

A scoring method based on Graph Neural Networks for protein-protein docking

Yi-yang Liao, Qian-li Yang, Jun-jie Ying, Jia-jun Zou, Jun Xie

Xiamen University

{23020221154165,36920221153135,36920221153141,31520221154243,36920221153129}@stu.xmu.edu.cn

Abstract

The physical interaction of proteins plays a key functional role in many important cellular processes. To understand the molecular mechanisms underlying these functions, is crucial to determine the structure of protein complexes. However, using experimental methods to obtain the structure of protein complexes is very expensive in time and resources. In order to assist the experiment, a computational modeling method for protein structure has been proposed, which is called protein docking. One of the challenges in computational modeling is how to select the right protein from the large number of protein complexes generated. Here, we develop a deep learning-based approach to evaluate protein docking models, utilizing graph neural networks that extract interface regions and represent them as graphs, with chemical and physical characteristics of atoms calculated by the latest protein representation work named as dmasif as features of nodes and edges in the graph, respectively, trained, validated and tested on benchmark datasets. The results show that we only trained with part of the training set of the previous work, achieve the performance of the previous work, and perform better on some metrics.

Introduction

Proteins, variable-length chains of amino acid residues assembled from 20 typical amino acids, mediate the fundamental processes of life. Amino acids assemble to form polymer chains, which can fold into 3D structures whose shape largely determines the function of proteins. These folded structures can be described at four levels: primary structure, which simply captures the linear sequence of amino acids; secondary structure, which describes the local arrangement of amino acids, including structural motifs such as α -helices and β -sheets; tertiary structure, Describes the complete spatial arrangement of all residues; Quaternary structure, describing how multiple distinct amino acid chains can aggregate to form larger complexes(Sun, Foster, and Boyington 2004). The quaternary structure of proteins also be referred to as protein-protein interactions (PPIs). PPIs play a crucial role in cellular activities and are involved in many essential biological processes in cells(Berggård, Linse, and James 2007; Scott et al. 2016; Typas and Sourjik 2015). In order to understand the functional mechanism

of PPIs and the basic physicochemical information of protein complexes, the 3D structures of protein complexes have been determined experimentally. And with the development of cryo-electron microscopy (cryo-EM), the experimental structural biology community is determining the structures of protein complexes at a steady rate, but the structures of many important protein interactions remain undetermined. However, experimental methods are usually technically difficult and require a lot of time and resources. To complement the experimental approach, computational modeling methods for the complex structure of proteins, often referred to as protein docking(Aderinwale et al. 2020), have been extensively studied over the past 20 years.

Protein-protein docking aims to assemble protein complexes through two individual protein structures(Porter et al. 2019), which can be divided into two main categories: template-based methods and ab initio methods. Template-based methods use a known structure as a scaffold of modeling(Anishchenko et al. 2015), and ab initio docking assembles individual structures and scores the resulting models to select the most plausible model, which includes two basic processes: sampling and scoring(Huang 2014). Given two separate protein structures, protein docking attempts to sample the putative binding mode of one protein relative to the other. These sampled protein complexes are often referred to as decoys. Then, the sampled decoys are evaluated using a scoring function, which is expected to correctly identify the protein-protein complex conformation close to the native, giving the native decoy a better score to correctly predict complex conformations similar to the native.

As more and more PPIs are discovered in biological functions, protein-protein docking has made considerable progress in all aspects. However, how to distinguish near-native conformations from a large number of sampled decoys with an appropriate scoring function remains a long-standing challenge. Previous work has divided protein docking scoring methods into three categories: force-field-based, knowledge-based, and machine-learning based. Machine learning-based scoring functions can discover complex non-linear combinations of protein-protein interface features, so machine learning-based scoring functions may outperform traditional scoring functions. However, many machine learning-based scoring functions still fail to exploit spatially arranged details or higher-order interaction patterns on the

interface. GNN has the powerful function of modeling the dependencies between nodes in the graph, which has made a breakthrough in the field of graph analysis related research. Using GNN can capture all atoms on an interface of any size in a more flexible way, and we are trying to develop a graph neural network-based scoring method for protein-protein docking between proteins.

Related work

Protein docking methods are roughly classified into two categories, template-based modeling methods and ab initio methods. Template-based modeling methods can use local complex(Tuncbag et al. 2011) or global complex(Anishchenko et al. 2015) information. In reality, ab initio methods do not rely on prior information and therefore to be used more. In general, the task of protein-protein docking involves two steps: sample and scoring(Sunny and Jayaraj 2022). Among the sampling steps, the fast Fourier transform proposed in 1992(Katchalski-Katzir et al. 1992; Padhorny et al. 2016) has achieved great success; and other methods like particle swarm optimization(Moal and Bates 2010) and geometric hashing(Fischer et al. 1995; Venkatraman et al. 2009) also perform well. In the more recent progress, normal mode analysis(Oliwa and Shen 2015) and protein dynamics simulation(Gray et al. 2003) take protein flexibility into consideration.

