Protein Interface Prediction using Graph Convolutional Networks

https://papers.nips.cc/paper/7231-protein-interface-prediction-using-graph-convolutional-networks.pdf

Reviewed by: Eric Wang
University of Virginia
https://qdata.github.io/deep2Read/

Protein Interface Prediction using Graph Convolutional Networks

Alex Fout[†]
Department of Computer Science
Colorado State University
Fort Collins, CO 80525
fout@colostate.edu

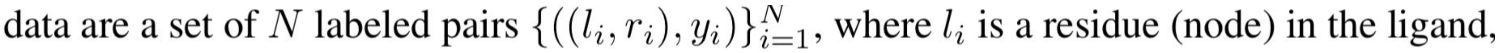
Jonathon Byrd[†]
Department of Computer Science
Colorado State University
Fort Collins, CO 80525
jonbyrd@colostate.edu

Basir Shariat[†]
Department of Computer Science
Colorado State University
Fort Collins, CO 80525
basir@cs.colostate.edu

Asa Ben-Hur
Department of Computer Science
Colorado State University
Fort Collins, CO 80525
asa@cs.colostate.edu

Intro

- Images are represented as feature values on a 2D grid, whereas the solved crystal structure of a protein can be thought of as a collection of features on an irregular 3D grid corresponding to the coordinates of its atoms. In both cases, we are trying to recognize an object within a larger context.
- The neighborhood of a node used in the convolution operator is the set of **k closest** residues as determined by the **mean distance** between their atoms.
- Task is to classify pairs of nodes from two separate graphs representing those proteins.

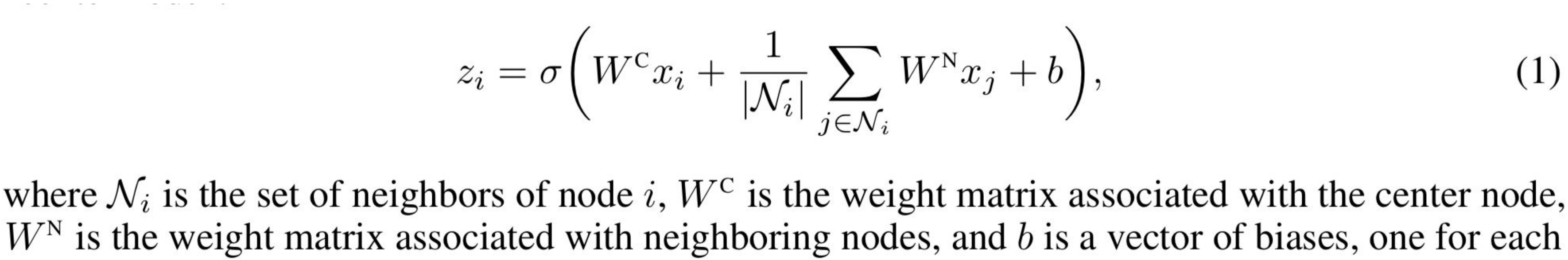


data are a set of N labeled pairs $\{((l_i, r_i), y_i)\}_{i=1}^{N}$, where l_i is a residue (node) in the ligand,

- Ri is a residue (node) in the receptor protein, and yi ∈ {−1, 1} is the associated label that indicates if the two residues are interacting or not

Graph Convolution

- xi the feature vector associated with node i
- Aij the feature vector associated with the edge between nodes i and j
- To design convolution operators that can be applied to graphs without a regular structure, and without imposing a particular order on the neighbors of a given node



$$z_i = \sigma\left(W^{\mathrm{C}} x_i + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} W^{\mathrm{N}} x_j + b \right), \tag{1}$$

where $\mathcal{N}_i$ is the set of neighbors of node i, W^{C} is the weight matrix associated with the center node, W^{N} is the weight matrix associated with neighboring nodes, and b is a vector of biases, one for each

