pdb code 2dri, (F). The colorbar on the right shows the distribution of the local scores, from 0.3 (red) to 0.7 (blue).

5. Conclusion

In this work, we applied spherical convolutions to capture the 3D structure of a protein graph. The results demonstrate that our method gives a significant improvement in the quality of predictions compared to the baseline without orientational relations between the graph nodes. The spherical convolution method can also be combined with other approaches for the protein MQA, and can also potentially use more input features. Thus, we believe it will be possible to achieve even higher prediction results adding biological and chemical information to the input graphs. In addition, we would like to notice that the idea of spherical convolutions is universal and can be applied to various types of graph-learning tasks, provided that the graph structure permits us to define an equivariant coordinate systems for each graph node.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://team.inria.fr/nano-d/software/s-gcn/.

Acknowledgments

We would like to thank Kliment Olechnovič from Vilnius University for his help on graph construction and active discussions during the project. We would also like to thank Elodie Laine from Sorbonne Université for the discussions during the study and proof-reading the manuscript, Jinbo Xu from Toyota Technological Institute at Chicago for his helpful comments on the manuscript, Maria Kadukova from Moscow Institute of Physics and Technology for her help with pairwise sequence comparison, and Vadim Strijov from Moscow Institute of Physics and Technology for his course "My first scientific paper". This work was partially supported by the Inria International Partnership program BIOTOOLS.

Broader impact

Our work will likely stimulate the development of new representation-learning methods applied to 3D graphs. The latter may represent molecules, such as proteins or nucleic acids. However, these graphs can also describe more general 3D data from other research domains, e.g. from astronomy or earth science.

From the application point of view, we believe our method will be useful in the structural bioinformatics and structural biology communities. Indeed, there has been a very rapid improvement of methods for 3D protein structure prediction, mostly owing to novel developments in deep learning and in algorithms extracting coevolution signals from sequence data. However, the question of how to assess the quality of the predicted models, and which parts of the predicted structures are likely to be less accurate, is still open. Thus, we hope that the proposed approach will help bioinformaticians and structural biologists to better use and analyze available computational data.

From a more general perspective, proteins are responsible for the main cellular functions in any organism. They maintain the shape of the cells, control chemical catalysis, play the role of cellular motors, and regulate vital processes. This makes the study of protein structures and interactions an important part of molecular biology. Understanding protein structure is also crucial for therapeutic purposes toward the development of new drugs, and the improvement of existing ones. To conclude, gaining knowledge of proteins and their functions contribute to a better understanding of the life machinery and organization, which has a significant social impact.

ORCID iDs

Ilia Igashov https://orcid.org/0000-0002-6214-2827
Nikita Pavlichenko https://orcid.org/0000-0002-7330-393X
Sergei Grudinin https://orcid.org/0000-0002-1903-7220

References

Abriata L A, Tamò G E and Peraro M D 2019 A further leap of improvement in tertiary structure prediction in CASP13 prompts new routes for future assessments *Proteins: Struct. Funct. Bioinform.* **87** 1100–12

Anderson B, Hy T S and Kondor R 2019 Cormorant: covariant molecular neural networks *Advances in Neural Information Processing Systems* pp 14510–9

Baldassarre F, Hurtado D Mendez, Elofsson A and Azizpour H 2021 GraphQA: Protein model quality assessment using graph convolutional network *Bioinformatics* 37 360–6

Blum L C and Reymond J-L 2009 970 million drug like small molecules for virtual screening in the chemical universe database GDB-13 *J. Am. Chem. Soc.* **131** 8732

Callaway E 2020 It will change everything': DeepMind's AI makes gigantic leap in solving protein structures *Nature* **588** 203–4

Cao R and Cheng J 2016 Protein single-model quality assessment by feature-based probability density functions *Sci. Rep.* **6** 23990

Cao Y and Shen Y 2020 Energy-based graph convolutional networks for scoring protein docking models *Proteins: Struct. Funct. Bioinform.* accepted (https://doi.org/10.1002/prot.25888)

Chen C, Weike Y, Zuo Y, Zheng C and Ong S P 2019 Graph networks as a universal machine learning framework for molecules and crystals *Chem. Mater.* **31** 3564–72

Cheng J *et al* 2019 Estimation of model accuracy in CASP13 *Proteins: Struct. Funct. Bioinform.* **87** 1361–77

Cohen T S, Geiger M, Köhler J and Welling M 2018 Spherical CNNs *Sixth Int. Conf. on Learning Representations*

Conover M, Staples M, Dong S, Sun M and Cao R 2019 AngularQA: protein model quality assessment with LSTM networks *Comput. Math. Biophys.* **7** 1–9