Nevertheless, the more challenging step is the scoring of docking decoys(Moal et al. 2013). In the step of ab initio methods, a large number of decoys can often be generated, but only a few of them are near-native. And due to the large number of decoys, high-speed and high-precision scoring methods have an important impact on the performance of protein docking. The scoring methods that have been applied can be roughly divided into three categories: Force field-based(Akbal-Delibas et al. 2016; Degiacomi 2019; Gainza et al. 2020), knowledge-based(Lu, Lu, and Skolnick 2003; Huang and Zou 2008), and machine learning-based(Basu and Wallner 2016; Geng et al. 2020; Wang et al. 2020). Force field-based methods usually consider the different physics-related energy terms, such as van der Waals potential and electrostatic potential and always minimize them to reach the most stable state. Knowledge-based scoring methods generally convert existing knowledge on protein interactions into statistical potentials. There are also some scoring methods combined the Force field-based methods and knowledge-based methods(Vreven, Hwang, and Weng 2011; Zimmermann et al. 2012). At the same time, with the development of bioinformatics, machine learning-based methods are gradually emerging and have achieved impressive performance. By applying complex nonlinear models, machine learning methods can discover complex patterns in protein-protein interaction. Therefore, machine learning-based scoring methods may perform better than traditional scoring methods. Among the machine learning approach, the recent rise of deep learning methods can extract more complex features to make better classification results.

In this work, we apply a graph neural network (GNN) to capture all atoms at an interface of any size in a flexible manner. At the same time, GNN represent the interface is

rotationally invariant, meaning arbitrary rotations of a complex are accounted for. We also try to apply geometric vector perceptrons (GVPs)(Jing et al. 2020) to make it easier for a GNN to learn functions whose significant features are both geometric and relational.

Table 1: Dockground dataset splits for training and testing our model

Fold	PDB ID
1	1A2K,1E96,1HE1,1HE8,1WQ1,1F6M,1MA9,2BTF 1G20,1KU6,1T6G,1UGH,1YVB,2CKH,3PRO
2	1AKJ,1P7Q,2BNQ,1DFJ,1NBF,1R4M,1XD3 2BKR,1GPW,1HXY,1U7F,1UEX,1ZY8,2GOO,1EWY
3	1AVW,1BTH,1BUI,1CHO,1EZU,1OOK,1OPH,1PPF 1TX6,1XX9,2FI4,2KAI,1R0R,2SNI,3SIC
4	1BVN,1TMQ,1F51,1FM9,1A2Y,1G6V,1GPQ 1JPS,1WEJ,1L9B,1S6V,1W1I,2A5T,3FAP

Materials and Methodology

Dockground Benchmark

To train and test our model, we first used the Dockground dataset 1.0(Liu, Gao, and Vakser 2008). Docking decoys in this dataset were built by Gramm-X(Tovchigrechko and Vakser 2005). The dataset includes 58 target complexes, each with averages of 9.83 correct and 98.5 incorrect decoys. A decoy was considered as correct following the CAPRI criteria(Lensink et al. 2018), which consider interface root mean square deviation (iRMSD), ligand RMSD (lRMSD), and the fraction of native contacts (fnat). The iRMSD is the C α RMSD of interface residues with respect to the native structure. Interface residues in a complex are defined as all the residues within 10.0 Å from any residues of the other subunit. lRMSD is the C α RMSD of ligands when receptors are superimposed, and fnat is the fraction of contacting residue pairs, that is, residue pairs with any heavy atom pairs within 5.0 Å, that exist in the native structure. To remove redundancy, we grouped the 58 complexes using sequence alignment and TM-align (Zhang and Skolnick 2004). Two complexes were assigned to the same group if at least one pair of proteins from the two complexes had a TM-score of over 0.5 and sequence identity of 30% or higher. This resulted in 29 groups (Table 1). In Table 1, complexes (PDB IDs) of the same group are shown in lower case in a parenthesis followed by the PDB ID of the representative. These groups were split into four subgroups to perform four-fold cross-validation, where three subsets were used for training, while one testing subset was used for testing the accuracy of the model. Thus, by cross-validation, we have four models tested on four independent testing sets. Among the training set, we used 80% of the complexes (i.e., unique dimers) for training a model and the remaining 20% of the complexes as a validation set, which was used to determine the best hyper-parameter set for training. In the results, the accuracy of targets when treated in the testing set was reported.

