Quality prediction of proteins models with spherical convolutions on three-dimensional graphs

***Nikita Pavlichenko*[1], *Sergei Grudinin*[2], *Ilia Igashov*[1,2].**
pavlichenko.nv@phystech.edu, sergei.grudinin@inria.fr, igashov.i@yandex.ru
[1] Moscow Institute of Physics and Technology, 141701 Dolgoprudniy, Russia;
[2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

Convolutional neural networks have become very popular in recent years, and, in particular, have found widespread application in computer vision. Recently, active work has also begun on graph convolutional networks. In general, the graphs, unlike the pictures, are irregular structures, and in many tasks of learning on graphs sample objects also do not have unified topology. Therefore, the existing operations of convolution on the graphs are very much simplified, and the task of pulling on the graphs remain open in general. The purpose of this work is to research new operations of convolution on three-dimensional graphs within the framework of solving the problem of quality estimation of three-dimensional models of proteins (the problem of regression on the graph nodes). These operations are theoretical mechanics inspired methods based on the expansion of a function of spherical coordinates as a linear combination of spherical harmonics. It helps solve the problem of capturing some local 3D structure of protein residues.

Key words: *graph convolutional networks, spherical convlutions, three-dimensional graphs learning.*

1 Introduction

Protein molecules are an important part of any biological form. They determine cellular functions and behavior of various biological and chemical structures. It makes the discovery and prediction of proteins structure one of the most important points of medical, chemical and genetic science researches.

Molecules of proteins consist of smaller molecules called amino acids. These amino acids form a chain that is folded and placed in space. Thus, protein functions are determined by their positions in a 3D space. So, having this chain of amino acids we need to identify how they are located. There are ways to do this experimentally, but it can be time-consuming, expensive, and not always possible. To solve these disadvantages, computational algorithms [1] [9] [16] were developed that generate different chain foldings. The problem is that no algorithm is the best one. Some of the proteins are better modeled by one algorithm, others by others. Therefore, we are facing the problem of quality assessment (QA) of these protein models.

This problem has recently got attention from the machine learning community. Various artificial intelligence methods were applied such as neural networks [15] and support vector machines [13] [14]. More recent approaches mostly include deep learning methods [5] [4] [12] [3]. The newest approach is to use graph machine learning methods such as Graph Convolutional Networks (GCN) [2], where the protein is in some way represented as a graph. This work brings the new idea of capturing the 3D structure of this graph to improve the quality of GCN using convolutions based on spherical harmonics.

2 Problem statement

Consider a 3D model of a protein in space. The protein represents a chain of amino acids rolled up in space. Through dividing space around a protein into cells, for example, by the

Voronoi method, we can get a 3D-graph, the vertices of which are amino acids of protein, and edges are carried out between those amino acids that are in adjacent cells. Denote the resulting graph by $G = (V, E)$, where verticles $V = (v_1, \ldots, v_n)$ are a set of amino acids, E are edges. For the i-th vertex, we denote for $\mathcal{N}(v_i)$ the set of its neighbors in a graph G and for G an adjacency matrix $\boldsymbol{A}$:

$$A_{ij} = \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & \text{otherwise.} \end{cases}$$

Consider that each vertex v_i is described by some real d-dimensional vector of attributes $x(v_i) = \boldsymbol{x}_i$. In the simplest case, it can be a one-hot representation of an amino acid type. We can form feature matrix $\boldsymbol{X}$ from vectors $x(v_i)$ for every vertex x_i. Using these data we will solve a regression problem: to predict for each vertex v_i a real number - its "score". In other words, how correctly it is placed in the given 3D-model in comparison with the actual conformation of this protein.

