A Graph Encoder

A protein with n amino acids can be represented as an undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{P})$, where $\mathbb{V} = \{\mathbf{h}_i\}_{i=1,\ldots,n}$ represents the set of amino acids, and $\mathbf{h}_i \in \mathbb{R}^d$ represents the eigenvector of amino acid i. The adjacency matrix $\mathbb{P} \in \mathbb{R}^{n \times n}$ represents the adjacency relationship of the protein graph $\mathbb{G}$. $\mathbb{P}$ is obtained by two steps:

1. Calculate the Euclidean distance between all residue pairs. First, we obtain the coordinates of the side chain center of mass of each residue. There are two ways to obtain them: one is to use the PDB file of the protein, which is calculated based on the real protein structure; the other is to use ESMFold [14] to predict the coordinates. Then we calculate the Euclidean distance between all residue pairs, and get the distance matrix $D \in \mathbb{R}^{n \times n}$.
2. Calculate the adjacency matrix P. We convert this protein distance matrix D into an adjacency matrix P:

$$P_{ij} = \begin{cases} 1, & D_{ij} \leq d_t \\ 0, & D_{ij} > d_t \end{cases} \tag{1}$$

Where d_t=14 Å is a hyper-parameter. It is worth noting that each node in the protein graph $\mathbb{G}$ is self-looping.

The input to the GAT layer is a set of amino acid node features:

$$H = \{h_1, \ldots, h_i, \ldots, h_n\}, \quad h_i \in \mathbb{R}^d.$$

where n is the number of nodes, and d is the number of features in each node. The layer produces a new set of node features $H' = \{h'_1, \ldots, h'_i, \ldots, h'_N\}$ as its output, where $h'_i \in \mathbb{R}^d$.

In Eq.2, $W \in \mathbb{R}^{d \times d}$, a^T are all learnable parameters, and $||$ represents the connection operation, LeakyReLU is a nonlinear activation function. Eq.3 adopts the Softmax function to normalize all e_{ij} in the neighborhood of node i to obtain the attention weight α_{ij}. The node representation is updated using the attention weight α_{ij} as shown in Eq.4, where σ represents the nonlinear activation function. Eq.5 shows the update process of multi-head attention, where W_O represents the learnable linear transformation matrix.

In GAT layers, we adopt the same initial residual connection and identity mapping strategy [3] as used in [22,25] to mitigate the over-smoothing problem.

$$e_{ij} = \mathrm{LeakyReLU}\left(\mathrm{a}^T \left[Wh_i || Wh_j\right]\right) \tag{2}$$

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k \epsilon N_i} \exp\left(e_{ik}\right)} \tag{3}$$

$$h'_i = \sigma\left(\sum_{j \epsilon N_i} \alpha_{ij} W_C h_j\right) \tag{4}$$

$$h'_i = \mathrm{Concat}\left(h_i^{\prime(1)}, h_i^{\prime(2)}, \ldots, h_i^{\prime(h)}\right) W_O \tag{5}$$

B Amino acid features

The details of features are shown in Table 1, including their dimensions and the software used for each feature extraction.

Table 1: The information of amino acid features used in COAT-GNN

Feature Name	**Program**	**Dimension**
PSSM	PSI-Blast	20
HHM	HHblits	20
DSSP	DSSP	14
Pseudo-position embedding	-	1
Atom Feature	-	11
Complete feature		66

B.1 Position-specific scoring matrix (PSSM)

PSSM is widely used in many related protein interaction problems and exhibits good performance[25,13,9,6]. In PSSM, the value computed by aligning an input sequence with specific protein database represents the evolutionary conservation (ECO) of each amino acid position. Here, the PSI-Blast v2.15.0 with default parameters was used to compute the PSSM matrices[1], in which the expectation value (E-value) was set to 0.001 and the number of iteration was set to 3. And the PSI-Blast performs multiple Sequence alignment (MSA) on each input protein sequence against the UniRef90 database.

B.2 Hidden markov models matrix (HHM)

The HHblits software is another sequence alignment tool based on the pairwise alignment of hidden Markov models, which is a faster and more sensitive than PSI-Blast software[19,18]. Therefore, the HHM was added to complementarily get the view of the evolutionary information of protein sequence as much as possible. In this study, each protein sequence was aligned with Uniclust30 database by hhblits v3.3.0 with default parameters, in which the E-value cutoff was set 0.001 and number of iterations was set to 2.