Table 2: Atom features

Features	Representation
Atom type	C,N,O,S,F,P,Cl,Br,B,H(one hot)
The degree(connections) of atom	0,1,2,3,4,5(one hot)
The number of connected hydrogen atoms	10,1,2,3,4(one hot)
The number of implicit valence electrons	0,1,2,3,4,5(one hot)
The chemical features	Six values from 0 to 1
Aromatic	0 or 1

CAPRI Benchmark

The CAPRI score set(Lensink and Wodak 2014) was used as an external test set. It consists of 13 protein dimers for a total of 16 666 models generated by over 40 different research teams using a variety of software.Each decoy set included 500 to 2,000 models generated using different methods by CAPRI participants.It is acknowledged as the most diverse set of docking models with targets of different complexity.

Our Model

In this section, we describe our model, which uses the graph neural network.This algorithm is inspired by a recent work in drug–target interactions(Lim et al. 2019), which designed a two-graph representation for capturing intermolecular interactions for protein–ligand interactions. We will first explain how the 3D structural information of a protein–complex interface is embedded as a graph. Then, we describe how we used a graph attention mechanism to focus on the intermolecular interaction between a receptor and a ligand protein. The overall protocol is illustrated in Figure 1.For an input protein docking decoy, the interface region is identified as a set of residues located within 10.0 Å of any residues of the other protein. A residue–residue distance is defined as the shortest distance among any heavy

atom pairs across the two residues. Using the extracted interface region, two graphs are built representing two types of interactions: the graph G1 describes heavy atoms at the interface region, which only considers the covalent bonds between atoms of interface residues within each subunit as edges. Another graph G2 connects both covalent (thus includes G1) and non-covalent residue interaction as edges, where a non-covalent atom pair is defined as those which are closer than 10.0 Å of each other. Both graphs will be processed by a graph neural network (GNN) to output a score, which is a probability that the docking decoy has a CAPRI acceptable quality (thus making higher scores better).

Building Graphs

A key feature of this work is the graph representation of an interface region of a complex model. Graph G is defined by $G(V, E, \text{ and } A)$, where V denotes the node set, E is a set of edges, and A is the adjacency matrix, which numerically represents the connectivity of the graph. For a graph G with N nodes, the adjacency matrix A has a dimension of $N \times N$, where $A_{ij} > 0$ if the i -th node and the j -th node are connected, and $A_{ij} = 0$ otherwise. The adjacency matrix A^1 for graph G1 describes covalent bonds at the interface and thus defined as follows:

$$A_{ij}^1 = \begin{cases} 1 & \text{if atom } i \text{ and atom } j \text{ are connected by a covalent bond or if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$A_{ij}^2 = \begin{cases} A_{ij}^1, & \text{if } i, j \in \text{receptor or } i, j \in \text{ligand} \\ e^{-\frac{(d_{ij}-\mu)^2}{\sigma}}, & \text{if } d_{ij} \leq 10\text{\AA} \text{ and } i \in \text{receptor or } j \in \text{ligand} \\ & \text{or if } d_{ij} \leq 10\text{\AA} \text{ and } j \in \text{receptor and } i \in \text{ligand} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The matrix A^2 for G2 describes both covalent bonds and noncovalent interactions between atoms within 10.0 Å to each other. It is defined as follows:

where d_{ij} denotes the distance between the i -th and the j -th atoms. μ and σ are learnable parameters, whose initial values are 0.0 and 1.0, respectively. The formula $e^{-(d_{ij}-\mu)^2/\sigma}$ decays as the distance increases between atoms.

As for the node features in the graph, we considered the physicochemical properties of atoms. We used the same features as used in previous works(Lim et al. 2019; Torng and

Altman 2019) as shown in Table 2. Thus, the length of a feature vector of a node from Table 2 was 34 (10 + 6+5 + 6 + 6 + 1), which was embedded by a one-layer fully connected (FC) network into 140 features.

Generate the chemical features

For the chemical features of atoms, we calculated using dmasif, the state of the art protein characterization work. This work present a new framework for deep learning on protein structures that addresses these limitations. Among