3 Graph Convolutional Networks

We will develop an approach for machine learning on graphs using Graph Convolutional Network (GCN) [7]. The idea of this approach is an aggregation of neighbor features for every vertex and forming new feature vectors on the layer's output. So, for GCN with L layers, $L \geqslant 2$ we have:

$$A_{ij} = \begin{cases} \boldsymbol{H}^{(0)} = \boldsymbol{X} \\ \boldsymbol{H}^{(l)} = \sigma\left(\boldsymbol{A}\boldsymbol{H}^{l-1}\boldsymbol{W}^{(l-1)}\right) \quad \text{for } l \in \{1, \ldots, L-1\} \\ \boldsymbol{H}^{(L)} = \boldsymbol{A}\boldsymbol{H}^{L-1}\boldsymbol{W}^{(L-1)}. \end{cases}$$

In the above $\boldsymbol{H}^{(l)}$ denotes feature matrix at l-th layer and $\boldsymbol{W}^{(l)}$ denotes weight matrix. Common choice for activation function σ is a ReLU activation: $\mathrm{ReLU}(x) := \max(x, 0)$ or an ELU activation: $\mathrm{ELU}(x) := \begin{cases} x, & x \geqslant 0 \\ \alpha(e^x - 1), & x < 0 \end{cases}$. Weights matrix $\boldsymbol{W}^{(l)}$ are trained by stochastic gradient descent or its modifications.

We rely on this approach because it has already shown outperformance in the protein quality assessment problem [2].

The main problem here is that classic GCN is not able to use information about locations of vertices in space. This is because the coordinates in space are given ambiguously: the turned or shifted proteins are actually isomorphic. So our improvement is to capture such information from the sequential structure of a protein.

4 Spherical Convolutional Networks

4.1 Spherical harmonics

Let us consider a function $f(\Omega) : [0, \pi] \times [0, 2\pi) \to \mathbb{R}$ — mapping from unit sphere to real numbers. That function might be unknown or its evaluation might be time-consuming. So, we want to expand this function as a linear combination of simpler functions. The common approach that came from theoretical mechanics is to use spherical harmonics [10]. We will use the following form of spherical functions:

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

where P_l^m are associated Legendre polynomials [10].

On the unit sphere, any square-integrable function can thus be expanded as a linear combination of these:

$$f(\varphi, \psi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} f_l^m Y_l^m(\varphi, \psi).$$

We will use this fact to train a function of vertex features using the first few components of this expansion. Since the function is real, the real representation of spherical harmonics can be used:

$$Y_{lm} = \begin{cases} \sqrt{2}(-1)^m \operatorname{Im}\left[Y_l^{|m|}\right] & \text{if } m < 0 \\ Y_l^0 & m = 0 \\ \sqrt{2}(-1)^m \operatorname{Re}\left[Y_l^m\right] & \text{if } m > 0. \end{cases}$$

4.2 Spherical convolution

Consider a vertex v_i. Since all the amino acids in the protein are connected in a peptide chain, it is easy to construct its local coordinate system for the amino acid v_i under consideration — on two dihedral corners, which are unambiguously determined from the geometry of the peptide chain. Write the coordinates of all the neighbors of $v_j \in \mathcal{N}(v_i)$ of the amino acid in the obtained coordinate system. Then proceed to spherical coordinates, and project all vertices onto a unit sphere with the center in v_i. Now, each vertex $v_j \in \mathcal{N}(v_i)$ can be matched with a pair of angles $\Omega_i^j = (\varphi_i^j, \psi_i^j)$ that specify the angular position of the projection of the vertex v_j onto a unit sphere in the local coordinate system of v_i.

Now, having an unambiguous orientation for each vertex of the graph, we can introduce the convolution operation. Let us consider a certain matrix function $f(\Omega) : [0, \pi] \times [0, 2\pi) \to \mathbb{R}^{d \times d'}$ on a single sphere. We can expand it into a series on the basis of spherical functions $\{Y_l^m\}_{l,m}$ and leave the first few components:

$$f(\Omega) \approx f_W(\Omega) = \sum_{l=0}^{L} \sum_{m} \boldsymbol{W}_l^m Y_l^m(\Omega),$$

where $\boldsymbol{W}_l^m$ denotes coefficient matrix in the expansion of matrix function f on the basis of $\{Y_l^m\}_{l,m}$. Then we can introduce the spherical convolution operation for the vertex v_i in the following way:

$$f_W \circ v_i = \sum_{v_j \in \mathcal{N}(v_i)} f_W(\Omega_i^j) x(v_i).$$

Considering $\boldsymbol{W}_l^m$ matrices to be optimized parameters, we will thus train spherical filters.