The values of PSSM and HHM are normalized to scores distributed between 0 and 1 by the Equation:

$$V^* = \frac{V - V_{\min}}{V_{\max} - V_{\min}}$$

Where V^* represents the normalized value, V is the original value, $V_{\min}$ and $V_{\max}$ represent the minimum and maximum value of this type of feature in the training set.

Table 2: The information of datasets

Dataset	Structure Source	Protein quantity	Interacting residues(%)	Non-interacting residues(%)	of Interacting residues (%)
PPBS					
Train	PDB	12,691	577,903	2,499,857	18.78
Validation_70	PDB	411	21,946	84,878	20.54
Validation_homology	PDB	1,451	64,301	272,840	19.07
Validation_topology	PDB	589	31,678	119,934	20.89
Validation_none	PDB	747	38,393	162,677	19.09
Test_70 (T_70)	PDB	547	28,654	105,921	21.29
Test_homology (T_homo)	PDB	1,481	63,423	285,534	18.18
Test_topology (T_topo)	PDB	914	48,293	198,972	19.53
Test_none (T_none)	PDB	1,067	48,149	229,656	17.33
GraphSet					
Train	PDB	334	10,336	55,872	15.61
Test_60 (T_60)	PDB	60	2,075	11,069	15.79
Test_287 (T_287)	PDB	287	8,566	51,810	14.19
BCE					
Train	PDB	2,057	79,846	552,089	12.64
Validation	PDB	919	37,336	258,737	12.61
Test (T_BCE)	PDB	484	15,483	105,418	12.81
DELPHI_Set					
Train	ESMFOLD	9,358	395,069	2,762,366	12.51
Test (T_355)	ESMFOLD	355	11,467	84,473	11.95

Table 3: The selected hyper-parameters in COAT-GNN across 4 benchmark datasets.

Dataset	α	β	L
PPBS	1.0	0.5	4
GraphSet	1.0	0.3	5
DELPHI_SET	1.0	0.3	4
BCE	1.0	0.3	4

B.3 Define secondary structure of proteins (DSSP)

The DSSP matrix contains secondary structure information of protein, which is calculated by DSSP algorithm tool based on ESMFold-predicted structures or crystal structures obtained from the PDB dataset[10]. The process of DSSP matrix with shape of $N_v \times 14$ based on DSSP profile generated by DSSP algorithm is same as the method in the DeepProSite for the protein sequence which has N_v residues[6]. The first dimension is represented by relative solvent accessibility (RSA) for residue which is normalized from the maximal solvent accessible surface area for specific amino acid. Then the one-hot of eight secondary structure states were incorporated in DSSP matrix (nine dimensions, missing residue is represented as 'unkown'). The sine and cosine values of the protein backbone torsion angels Φ and Ψ are combined to DSSP matrix.

B.4 Pseudo-position embedding feature

The pseudo-position feature of residue reflects the relative position information of each residue to reference residue, which is conducive to improving the spatial perception ability of the model. Due to the flexibility of side-chain of amino acid, the coordinate of the geometric center of side-chain (SC) was used to represent the residue pseudo-position. The Pseudo-position matrix with the shape of $N_v \times 3$ can be generated for the protein that contains N_v residues, in which each residue is represented by the three-dimensional coordinate of the geometric center of side-chain. The first residue of protein sequence was regarded as reference residue and its three-dimensional coordinate was set as reference position P_0, then the pseudo-position embedding PE_i of each residue was computed by Equation:

$$PE_i = \frac{1}{r}|\overrightarrow{P_0P_i}|$$

where $|\overrightarrow{P_0P_i}|$ means the Euclidean distance between residue i and refernce residue, r is a hyper-parameter and was set to 1 in this study. Then the pseudo-position embedding feature matrix with the shape of $N_v \times 1$ can be obtained for each protein sequence. Additionally, the centroid coordinates of amino acid side chains are retained to characterize their spatial positions.