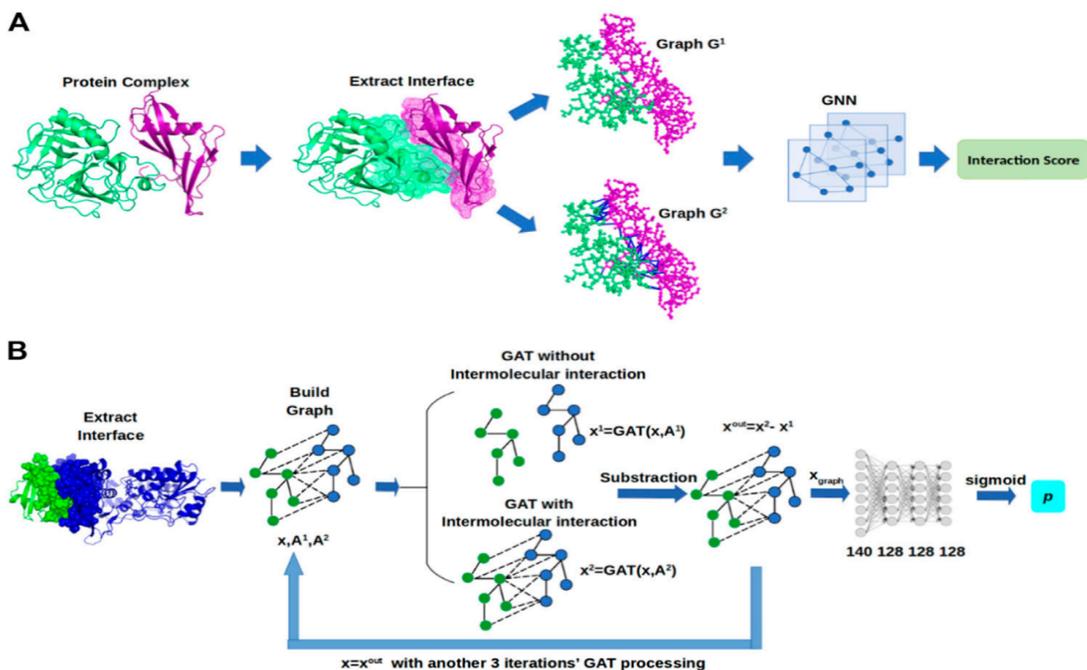


Figure 1: Framework of GNN-DOVE. GNN-DOVE extracts the interface region of protein complex and further reconstructs graph with/without intermolecular interactions as input, then outputs the probability that indicates if the input structure is acceptable or not. (A) Overall logical steps of the pipeline. (B) Architecture of the GNN network with the gated graph attention mechanism.

the key advantages of our method are the computation and sampling of the molecular surface on-the-fly from the underlying atomic point cloud and a novel efficient geometric convolutional layer. As a result, Dmsif is able to process large collections of proteins in an end-to-end fashion, taking as the sole input the raw 3D coordinates and chemical types of their atoms, eliminating the need for any hand-crafted pre-computed features.

Attention and Gate-Augmented Mechanism

The constructed graphs are used as the input to the GNN. More formally, graphs are the adjacency matrix A^1 and A^2 , and the $x^{in} = \{x_1^{in}, x_2^{in}, \dots, x_N^{in}\}$ with $x \in \mathbb{R}^F$, where F is the dimension of the node feature.

We first explain the attention mechanism of our GNN. With the input graph of x^{in} , the pure graph attention coefficient is defined in Eq.3, which denotes the relative importance between the i -th and the j -th node:

$$e_{ij} = x_i'^T E x_j' + x_j'^T E x_i' \quad (3)$$

where x_i' and x_j' are the transformed feature representations defined by $x_i' = W x_i^{in}$ and $x_j' = W x_j^{in}$. $W, E \in \mathbb{R}^{F \times F}$ are learnable matrices in the GNN. e_{ij} and e_{ji} become identical to satisfy the symmetrical property of the graph by adding $x_i'^T E x_j'$ and $x_j'^T E x_i'$. The coefficient will only be computed for i and j where $A_{ij} > 0$.

Attention coefficients will also be computed for elements

in the adjacency matrices. They are formulated in the following form for the element (i, j) :

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{j \in N_i} \exp(e_{ij})} A_{ij} \quad (4)$$

where a_{ij} is the normalized attention coefficient for the i -th and the j -th node pair, e_{ij} is the symmetrical graph attention coefficient computed in Eq.3, and N_i is the set of neighbors of the i -th node that includes interacting nodes j where $A_{ij} > 0$. The purpose of Eq.4 is to consider both the physical structure of the interaction, A_{ij} , and the normalized attention coefficient, e_{ij} , to define the attention.