Derevyanko G, Grudinin S, Bengio Y and Lamoureux G 2018 Deep convolutional networks for quality assessment of protein folds *Bioinformatics* **34** 4046–53

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

Faraggi E and Kloczkowski A 2014 A global machine learning based scoring function for protein structure prediction *Proteins: Struct. Funct. Bioinform.* **82** 752–9

Fisher R A 1915 Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population *Biometrika* **10** 507–21

Fout A, Byrd J, Shariat B and Ben-Hur A 2017 Protein interface prediction using graph convolutional networks *Advances in Neural Information Processing Systems* pp 6530–9

Gilmer J, Schoenholz S S, Riley P F, Vinyals O and Dahl G E 2017 Neural message passing for quantum chemistry *Proc. 34th Int. Conf. on Machine Learning-Volume 70* (JMLR.org) pp 1263–72

Greener J G, Kandathil S M and Jones D T 2019 Deep learning extends de novo protein modelling coverage of genomes using iteratively predicted structural constraints *Nat. Commun.* **10** 3977

Hiranuma N, Park H, Anishchanka I, Baek M, Dauparas J and Baker D 2021 Improved protein structure refinement guided by deep learning based accuracy estimation *Nat. Commun.* **12** 1340

Hobson E W 1955 *The Theory of Spherical and Ellipsoidal Harmonics* (New York: Chelsea Pub. Co.)

Hoffmann A and Grudinin S 2017 NOLB: nonlinear rigid block normal-mode analysis method *J. Chem. Theory Computat.* **13** 2123–34

Hou J, Tianqi W, Cao R and Cheng J 2019 Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13 *Proteins: Struct. Funct. Bioinform.* **87** 1165–78

Igashov I, Olechnovic K, Kadukova M, Venclovas C and Grudinin S 2021 VoroCNN: deep convolutional neural network built on 3D Voronoi tessellation of protein structures *Bioinformatics* accepted (https://doi.org/10.1093/bioinformatics/btab118)

Ingraham J, Garg V, Barzilay R and Jaakkola T 2019 Generative models for graph-based protein design *Advances in Neural Information Processing Systems* pp 15794–805

Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *32nd Int. Conf. on Machine Learning*

Jinbo X 2019 Distance-based protein folding powered by deep learning *Proc. Natl Acad. Sci.* **116** 16856–65

Jing B, Eismann S, Suriana P, Townshend R J L and Dror R 2020 Learning from protein structure with geometric vector perceptrons (arXiv:2009.01411)

Jing X and Jinbo X 2020 Improved protein model quality assessment by integrating sequential and pairwise features using deep learning *Bioinformatics* 36 5361–7

Jing X, Wang K, Ruqian L and Dong Q 2016 Sorting protein decoys by machine-learning-to-rank *Sci. Rep.* **6** 31571

Karasikov M, Pagès G and Grudinin S 2019 Smooth orientation-dependent scoring function for coarse-grained protein quality assessment *Bioinformatics* **35** 2801–8

Kingma D P and Jimmy B 2015 Adam: a method for stochastic optimization *Third Int. Conf. for Learning Representations*

Kipf T N and Welling M 2017 Semi-supervised classification with graph convolutional networks *Fifth Int. Conf. on Learning Representations*

Klicpera J, Groß J and Günnemann S 2020 Directional message passing for molecular graphs *Eighth Int. Conf. on Learning Representations*

Kondor R, Lin Z and Trivedi S 2018 Clebsch–Gordan nets: a fully Fourier space spherical convolutional neural network *Advances in Neural Information Processing Systems* pp 10117–26

Kryshtafovych A, Schwede T, Topf M, Fidelis K and Moult J 2019 Critical assessment of methods of protein structure prediction (CASP)—round XIII *Proteins: Struct. Funct. Bioinform.* **87** 1011–20

Liu T, Wang Y, Eickholt J and Wang Z 2016 Benchmarking deep networks for predicting residue-specific quality of individual protein models in CASP11 *Sci. Rep.* **6** 19301

Liu Y, Zeng J and Gong H 2014 Improving the orientation-dependent statistical potential using a reference state *Proteins* **82** 2383–93

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** 2722–8

Moult J, Pedersen J T, Judson R and Fidelis K 1995 A large-scale experiment to assess protein structure prediction methods *Proteins: Struct. Funct. Bioinform.* **23** ii–iv

Nachmani E and Wolf L 2020 Molecule property prediction and classification with graph hypernetworks (arXiv:2002.00240)

Olechnovič K, Kulberkytė E and Venclovas Česlovas 2012 CAD-score: a new contact area difference-based function for evaluation of protein structural models *Proteins: Struct. Funct. Bioinform.* **81** 149–62