4.3 Spherical convolution layer

Let us have a 3D model of a protein consisting of N amino acids, the i-th amino acid is described by the trait vector $\boldsymbol{x}_i \in \mathbb{R}^d$. Let us denote all the vertices through the $\boldsymbol{X} \in \mathbb{R}^{N \times d}$ feature matrix. Then one layer of spherical convolution is written down as follows:

$$\boldsymbol{X} \longrightarrow \boldsymbol{X}' = \sigma(f_W \circ \boldsymbol{X}) = \sigma\left(\sum_{l,m} Y_l^m(\boldsymbol{A}_\Omega) \boldsymbol{X} \boldsymbol{W}_l^m\right),$$

where σ is an activation function, Y_l^m are spherical functions, $\boldsymbol{W}_l^m$ are optimized parameters and $\boldsymbol{A}_\Omega$ is the adjacency matrix of graph G, which cells contain the spherical coordinates of

the vertices in the corresponding local coordinate system:

$$[\boldsymbol{A}_\Omega]_{i,j} = \begin{cases} \Omega_i^j, & (v_i, v_j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

5 Experiment

5.1 Dataset

All models are taken from the CASP competition data (`http://predictioncenter.org`), which is dedicated to the prediction of the 3D structure of proteins by the amino acid sequence. We take data from CASP12 and CASP13 as a test sample and data from earlier CASPs as a training sample.

For each protein, there is one real experimentally obtained 3D structure and many generated 3D structures. Our task is to predict the CAD-score [11] for each of the generated structures i.e. how similar each of the generated structures is to the real one. Graphs are generated using Voronoi diagrams with weights — an area of contact.

We have precalculated spherical harmonic of the order of 5 for every pair of incident residues. This took 60GB of HDD space when calculated on CASP8-CASP12. So, it appears to be the main problem there, comparing to the fact that raw data takes only 15GB and that value grows as n^2, where n is an order of spherical harmonic.

5.2 Metrics

In this task, there is a problem of selecting the right quality metric. It was found that MSE's conventional metric used for training was not suitable for evaluating the quality of the algorithm. This is due to the fact that it does not reflect what the algorithm was created for at all — to select protein folding models. So, we need a more suitable metric. Often Pearson correlation between CAD models or residues and algorithm predictions is chosen for this purpose [2]. We will also use a **z-score** metric because it represents how good is the model with the maximum predicted score.

We will calculate these metrics for every protein, and then average them. As a result, we will get the global metric value.

5.3 Baseline

For the baseline, we will use an architecture inspired by GraphQA [2] network. Firstly, we encode residues features to high-dimensional space using linear layers with size increasing by a power of two. For the next step, we process obtained features through 8 graph convolutional layers to account for the relationships of residues. Finally, we pass the result through a multi-layer perceptron and get single output with a sigmoid activation function as the CAD score is from 0 to 1. We have chosen ELU as an activation function for hidden layers. Adam optimizer [6] is used for training with a learning rate set to 0.001.

This model was trained on CASP8, CASP9, CASP10, CASP11 datasets for 50 iterations and evaluated on CASP12. We decided to train it on CPU because the slowest part of the training was the loading features and adjacency matrices from HDD.