Based on the attention mechanism, the new node feature of each node is updated by considering its neighboring nodes, which is a linear combination of the neighboring node features with the final attention coefficient a_{ij} :

$$x_i'' = \sum_{j \in N_i} a_{ij} x_j' \quad (5)$$

Furthermore, the gate mechanism is further applied to update the node feature since it is known to significantly boost the performance of GNN. The basic idea is similar to that of ResNet, where the residual connection from the input helps to avoid information loss, alleviating the gradient collapse problem of the conventional backpropagation. The gated graph attention can be viewed as a linear combination of x_i and x_i'' , as defined in Eq.6:

$$x_i^{out} = c_i x_i + (1 - c_i) x_i'' \quad (6)$$

where $c_i = [D(x_i || x_i'') + b]$, $D \in \mathbb{R}^{2F}$ is a weight vector that is multiplied (dot product) with the vector $x_i || x_i''$, and b is a constant value. Both D and b are learnable parameters and are shared among different nodes. $x_i || x_i''$ denotes the concatenation vector of $x_i || x_i''$.

We refer to attention and gate-augmented mechanism as the gate-augmented graph attention layer (GAT). Then, we can simply denote $x_i^{out} = GAT(x_i^{in}, A)$. The node embedding can be iteratively updated by GAT, which aggregates information from neighboring nodes.

Graph Neural Network Architecture of our model

Using the GAT mechanism described before, we adopted four layers of GAT in GNN-DOVE to process the node embedding information from neighbors and to output the updated node embedding. For the two adjacency matrices A^1 and A^2 , we used a shared GAT. The initial input of the network is atom features. With two matrices, A^1 and A^2 , we have $x_1 = GAT(x^{in}, A^1)$ and $x_2 = GAT(x^{in}, A^2)$. To focus only on the intermolecular interactions within an input protein complex model, we subtracted the embedding of the two graphs as the final node embedding. By subtracting the updated embedding x_1 from x_2 , we can capture the aggregation information that only comes from the intermolecular interactions with other nodes in the protein complex model. Thus, the output node feature is defined as

$$x^{out} = x^2 - x^1 \quad (7)$$

Then, the updated x^{out} will become x^{in} to iteratively augment the information through the three following GAT layers. After the node embeddings were updated by the four GAT layers, the node embedding of the whole graph was

summed up as the entire graph representation, which is considered as the overall intermolecular interaction representation of the protein complex model:

$$x_{graph} = \sum_{k \in G} x_k \quad (8)$$

Finally, FC layers were applied to x_{graph} to classify whether the protein complex model is correct or incorrect. In total, four FC layers were applied. The first layer takes 140 feature values from Eq.8. The three subsequent layers have a dimension of 128. RELU activation functions were used between the FC layers, and a sigmoid function was applied for the last layer to output a probability value.

Training Networks

Since the dataset was highly imbalanced with more incorrect decoys than acceptable ones, we balanced the training data by sampling the same number of acceptable and incorrect decoys in each batch. We sampled the same number of correct and incorrect decoys. To achieve this, a positive (i.e., correct) decoy may be sampled multiple times in one epoch of training.

For training, cross-entropy loss(Goodfellow, Bengio, and Courville 2016) was used as the loss function, and the Adam optimizer was used for parameter optimization. To avoid overfitting, a dropout(Srivastava et al. 2014) of 0.3 was applied for every layer, except the last FC layer. Models were trained for 30 epochs with a batch size of 4. Weights of every layer were initialized using the Glorot uniform(Glorot and Bengio 2010) to have a zero-centered Gaussian distribution, and bias was initialized to 0 for all layers.