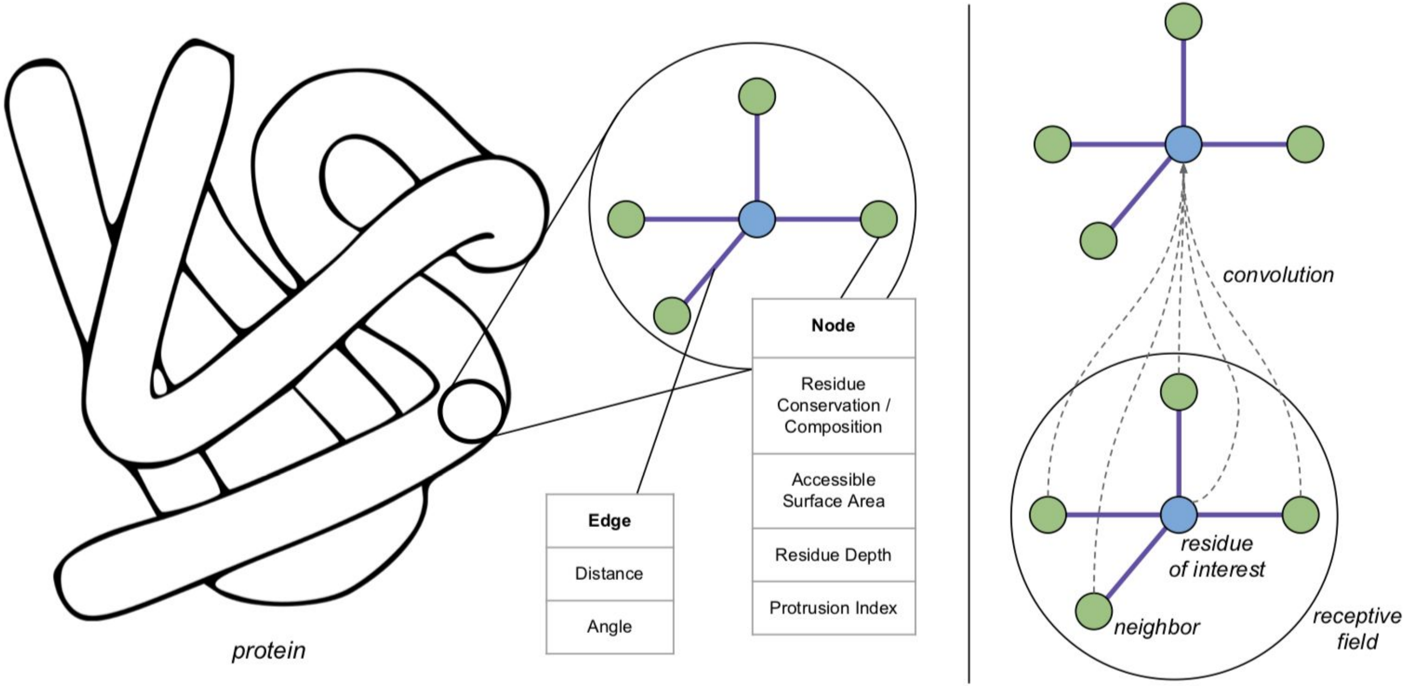


Figure 1: Graph convolution on protein structures. Left: Each residue in a protein is a node in a graph where the neighborhood of a node is the set of neighboring nodes in the protein structure; each node has features computed from its amino acid sequence and structure, and edges have features describing the relative distance and angle between residues. Right: Schematic description of the convolution operator which has as its receptive field a set of neighboring residues, and produces an activation which is associated with the center residue.

In order to provide for some differentiation between neighbors, we incorporate features on the edges between each neighbor and the center node as follows:

$$z_i = \sigma\left(W^{\mathrm{C}} x_i + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} W^{\mathrm{N}} x_j + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} W^{\mathrm{E}} A_{ij} + b \right), \tag{2}$$

where W^{E} is the weight matrix associated with edge features.

For comparison with order-independent methods we propose an order-dependent method, where order is determined by distance from the center node. In this method each neighbor has unique weight matrices for nodes and edges:

$$z_i = \sigma\left(W^{\mathrm{C}} x_i + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} W_j^{\mathrm{N}} x_j + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} W_j^{\mathrm{E}} A_{ij} + b \right). \tag{3}$$

Here W_j^{N}/W_j^{E} are the weight matrices associated with the j^{th} node or the edges connecting to the j^{th}

Graph Convolution cont.

- Our graph operators on the other hand maintain the **structure of the graph**, which is necessary for the protein interface prediction problem, where we classify **pairs of nodes from different graphs**, rather than entire graphs.

- Using convolutional architectures that use only convolutional layers **without downsampling** is common practice in the area of graph convolutional networks, especially if classification is performed at the **node or edge level**.

- This practice has support from the success of networks without pooling layers in the realm of object recognition. The downside of not downsampling is **higher memory and computational costs**.