Olechnovič K, Monastyrskyy B, Kryshtafovych A and Venclovas Česlovas 2019 Comparative analysis of methods for evaluation of protein models against native structures *Bioinformatics* **35** 937–44

Olechnovič K and Venclovas Česlovas 2014 Voronota: a fast and reliable tool for computing the vertices of the Voronoi diagram of atomic balls *J. Comput. Chem.* **35** 672–81

Olechnovič K and Venclovas Česlovas 2017 VoroMQA: assessment of protein structure quality using interatomic contact areas *Proteins: Struct. Funct. Bioinform.* **85** 1131–45

Pagès G, Charmettant B and Grudinin S 2019 Protein model quality assessment using 3D oriented convolutional neural networks *Bioinformatics* **35** 3313–19

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

Randall A and Baldi P 2008 SELECTpro: effective protein model selection using a structure-based energy function resistant to BLUNDERs *BMC Struct. Biol.* **8** 52

Ray A, Lindahl E and Wallner Born 2012 Improved model quality assessment using ProQ2 *BMC Bioinform.* **13** 224

Rupp M, Tkatchenko A, Müller K-R and von Lilienfeld O A 2012 Fast and accurate modeling of molecular atomization energies with machine learning *Phys. Rev. Lett.* **108** 058301

Sanyal S, Anishchenko I, Dagar A, Baker D and Talukdar P 2020 ProteinGCN: protein model quality assessment using graph convolutional networks *BioRxiv* 2020.04.06.028266

Sato R and Ishida T 2019 Protein model accuracy estimation based on local structure quality assessment using 3D convolutional neural network *PLoS One* **14** e0221347

Scarselli F, Gori M, Tsoi A C, Hagenbuchner M and Monfardini G 2009 The graph neural network model *IEEE Trans. Neural Netw.* **20** 61–80

Schütt K, Kindermans P-J, Felix H E S, Chmiela S, Tkatchenko A and Müller K-R 2017 SchNet: a continuous-filter convolutional neural network for modeling quantum interactions *Advances in Neural Information Processing Systems* pp 991–1001

Senior A W *et al* 2019 Protein structure prediction using multiple deep neural networks in the 13th critical assessment of protein structure prediction (CASP13) *Proteins: Struct. Funct. Bioinform.* **87** 1141–8

Senior A W *et al* 2020 Improved protein structure prediction using potentials from deep learning *Nature* **577** 706–10

Shen M-yi and Sali A 2006 Statistical potential for assessment and prediction of protein structures *Protein Sci.* **15** 2507–24

Sun M, Zhao S, Gilvary C, Elemento O, Zhou J and Wang F 2020 Graph convolutional networks for computational drug development and discovery *Briefings Bioinform.* **21** 919–35

Thomas N, Smidt T, Kearnes S, Yang L, Li Li, Kohlhoff K and Riley P 2018 Tensor field networks: rotation-and translation-equivariant neural networks for 3D point clouds (arXiv:1802.08219)

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

Wallner Born and Elofsson A 2003 Can correct protein models be identified? *Protein Sci.* **12** 1073–86

Wang Z, Tegge A N and Cheng J 2009 Evaluating the absolute quality of a single protein model using structural features and support vector machines *Proteins: Struct. Funct. Bioinform.* **75** 638–47

Weiler M, Geiger M, Welling M, Boomsma W and Cohen T 2018 3D steerable CNNs: learning rotationally equivariant features in volumetric data *Advances in Neural Information Processing Systems* pp 10381–92

Won J, Baek M, Monastyrskyy B, Kryshtafovych A and Seok C 2019 Assessment of protein model structure accuracy estimation in CASP13: challenges in the era of deep learning *Proteins: Struct. Funct. Bioinform.* **87** 1351–60

Zamora-Resendiz R and Crivelli S 2019 Structural learning of proteins using graph convolutional neural networks *bioRxiv* 610444

Zemla A, Venclovas Česlovas, Moult J and Fidelis K 1999 Processing and analysis of CASP3 protein structure predictions *Proteins: Struct. Funct. Bioinform.* **37** 22–9

Zhang J and Zhang Y 2010 A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction *PLOS One* **5** e15386

Zheng W, Yang Li, Zhang C, Pearce R, Mortuza S M and Zhang Y 2019 Deep-learning contact-map guided protein structure prediction in CASP13 *Proteins: Struct. Funct. Bioinform.* **87** 1149–64

Zhou H and Skolnick J 2011 GOAP: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction *Biophys. J.* **101** 2043–52

Zhou H and Zhou Y 2002 Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction *Protein Sci.: A Pub. Protein Soc.* **11** 2714–26