B.5 Atom feature

Atom features of protein have shown significant advantages in improving models for protein-related problems[8,25]. Thus, the nine features were extracted from the PDB files: whether it is carbon atom in benzene ring, atomic aromatic property, atom solvent accessible area, atom mass, whether it is in sidechain, electronic charge, the number of hydrogen atoms bonded to it, whether it is in a ring and its van der Waals radius. The atomic solvent accessible area is calculated by Pymol software[4], and the dot_solvent and the dot_density was set to 1 and 4, respectively. Since each amino acid residue has a different number of atoms, the average of the nine atomic features within each residue is taken as the final atomic feature of each residue, as shown in Equation:

$$f_i = \frac{1}{N_A}\sum_{j=1}^{j=N_A} E_{i,j}$$

Where the $\left\{E_{i,j}\right\}_{i=1,\ldots 9:j=1,\ldots N_A}$ denotes the feature i of atom j in a residue, N_A is the total number of atoms in a residue, f_i represents the feature i of the residue. To comprehensively characterize amino acid interface properties, we calculate both the sum and average of atomic solvent-accessible surface area per amino acid. Finally, the eleven-dimensional atomic feature is obtained for each residue.

C Experiment Details

C.1 Datasets and processing

In this study, the performance of our proposed COAT-GNN is evaluated on four widely recognized datasets (table 2) commonly employed in protein-protein binding site prediction.

PPBS Datasets: PPBS datasets, originally developed by ScanNet [20] using the DockGround database [11], treat each unique PDB chain involved in one or more interfaces as individual examples. To evaluate model generalization across different homology levels, the datasets enforce a 90% maximum sequence identity between validation/test and training sets, with validation and test examples divided into four progressively challenging subgroups: (1) Val/Test 70: sharing at least 70% sequence identity with at least one training set protein, (2) Val/Test Homology: with≤70% sequence identity to any training protein but belonging to the same superfamily, (3) Val/Test Topology : no superfamily-level homology but sharing similar topology with at least one training protein, and (4) Val/Test None: no homology at the superfamily or topology levels. Proteins with over one thousand amino acids are excluded from each subgroup in all experiments.

GraphSet datasets: We adapted datasets from AGAT [25], including the training set GraphSet-train and test sets (test_60, test_287), originally constructed by GraphPPIS [22] from the PDB dataset [2] with <25% sequence similarity. PPI sites were defined as surface residues showing $\geq$ 1.0 Å^2 solvent accessibility reduction upon binding. In our experiments, we performed 5-fold cross-validation on GraphSet-train and evaluated model generalization on test_60 and test_287.

DELPHI_SET datasets: DELPHI datasets [12] consist of Train9982 and Test355. Test355 was derived from BioLip [21], with interaction sites defined as residues containing atoms from different chains within 0.5 Å plus the sum of their van der Waals radii. Train9982 was constructed from a large-scale dataset [24], processed using PSI-CD-HIT [7] at 25% sequence identity to ensure diversity. However, as Train9982 contains only sequence information, it cannot be directly used for training structure-based PPI prediction methods. To address this limitation, we utilized the ESMFOLD [14] to generate protein structures in DELPHI datasets. Proteins with over one thousand amino acids are excluded from each subgroup in all experiments. We conducte 5-fold cross-validation on Train9982, and assessed model generalization using Test355.

BCE datasets: The BCE dataset was also constructed by ScanNet [20] using antigen structures from SabDab[5], with sequences clustered at 70% identity and divided into five cross-validation folds. Binding sites were defined as residues containing heavy atoms within 4 Å of another chain in the biological assembly, with sites from multiple assemblies combined through sequence alignment (details on binding site aggregation across multiple assemblies are shown in ScanNet). We use only proteins shorter than one thousand amino acids in all experiments, and employee folds 1-3 for training, fold 4 for validation, and fold 5 (T_BCE) for testing.