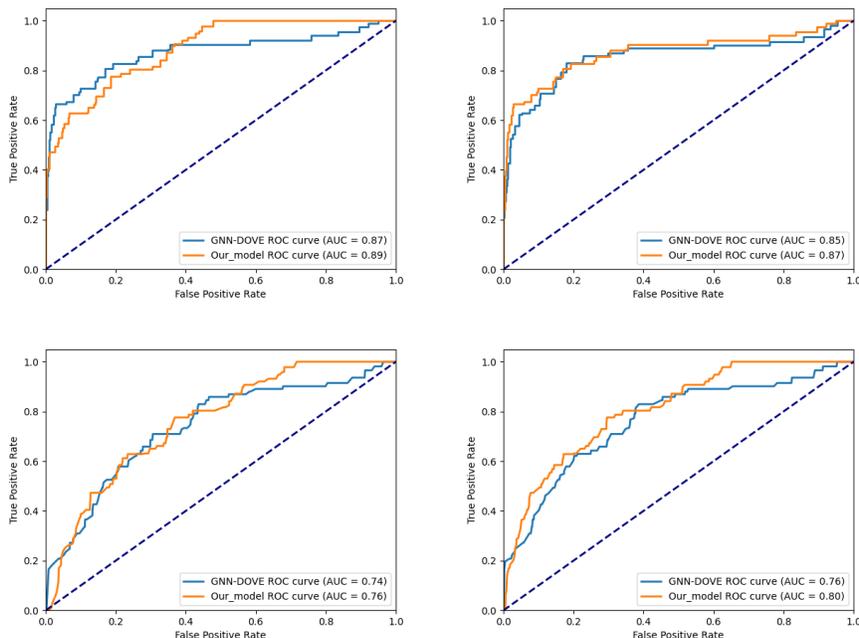


Figure 2: The ROC with indicated AUC and for our model in comparison to GNN-DOVE

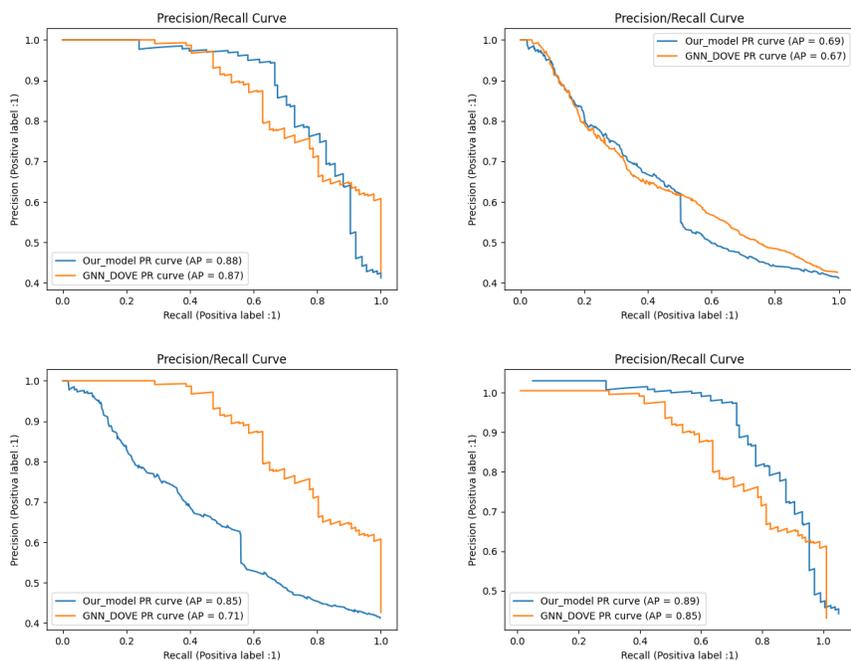


Figure 3: The precision-recall curves for out model vs GNN-DOVE with AP indicated

Table 3: The performance of our model for each fold in the cross validation on the DockGround dataset

Fold	Accuracy	True positive rate	True negative rate
1	93.02%	45.76%	96.50%
2	91.05%	44.45%	95.29%
3	84.71%	39.56%	90.79%
4	92.15%	45.88%	95.09%

First, we performed four-fold cross-validation on the Dockground dataset (Table 1). For fold 1, we found a learning rate of 0.0002 with a weight decay of 0 achieved the highest accuracy on the validation set. We used this parameter combination throughout the other three folds in the cross-validation. The training process generally converged after approximately 30 epochs. In the end, we used the full dataset for training, where 20 percent of the data was selected for validation to check for convergence.

Result

Performance on the Dockground Dataset

We evaluated the performance of our model for each fold in the cross validation on the DockGround dataset. We benchmarked our model against GNN-DOVE (Wang et al. 2020), a recent highly performant deep learning model. The average AUC achieved by the trained model on the test set was 0.83 indicating that the model effectively learned to classify unseen protein-protein docking decoys with structures and sequences differing from those seen in the training set. Compared with our model, GNN-DOVE only achieved by the trained model on the test set was 0.80. More interest-

ingly, when comparing the average precision (AP), our model outperforms GNN-DOVE. GNN-DOVE achieves an AP of 0.77 compared to our AP of 0.83. This indicates that our model typically achieves a greater precision in comparison to the standard Rosetta scoring function. As for some common deep learning evaluation metrics, such as classification accuracy, our model achieves good results on every fold, especially when classifying negative samples. This indicates that our model has a strong ability to classify the negative class samples, but lacks the ability to classify the positive class samples.

Performance on the CAPRI Scoring Dataset

We evaluated the performance of our model and GNN-DOVE on the CAPRI score set. The average AUC achieved by the trained model on the test set was 0.65 indicating that the model effectively learned to classify unseen protein-protein docking decoys with structures and sequences differing from those seen in the training set. Compared with our model, GNN-DOVE only achieved by the trained model on the test set was 0.46 indicating that the model can be considered as unsatisfactory. Unfortunately, our model does not perform well on average accuracy, and GNN-DOVE does