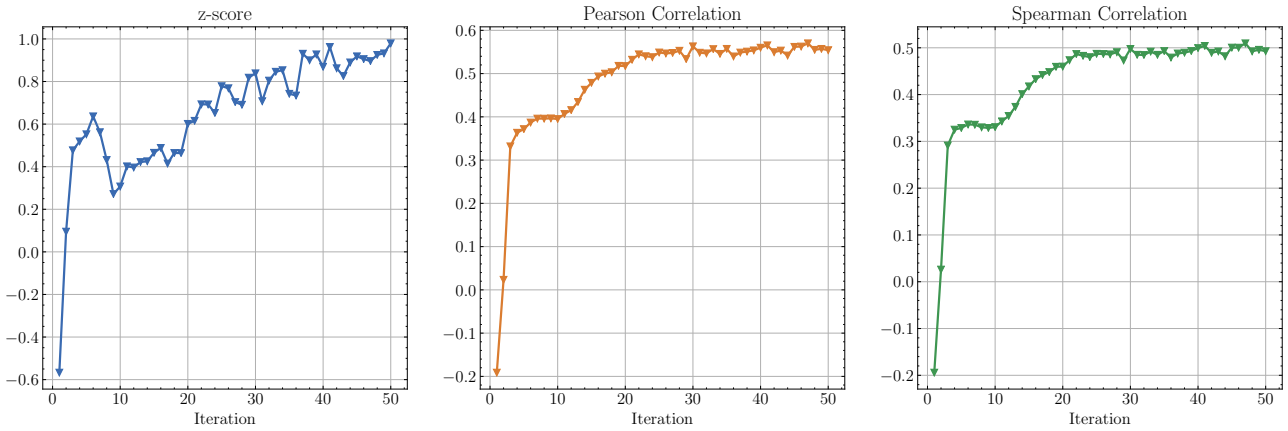

Figure 1 z-score and Pearson and Spearman correlation coeffients for each iteration on CASP12 dataset of baseline GCN.

We managed to achieve the maximum Pearson correlation value of 0.57 and the maximum z-score value of 0.979. It is significantly lower than many state-of-the-art algorithms [2] but it can be explained by the fact that we only used the geometric structure of proteins. All the high results were obtained only with the use of complex features that represent chemical relations between residues and some biological features that represent protein evolution [2].

5.4 Spherical Convolutional Networks

We have run the same architecture with spherical convolutional layers instead of GCN. The main problem there was the fact, that the number of trained parameters has increased in 25 times because we have used spherical harmonics of the order of 5. So, we have decreased the learning rate and started learning on GPU because the loading from HDD now is not a bottleneck.

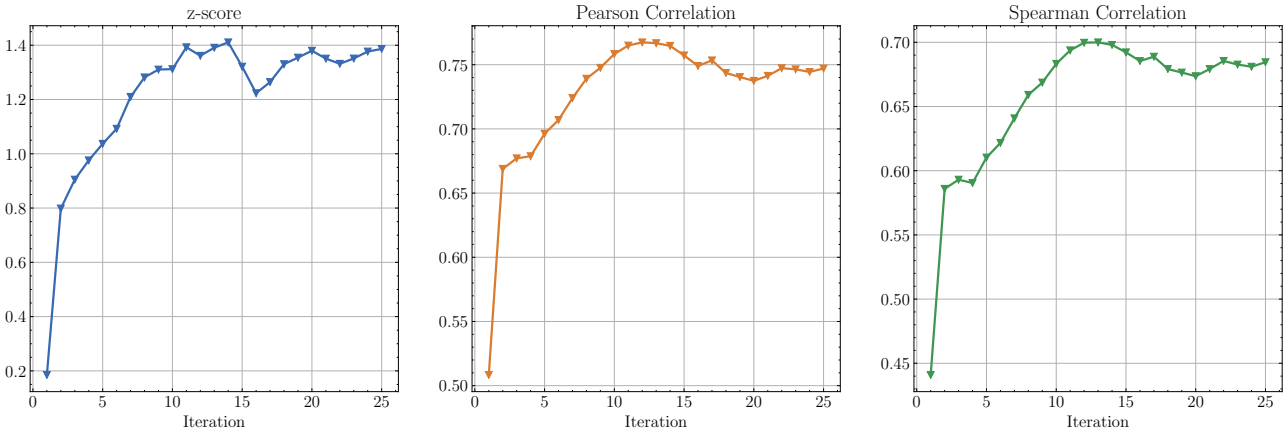

Figure 2 z-score and Pearson and Spearman correlation coeffients for each iteration on CASP12 dataset of SCN.