C.2 Environment and hyper-parameters

Training setup. The hyper-parameters for all the methods are selected based on the validation splits (available in PPBS and BCE), or based on five-fold cross-validation on the training splits when a validation set is unavailable (e.g., GraphSet and DELPHI_Set) following [22,25,23], see Appendix C.2 for more details on experimental settings. For our method, we use AdamW optimizer with an initial learning rate of 1×10^{-3}, weight decay 1×10^{-4} and 100 training epochs. The learning rate is scheduled to decay by a factor of 0.5 if the performance do not improve for 6 consecutive epochs on the validation set, but would be lower-bounded at 1×10^{-5}. A dropout is used with probability 20%. The bandwidth of the Gaussian kernel σ in Eq.3 is 4 and the node embedding dimension d is 256. Following [22,25,8], we construct the residue-level graph using a distance cutoff of 14 Å in the Graph Encoder layer and the number of GAT layers is 4. We tune three hyper-parameters based on the average of AUPRC and AUROC on the validation sets, i.e., the ratio parameter α in Eq.8 and β in Eq.6 both in $\{0.1, 0.3, 0.5, 0.7, 1.0\}$, and the number of COAT-GNN layers L in $\{1, 2, 3, 4, 5, 6\}$.

Running environment. The experiments are conducted on a machine equipped with an Intel Core i9-13900K CPU, 128 GB of RAM, and a single NVIDIA A6000 GPU. Our method is implemented on PyTorch 1.10.2 and Python 3.6.

Hyper-parameters setting. Table 3 summarizes the selected hyperparameter values for α (attribute update-ratio in Eq.8), β (topology update-ratio in Eq.6), and the number of COAT layers L across different datasets as based on the validation data. It can be observed that the same value of $\alpha = 1.0$ is consistently used, and only minor adjustments to β and L are made across datasets. This consistency indicates that our model is relatively insensitive to hyperparameter variations, demonstrating strong robustness and general applicability across diverse data conditions.

The choice of the hyper-parameters (or their tuning) in the competing models, as well as the training details are listed below:

AGF[1]. Following the setting of AGF [8]: for the model parameters, the number of basic modules is 8, 10 or 12, and the number of neurons of a single hidden layer in the model is 256. Additionally, the number of neurons in the middle-hidden layer in the FFN is 516. The head number of MSA is 6. For the learning rate, a decreasing mechanism is used, and the initial learning rate is 0.001 or 0.0005. If the AUPRC indicator on the verification set do not increase in 10 iterations, the learning rate is reduced to 0.6 times and set the minimum learning rate is 10-6. To prevent overfitting, the probability of dropout is 0.1/0.2. Finally, the number of epochs is 100.

Ensem[2]. Following the experiment setting in Ensem [17]: the number of encoder layers is 3, the dimension of hidden features of the gated convolutional network is selected from [128, 256]; the kernel size of Conv1D is selected from [5, 7, 9]; the learning rate is 0.001 or 0.0005, the dropout is 0.1/0.2 and the number of epochs is 100.

[1] https://github.com/fxh1001/AGF-PPIS
[2] https://github.com/idrblab/EnsemPPIS

DeepProSite[3]. Following the setup mentioned by DeepProsite [6]: the number of k-nearest neighbour is selected from [20, 30], the number of geometric edge features is 16, the dimensions of hidden embeddings is selected from [64,128,256], the number of encoder layers is chosen from [4,6,8]; the dropout is 0.1/0.2 and the learning rate is 0.001 or 0.0005.

ACNN[4]. Following the setup mentioned by ACNN [15]: the input channel and out channel of convolution & polling layer are 1 and 64, respectively. Three kernel sizes (3, 5 and 7) are used, the first fully connected layer has 512 nodes and the second fully connected layer has 256 nodes; the learning rate is 0.001/0.0005 and the dropout is 0.1/0.2.

GraphPPIS[5]. Following the experiment setting in GraphPPIS [22]: an 8 (10 or 12) layers GraphPPIS framework with 256 hidden units are utilized; $\alpha = 0.7$, $\gamma = 1.5$, the learning rate is 0.001 or 0.0005, weight decay of 0, the dropout rate is set to 0.1/0.2 to avoid overfitting and the cutoff used to transform protein distance maps into adjacency matrices is set to 14 Å.

GHGPR[6]. Following the setup mentioned by GHGPR [23]: the dimension of hidden features is 256, the feature extraction module consisted of a GPR component and 5/8/10 GraphHeat-ESGRET blocks. It had been trained for a total of 100 epochs. For optimization, Adaptive Moment Estimation (Adam) is used as the optimizer with a learning rate of 0.001/0.0005, a weight attenuation coefficient of 5e-5, and a penalty term of 1e-7; the dropout is 0.1/0.2; the gradient descent step scale of the ContraNorm function is 0.55 and the temperate coefficient is 1.