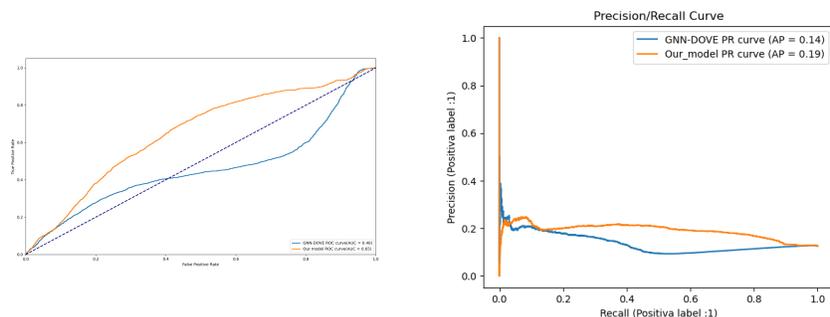


Figure 4: The evaluation metrics calculated on the CAPRI score set will be presented in the figure and table 4. (a) The ROC with indicated AUC and for our model in comparison to GNN-DOVE. (b) The precision-recall curves for our model vs GNN-DOVE with AP indicated

Table 4: The performance of our model and GNN-DOVE on the CAPRI score set

	AUC	AP	Accuracy	True positive rate	True negative rate
GNN-DOVE	0.46	0.14	44.85	75.55	38.22
Our model	0.65	0.19	44.44	62.48	45.51

likewise poorly. This is because there are many negative samples in CAPRI score set, and the prediction accuracy of the two models for the negative class samples is not high.

Discussion

In this work, we developed our model for protein docking decoy selection, which still follows the network model of SOTA (GNN-DOVE) for this task, we introduce the dmasif, which achieved the state of the art in current protein representation work, which can input proteins and output relevant chemical features. It is our innovation to introduce these chemical features into the GNN-DOVE model. Using only part of the training data of GNN-DOVE, we achieve the same performance as GNN-DOVE, and even surpass this work in some metrics. The performance of our model likely would be improved by considering other features such as geometric features, which can be introduced by the dmasif. On the other hand, leveraging different network architectures is also a good approach.

References

- Aderinwale, T.; Christoffer, C. W.; Sarkar, D.; Alnabati, E.; and Kihara, D. 2020. Computational structure modeling for diverse categories of macromolecular interactions. *Current opinion in structural biology*, 64: 1–8.
- Akbal-Delibas, B.; Farhoodi, R.; Pomplun, M.; and Haspel, N. 2016. Accurate refinement of docked protein complexes using evolutionary information and deep learning. *Journal of Bioinformatics and Computational Biology*, 14(03): 1642002.
- Anishchenko, I.; Kundrotas, P. J.; Tuzikov, A. V.; and Vakser, I. A. 2015. Structural templates for comparative protein docking. *Proteins: Structure, Function, and Bioinformatics*, 83(9): 1563–1570.
- Basu, S.; and Wallner, B. 2016. Finding correct protein–protein docking models using ProQDock. *Bioinformatics*, 32(12): 1262–1270.
- Berggård, T.; Linse, S.; and James, P. 2007. Methods for the detection and analysis of protein–protein interactions. *Proteomics*, 7(16): 2833–2842.
- Degiacomi, M. T. 2019. Coupling molecular dynamics and deep learning to mine protein conformational space. *Structure*, 27(6): 1034–1040.
- Fischer, D.; Lin, S. L.; Wolfson, H. L.; and Nussinov, R. 1995. A geometry-based suite of molecular docking processes. *Journal of Molecular Biology*, 248(2): 459–477.
- 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.
- Geng, C.; Jung, Y.; Renaud, N.; Honavar, V.; Bonvin, A. M.; and Xue, L. C. 2020. iScore: a novel graph kernel-based function for scoring protein–protein docking models. *Bioinformatics*, 36(1): 112–121.
- Glorot, X.; and Bengio, Y. 2010. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research.
- Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.
- Gray, J. J.; Moughon, S.; Wang, C.; Schueler-Furman, O.; Kuhlman, B.; Rohl, C. A.; and Baker, D. 2003. Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of molecular biology*, 331(1): 281–299.
- Huang, S.-Y. 2014. Search strategies and evaluation in protein–protein docking: principles, advances and challenges. *Drug discovery today*, 19(8): 1081–1096.