5.5 Comparsion

Table 1 the performance of state-of-the-art methods of Protein Quality Assessment. The important remark there is the fact that other approaches shown in this table use biological and chemical features but in turn, we use only the geometric representation of residues graph. Despite that fact, Spherical Convolutional Network has shown high performance.

	ρ	r	rank	z-score
ProQ3D	0.801	0.750	11.961	1.670
VoroMQA	0.803	0.766	17.171	1.410
SBROD	0.685	0.762	23.579	1.282
Ornate	0.828	0.781	10.776	1.780
Simple GCN	0.570	0.510	31.667	0.979
SCN	0.767	0.700	18.462	1.411

Table 1 Comparsion of Pearson, Spearman correlation coefficients and z-score and rank of the state-of-the-art algorithms, our GCN baseline and Spherical Convolutional Network on CASP12 dataset.

We see that Spherical Convolutional Network has shown significantly better performance than GCN. It shows that our approach is applicable and appears to be an improvement to Graph Convolutional Networks.

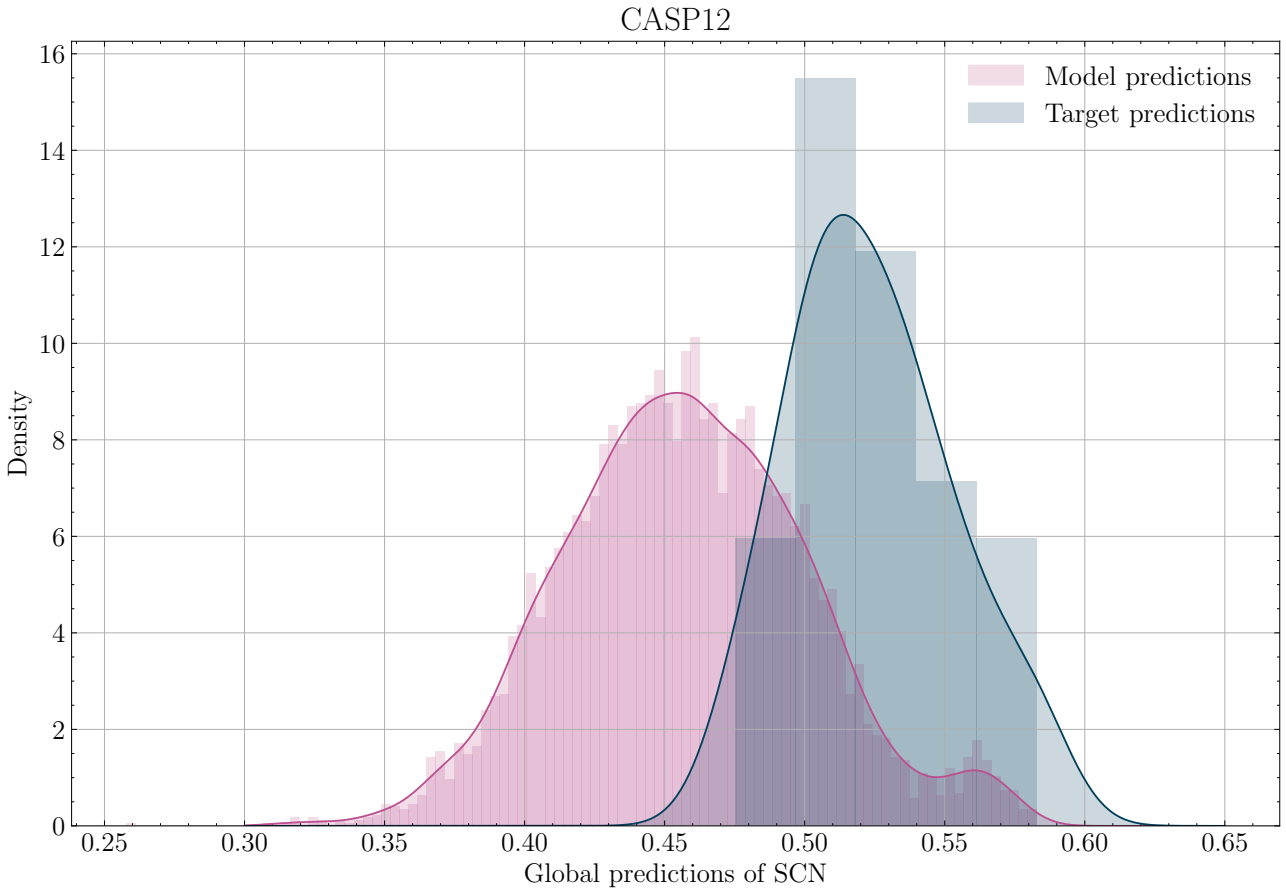

Figure 3 Distribution of SCN scores on target structures and models from CASP12. Solid lines represent kernel density estimations of the corresponding distributions.

6 Conclusion

For the first time, we applied spherical convolutions for capturing the 3D structure of the graph. The results have shown that it could give significant improvement in the quality of the

algorithm. This idea can be combined with other ideas introduced in other papers [12] [2] [14] so we hope that it can achieve even higher results in the problem of Protein Quality Assessment. We want to notice that the idea of spherical convolutions is universal and can be applied to various types of tasks of learning on graphs with 3D structure and hope that it will find such applications.

Finally, we wish that this idea will be developed and be applied in more complicated approaches of learning on graphs such as graph variational autoencoders [8] and Quality Assessment will become a popular task in the machine learning community.

References

[1] Konstantin Arnold, Lorenza Bordoli, Jürgen Kopp, and Torsten Schwede. The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling. *Bioinformatics*, 22(2):195–201, nov 2005.