AGAT[7]. Following the experiment setting in AGAT [25]: the distance threshold of converting the residues distance matrix into the adjacency matrix is 14 Å; the number of AGAT layers is 8/10/12; and the length of the node embedding is 256; the parameters involved in the initial residual and identity mapping equations are set to $\alpha = 0.7$, $\gamma = 1.5$; the learning rate of training is 0.001/0.0005 and the probability of dropout is 0.1 or 0.2. The cross-entropy loss function and Adam optimizer are used to optimize the model and 100 epochs are included in each training.

EGRET[8]. According to the experimental settings given by EGRET [16]: the number of k-nearest neighbour is selected from [20,30], the length of the node embedding is 256, and the number of EGRET layer is 8/10/12; Adam is used as the optimizer with a learning rate of 0.001 (0.0005), a weight attenuation coefficient of 5e-4; the training epochs is 150 and the probability of dropout is 0.1/0.2.

[3] https://github.com/WeiLab-Biology/DeepProSite
[4] https://github.com/biolushuai/attention-based-CNNsfor-PPIs-prediction
[5] https://github.com/biomed-AI/GraphPPIS
[6] https://github.com/dldxzx/GHGPR-PPIS
[7] https://github.com/AILBC/AGAT-PPIS
[8] https://github.com/Sazan-Mahbub/EGRET

D Standard Deviation of Baselines and Experiment Statistical Significance

Table 4: Standard deviation of the AUPRC and AUROC for the competing methods

Dataset	Test	Metric	Transformer				GNN			
			AGF	Ensem	ProSite	ACNN	GraphPPIS	GHGPR	AGAT	EGRET
PPBS	T_70	auprc	0.004	0.006	0.005	0.007	0.005	0.004	0.005	0.006
		auroc	0.002	0.004	0.003	0.004	0.004	0.002	0.003	0.003
	T_homo	auprc	0.005	0.006	0.004	0.006	0.004	0.004	0.003	0.005
		auroc	0.002	0.004	0.003	0.003	0.002	0.003	0.002	0.003
	T_topo	auprc	0.008	0.008	0.007	0.009	0.006	0.007	0.006	0.009
		auroc	0.004	0.005	0.005	0.006	0.004	0.003	0.003	0.004
	T_none	auprc	0.005	0.006	0.005	0.007	0.004	0.005	0.004	0.005
		auroc	0.003	0.004	0.004	0.005	0.003	0.002	0.003	0.002
Graph Set	T_60	auprc	0.006	0.007	0.006	0.005	0.005	0.004	0.006	0.005
		auroc	0.003	0.006	0.005	0.004	0.003	0.005	0.007	0.003
	T_287	auprc	0.005	0.006	0.004	0.006	0.004	0.005	0.005	0.005
		auroc	0.003	0.005	0.004	0.004	0.003	0.004	0.002	0.003
DELPHI_Set	T_355	auprc	0.004	0.005	0.005	0.004	0.003	0.001	0.003	0.004
		auroc	0.002	0.003	0.004	0.003	0.004	0.003	0.004	0.003
BCE	T_BCE	auprc	0.008	0.008	0.009	0.007	0.006	0.007	0.004	0.006
		auroc	0.006	0.007	0.005	0.006	0.005	0.006	0.005	0.007

Table 4 presents the standard deviations of the baseline models, all the results are reported based on three different random seeds.

Table 5: Statistical significance comparison using t-test across all datasets.