- Huang, S.-Y.; and Zou, X. 2008. An iterative knowledge-based scoring function for protein-protein recognition. *Proteins: Structure, Function, and Bioinformatics*, 72(2): 557–579.
- Jing, B.; Eismann, S.; Suriana, P.; Townshend, R. J.; and Dror, R. 2020. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*.
- Katchalski-Katzir, E.; Shariv, I.; Eisenstein, M.; Friesem, A. A.; Aflalo, C.; and Vakser, I. A. 1992. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, 89(6): 2195–2199.
- Lensink, M. F.; Velankar, S.; Baek, M.; Heo, L.; Seok, C.; and Wodak, S. J. 2018. The challenge of modeling protein assemblies: the CASP12-CAPRI experiment. *Proteins: Structure, Function, and Bioinformatics*, 86: 257–273.
- Lensink, M. F.; and Wodak, S. J. 2014. Score_set: a CAPRI benchmark for scoring protein complexes. *Proteins: Structure, Function, and Bioinformatics*, 82(11): 3163–3169.
- Lim, J.; Ryu, S.; Park, K.; Choe, Y. J.; Ham, J.; and Kim, W. Y. 2019. Predicting drug-target interaction using a novel graph neural network with 3D structure-embedded graph representation. *Journal of chemical information and modeling*, 59(9): 3981–3988.
- Liu, S.; Gao, Y.; and Vakser, I. A. 2008. Dockground protein-protein docking decoy set. *Bioinformatics*, 24(22): 2634–2635.
- Lu, H.; Lu, L.; and Skolnick, J. 2003. Development of unified statistical potentials describing protein-protein interactions. *Biophysical journal*, 84(3): 1895–1901.
- Moal, I. H.; and Bates, P. A. 2010. SwarmDock and the use of normal modes in protein-protein docking. *International journal of molecular sciences*, 11(10): 3623–3648.
- Moal, I. H.; Torchala, M.; Bates, P. A.; and Fernández-Recio, J. 2013. The scoring of poses in protein-protein docking: current capabilities and future directions. *BMC bioinformatics*, 14(1): 1–15.
- Oliwa, T.; and Shen, Y. 2015. cNMA: a framework of encounter complex-based normal mode analysis to model conformational changes in protein interactions. *Bioinformatics*, 31(12): 1151–1160.
- Padhorny, D.; Kazennov, A.; Zerbe, B. S.; Porter, K. A.; Xia, B.; Mottarella, S. E.; Kholodov, Y.; Ritchie, D. W.; Vajda, S.; and Kozakov, D. 2016. Protein-protein docking by fast generalized Fourier transforms on 5D rotational manifolds. *Proceedings of the National Academy of Sciences*, 113(30): E4286–E4293.
- Porter, K. A.; Desta, I.; Kozakov, D.; and Vajda, S. 2019. What method to use for protein-protein docking? *Current Opinion in Structural Biology*, 55: 1–7.
- Scott, D. E.; Bayly, A. R.; Abell, C.; and Skidmore, J. 2016. Small molecules, big targets: drug discovery faces the protein-protein interaction challenge. *Nature Reviews Drug Discovery*, 15(8): 533–550.
- Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.
- Sun, P. D.; Foster, C. E.; and Boyington, J. C. 2004. Overview of protein structural and functional folds. *Current protocols in protein science*, 35(1): 17–1.
- Sunny, S.; and Jayaraj, P. 2022. Protein-protein docking: Past, present, and future. *The protein journal*, 41(1): 1–26.
- Torng, W.; and Altman, R. B. 2019. Graph convolutional neural networks for predicting drug-target interactions. *Journal of chemical information and modeling*, 59(10): 4131–4149.
- Tovchigrechko, A.; and Vakser, I. A. 2005. Development and testing of an automated approach to protein docking. *Proteins: Structure, Function, and Bioinformatics*, 60(2): 296–301.
- Tuncbag, N.; Gursoy, A.; Nussinov, R.; and Keskin, O. 2011. Predicting protein-protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM. *Nature protocols*, 6(9): 1341–1354.
- Typas, A.; and Sourjik, V. 2015. Bacterial protein networks: properties and functions. *Nature Reviews Microbiology*, 13(9): 559–572.
- Venkatraman, V.; Yang, Y. D.; Sael, L.; and Kihara, D. 2009. Protein-protein docking using region-based 3D Zernike descriptors. *BMC bioinformatics*, 10(1): 1–21.
- Vreven, T.; Hwang, H.; and Weng, Z. 2011. Integrating atom-based and residue-based scoring functions for protein-protein docking. *Protein Science*, 20(9): 1576–1586.
- Wang, X.; Terashi, G.; Christoffer, C. W.; Zhu, M.; and Kihara, D. 2020. Protein docking model evaluation by 3D deep convolutional neural networks. *Bioinformatics*, 36(7): 2113–2118.
- Zhang, Y.; and Skolnick, J. 2004. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4): 702–710.
- Zimmermann, M. T.; Leelananda, S. P.; Kloczkowski, A.; and Jernigan, R. L. 2012. Combining statistical potentials with dynamics-based entropies improves selection from protein decoys and docking poses. *The Journal of Physical Chemistry B*, 116(23): 6725–6731.