[2] Federico Baldassarre, Hossein Azizpour, David Menéndez Hurtado, and Arne Elofsson. Graphqa: Protein model quality assessment using graph convolutional network. 2020.

[3] Matthew Conover, Max Staples, Dong Si, Miao Sun, and Renzhi Cao. AngularQA: Protein model quality assessment with LSTM networks. feb 2019.

[4] Georgy Derevyanko, Sergei Grudinin, Yoshua Bengio, and Guillaume Lamoureux. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, 34(23):4046–4053, jun 2018.

[5] David Menéndez Hurtado, Karolis Uziela, and Arne Elofsson. Deep transfer learning in the assessment of the quality of protein models.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization.

[7] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks.

[8] Thomas N. Kipf and Max Welling. Variational graph auto-encoders.

[9] Jesper Lundström, Leszek Rychlewski, Janusz Bujnicki, and Arne Elofsson. Pcons: A neural-network-based consensus predictor that improves fold recognition. *Protein Science*, 10(11):2354–2362, dec 2008.

[10] Claus Müller. *Spherical Harmonics*. Springer Berlin Heidelberg, 1966.

[11] Kliment Olechnovič, Eleonora Kulberkytė, and Česlovas Venclovas. CAD-score: A new contact area difference-based function for evaluation of protein structural models. *Proteins: Structure, Function, and Bioinformatics*, 81(1):149–162, sep 2012.

[12] Guillaume Pagès, Benoit Charmettant, and Sergei Grudinin. Protein model quality assessment using 3d oriented convolutional neural networks. *Bioinformatics*, 35(18):3313–3319, feb 2019.

[13] Arjun Ray, Erik Lindahl, and Björn Wallner. Improved model quality assessment using ProQ2. *BMC Bioinformatics*, 13(1):224, 2012.

[14] Karolis Uziela, Nanjiang Shu, Björn Wallner, and Arne Elofsson. ProQ3: Improved model quality assessments using rosetta energy terms. *Scientific Reports*, 6(1), oct 2016.

[15] Björn Wallner and Arne Elofsson. Can correct protein models be identified? *Protein Science*, 12(5):1073–1086, may 2003.

[16] Jinbo Xu. Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences*, 116(34):16856–16865, aug 2019.