Dataset	**test**	**metric**	**COAT-GNN**	**Best Baseline**	**t-value**	**p-value**	**95% CI**
PPBS	T_70	auprc	0.752 ± 0.004	0.717 ± 0.005	12.222	<0.001	[0.028, 0.042]
		auroc	0.907 ± 0.002	0.890 ± 0.003	10.542	<0.001	[0.013, 0.021]
	T_homo	auprc	0.740 ± 0.003	0.707 ± 0.005	12.654	<0.001	[0.027, 0.039]
		auroc	0.913 ± 0.001	0.899 ± 0.002	14.000	<0.001	[0.012, 0.016]
	T_topo	auprc	0.770 ± 0.008	0.748 ± 0.008	4.348	0.002	[0.010, 0.034]
		auroc	0.922 ± 0.003	0.914 ± 0.003	4.216	0.003	[0.004, 0.012]
	T_none	auprc	0.645 ± 0.003	0.618 ± 0.005	10.354	<0.001	[0.021, 0.033]
		auroc	0.879 ± 0.001	0.871 ± 0.004	4.339	0.009	[0.003, 0.013]
Graph Set	T_60	auprc	0.614 ± 0.002	0.599 ± 0.006	5.303	0.003	[0.008, 0.022]
		auroc	0.879 ± 0.002	0.870 ± 0.003	5.582	0.001	[0.005, 0.013]
	T_287	auprc	0.593 ± 0.008	0.572 ± 0.005	4.977	0.002	[0.011, 0.031]
		auroc	0.887 ± 0.002	0.874 ± 0.002	10.277	<0.001	[0.010, 0.016]
DELPHI_Set	T_355	auprc	0.578 ± 0.003	0.565 ± 0.005	4.985	0.002	[0.007, 0.019]
		auroc	0.893 ± 0.002	0.880 ± 0.004	6.500	0.001	[0.008, 0.018]
BCE	T_BCE	auprc	0.220 ± 0.004	0.211 ± 0.007	2.496	0.045	[0.000, 0.018]
		auroc	0.705 ± 0.005	0.687 ± 0.006	5.153	0.001	[0.010, 0.026]

Table 5 reports the results of the t-test conducted between COAT and the strongest baseline on each dataset, demonstrating the statistical significance of COAT's improvements.

E Computational Complexity

The core computational complexity of COAT-GNN is comparable to that of existing GNN and Transformer-based models. Similar to Graph Transformers, COAT-GNN performs message passing and attention-based aggregation over residue-level embeddings, with per-layer operations scaling approximately as $O(n^2 d)$, where n is the number of residues, d is the feature dimension.

The main novelty of COAT-GNN lies in the dynamic update of residue positions, which introduces two additional computations:

(1) **Spatial attention via Gaussian kernel (Eq.3)**: This step evaluates spatial proximity using a radial basis function over Euclidean distances between residues. The Gaussian computation is mathematically simple and numerically efficient—it involves only squared distance calculation and exponentiation—making it significantly cheaper than typical dot-product attention in Transformers.

(1) **Force-inspired coordinate updates (Eq.5–6)**: The positional refinement is based on simple weighted averaging of local coordinate offsets, using the same attention matrix as in the feature update. These updates are non-iterative, fully differentiable, and efficiently parallelizable on GPUs. They do not require backtracking, optimization loops, or matrix inversion.

Together, these two modules constitute the topology optimization pathway in COAT-GNN. While they are not present in conventional GNNs or Transformers, they incur only marginal overhead. In practice, they contribute a small fraction of total computation, yet offer substantial representational benefits by enabling conformational adaptation during learning.

References

1. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic acids research **25**(17), 3389–3402 (1997)
2. Burley, S.K., Berman, H.M., Kleywegt, G.J., Markley, J.L., Nakamura, H., Velankar, S.: Protein data bank (pdb): the single global macromolecular structure archive. Protein crystallography: methods and protocols pp. 627–641 (2017)
3. Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y.: Simple and deep graph convolutional networks. In: International conference on machine learning. pp. 1725–1735. PMLR (2020)
4. DeLano, W.L., et al.: Pymol: An open-source molecular graphics tool. CCP4 Newsl. Protein Crystallogr **40**(1), 82–92 (2002)
5. Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., Deane, C.M.: Sabdab: the structural antibody database. Nucleic acids research **42**(D1), D1140–D1146 (2014)
6. Fang, Y., Jiang, Y., Wei, L., Ma, Q., Ren, Z., Yuan, Q., Wei, D.Q.: Deepprosite: structure-aware protein binding site prediction using esmfold and pretrained language model. Bioinformatics **39**(12), btad718 (2023)
7. Fu, L., Niu, B., Zhu, Z., Wu, S., Li, W.: Cd-hit: accelerated for clustering the next-generation sequencing data. Bioinformatics **28**(23), 3150–3152 (2012)