Pairwise classification architecture

- Two approaches:
    - One is to construct explicit features that are **order invariant** by taking the **sum and element-wise products** of the two feature vectors. Note that pairwise kernels implicitly use *all* products of features, which we avoid by taking the element wise product.
    - Another approach is to present each example to the model in both possible orders, (li, ri) and (ri, li), and **average the two** predictions; the feature representation of an example is the **concatenation** of the features of the two residues.
- Both approaches yielded similar results

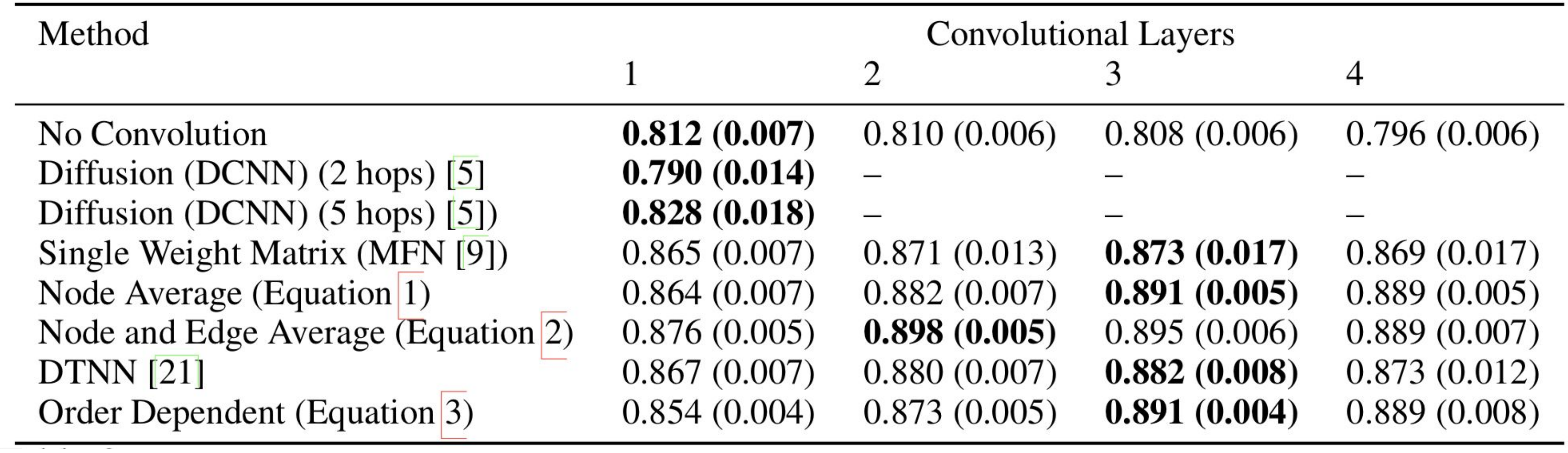
Data

- Version 5 of the **Docking Benchmark Dataset**, which is the standard benchmark dataset for assessing docking and interface prediction methods
- The structures are generated from x-ray crystallography or nuclear magnetic resonance experiments and contain the **atomic coordinates of each amino acid** residue in the protein
- Two residues from different proteins are considered part of the interface if any non-Hydrogen atom in one is **within 6Å** of any non-Hydrogen atom in the other when in complex
- Because in any given complex there are vastly more residue pairs that don’t interact than those that do, we **downsampled the negative examples** in the training set to obtain a 10:1 ratio of negative and positive examples.

Node and edge features

- Each node and edge in the graph representing a protein has features associated with it that are computed from the protein’s **sequence and structure**.
- 1. Protein **sequence** alone can be a good indicator of the propensity of a residue to form an interface, because each amino acid exhibits unique electrochemical and geometric properties.
    - The level of conservation of a residue in **alignments against similar proteins** also provides valuable information, since surface residues that participate in an interface tend to be **more conserved** than surface residues that do not.
- 2. Each node contains several features computed from the **structure**: residue surface accessibility, a measure of its protrusion, its distance from the surface, and the counts of amino acids within 8Å in two directions—towards the residue’s side chain, and in the opposite direction.