8. Fu, X., Yuan, Y., Qiu, H., Suo, H., Song, Y., Li, A., Zhang, Y., Xiao, C., Li, Y., Dou, L., et al.: Agf-ppis: a protein–protein interaction site predictor based on an attention mechanism and graph convolutional networks. Methods **222**, 142–151 (2024)
9. Hu, J., Zeng, W.W., Jia, N.X., Arif, M., Yu, D.J., Zhang, G.J.: Improving dna-binding protein prediction using three-part sequence-order feature extraction and a deep neural network algorithm. Journal of Chemical Information and Modeling **63**(3), 1044–1057 (2023)
10. Kabsch, W., Sander, C.: Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers: Original Research on Biomolecules **22**(12), 2577–2637 (1983)
11. Kundrotas, P.J., Anishchenko, I., Dauzhenka, T., Kotthoff, I., Mnevets, D., Copeland, M.M., Vakser, I.A.: Dockground: a comprehensive data resource for modeling of protein complexes. Protein Science **27**(1), 172–181 (2018)
12. Li, Y., Golding, G.B., Ilie, L.: Delphi: accurate deep ensemble model for protein interaction sites prediction. Bioinformatics **37**(7), 896–904 (2021)
13. Lin, P., Yan, Y., Tao, H., Huang, S.Y.: Deep transfer learning for inter-chain contact predictions of transmembrane protein complexes. Nature Communications **14**(1), 4935 (2023)
14. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al.: Evolutionary-scale prediction of atomic-level protein structure with a language model. Science **379**(6637), 1123–1130 (2023)
15. Lu, S., Li, Y., Nan, X., Zhang, S.: Attention-based convolutional neural networks for protein-protein interaction site prediction. In: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 141–144. IEEE (2021)
16. Mahbub, S., Bayzid, M.S.: Egret: edge aggregated graph attention networks and transfer learning improve protein–protein interaction site prediction. Briefings in Bioinformatics **23**(2), bbab578 (2022)
17. Mou, M., Pan, Z., Zhou, Z., Zheng, L., Zhang, H., Shi, S., Li, F., Sun, X., Zhu, F.: A transformer-based ensemble framework for the prediction of protein–protein interaction sites. Research **6**, 0240 (2023)
18. Remmert, M., Biegert, A., Hauser, A., Söding, J.: Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. Nature methods **9**(2), 173–175 (2012)
19. Steinegger, M., Meier, M., Mirdita, M., Vöhringer, H., Haunsberger, S.J., Söding, J.: Hh-suite3 for fast remote homology detection and deep protein annotation. BMC bioinformatics **20**, 1–15 (2019)
20. Tubiana, J., Schneidman-Duhovny, D., Wolfson, H.J.: Scannet: an interpretable geometric deep learning model for structure-based protein binding site prediction. Nature Methods **19**(6), 730–739 (2022)
21. Yang, J., Roy, A., Zhang, Y.: Biolip: a semi-manually curated database for biologically relevant ligand–protein interactions. Nucleic acids research **41**(D1), D1096–D1103 (2012)
22. Yuan, Q., Chen, J., Zhao, H., Zhou, Y., Yang, Y.: Structure-aware protein–protein interaction site prediction using deep graph convolutional network. Bioinformatics **38**(1), 125–132 (2022)
23. Zeng, X., Meng, F.F., Li, X., Zhong, K.Y., Jiang, B., Li, Y.: Ghgpr-ppis: a graph convolutional network for identifying protein-protein interaction site using heat kernel with generalized pagerank techniques and edge self-attention feature processing block. Computers in Biology and Medicine **168**, 107683 (2024)

24. Zhang, J., Ma, Z., Kurgan, L.: Comprehensive review and empirical analysis of hallmarks of dna-, rna-and protein-binding residues in protein chains. Briefings in bioinformatics **20**(4), 1250–1268 (2019)
25. Zhou, Y., Jiang, Y., Yang, Y.: Agat-ppis: a novel protein–protein interaction site predictor based on augmented graph attention network with initial residual and identity mapping. Briefings in bioinformatics **24**(3), bbad122 (2023)