Result



Method	Convolutional Layers			
	1	2	3	4
No Convolution	**0.812 (0.007)**	0.810 (0.006)	0.808 (0.006)	0.796 (0.006)
Diffusion (DCNN) (2 hops) [5]	**0.790 (0.014)**	–	–	–
Diffusion (DCNN) (5 hops) [5])	**0.828 (0.018)**	–	–	–
Single Weight Matrix (MFN [9])	0.865 (0.007)	0.871 (0.013)	**0.873 (0.017)**	0.869 (0.017)
Node Average (Equation 1)	0.864 (0.007)	0.882 (0.007)	**0.891 (0.005)**	0.889 (0.005)
Node and Edge Average (Equation 2)	0.876 (0.005)	**0.898 (0.005)**	0.895 (0.006)	0.889 (0.007)
DTNN [21]	0.867 (0.007)	0.880 (0.007)	**0.882 (0.008)**	0.873 (0.012)
Order Dependent (Equation 3)	0.854 (0.004)	0.873 (0.005)	**0.891 (0.004)**	0.889 (0.008)

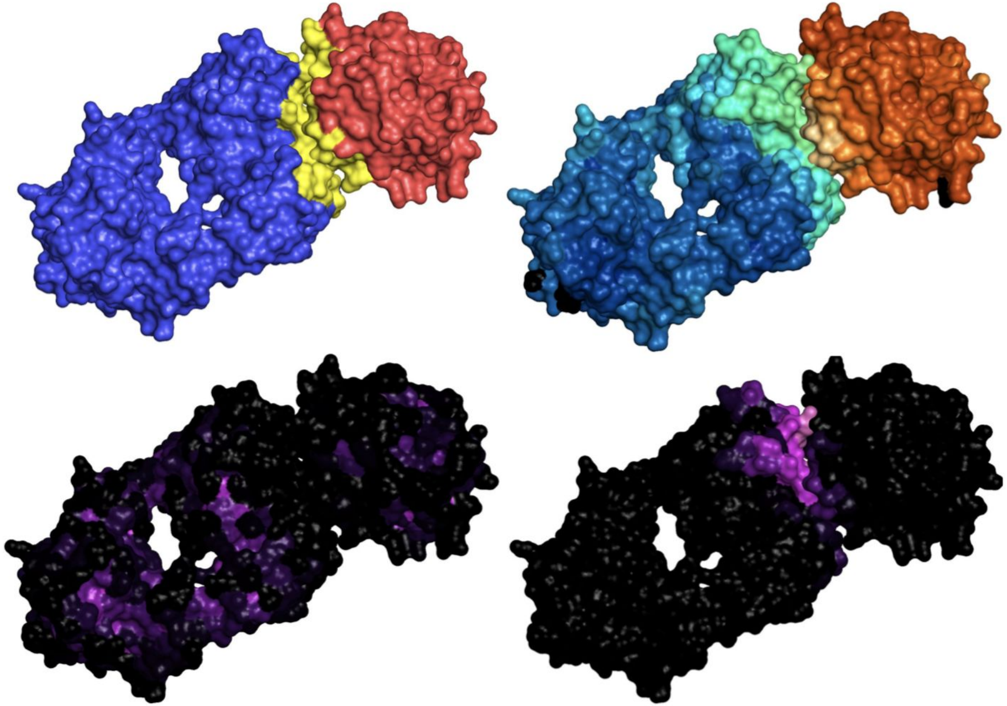


Figure 3: PyMOL [20] visualizations of the best performing test complex (PDB ID 3HI6). Upper left: Ligand (red) and receptor (blue), along with the true interface (yellow). Upper right: Visualization of predicted scores, where brighter colors (cyan and orange) represent higher scores. Since scores are for pairs of residues, we take the max score over all partners in the partner protein. Bottom row: Activations of two filters in the second convolutional layer, where brighter colors indicate greater activation and black indicates activation of zero. Lower left: A filter which provides high activations for buried residues, a useful screening criterion for interface detection. Lower right: Filter which gives high activations for residues near the interface of this complex.

Conclusion

- Our experiments did not demonstrate a big difference with the **inclusion of edge features**. There were very few of those, and unlike the node features, they were **static**: our networks learned latent representations only for the node features.
- CNNs typically require large datasets to learn effective representations. This may have limited the level of accuracy that we could attain using our **purely supervised** approach and the relatively **small number of labeled training examples**. Unsupervised pre-training would allow us to use the entire Protein Data Bank which contains close to 130,000 structures
- In designing our methodology we considered the question of the appropriate level at which to describe protein structure. In classifying image data, CNNs are usually applied to the raw pixel data. The analogous level of description for protein structure would be the **raw 3D atomic coordinates**, which we thought would prove too difficult. Using much larger training sets and unsupervised learning can potentially allow the network to begin with features that are closer to the raw atomic coordinates and learn a more detailed representation of the geometry of proteins.