RESEARCH **Open Access**

CurvAGN: Curvature-based Adaptive Graph Neural Networks for Predicting Protein-Ligand Binding Affinity

Jianqiu Wu[1], Hongyang Chen[1*], Minhao Cheng[2] and Haoyi Xiong[3]

*Correspondence:
dr.h.chen@ieee.org

[1] Research Center for Graph Computing, Zhejiang Lab, Yuhang, Hangzhou 311121, Zhejiang, China
[2] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Jiulongwan, Hongkong 999077, China
[3] Big Data Lab, Baidu Inc., Haidian, Beijing 100080, China

Abstract

Accurately predicting the binding affinity between proteins and ligands is crucial for drug discovery. Recent advances in graph neural networks (GNNs) have made significant progress in learning representations of protein-ligand complexes to estimate binding affinities. To improve the performance of GNNs, there frequently needs to look into protein-ligand complexes from geometric perspectives. While the "off-the-shelf" GNNs could incorporate some basic geometric structures of molecules, such as distances and angles, through modeling the complexes as homophilic graphs, these solutions seldom take into account the higher-level geometric attributes like curvatures and homology, and also heterophilic interactions.To address these limitations, we introduce the Curvature-based Adaptive Graph Neural Network (CurvAGN). This GNN comprises two components: a curvature block and an adaptive attention guided neural block (AGN). The curvature block encodes multiscale curvature informaton, then the AGN, based on an adaptive graph attention mechanism, incorporates geometry structure including angle, distance, and multiscale curvature, long-range molecular interactions, and heterophily of the graph into the protein-ligand complex representation. We demonstrate the superiority of our proposed model through experiments conducted on the PDBbind-V2016 core dataset.

Keywords: Multiscale curvature, Adaptive graph attention mechanism, Heterophily, Protein-ligand bingding affinity

Introduction

Protein-ligand binding affinity prediction is a critical step in drug discovery [1]. It allows researchers to identify potential drug candidates and optimize their properties before conducting expensive and time-consuming experiments. The increasing availability of three-dimensional (3D) structural protein data provides a new paradigm for structure-based drug discovery and 3D structural information has been proven to facilitate drug design [2]. Various computational methods have been developed to learn 3D structure information from a protein-ligand complex. These methods range from molecular docking [3–6] to more sophisticated machine learning [1, 7, 8] and deep learning approaches [9].

Docking methods have been widely adopted with a scoring function for binding affinity prediction, but their accuracy also limits the potential applications of docking methods [3, 4]. Traditional machine learning algorithms [7, 8] together with handcrafted features could sometimes deliver decent performance, but they are difficult to scale up due to cost of extensive feature engineering. To model 3D spatial structure, many deep learning approaches [10–12] divide the complex into 3D grid data and apply 3D convolutional neural works(3D CNNs) to extract useful features. These approaches have demonstrated better performance in predicting binding affinity than traditional machine learning-based models. However, the sparsity distribution of atoms in the complex can result in inefficient computations when using a 3D rectangular grid representation [13].

Modeling a protein-ligand complex as a graph where nodes correspond to atoms is a natural and effective approach [14, 15]. Graph neural networks (GNNs) have demonstrated remarkable capabilities in expressing graph structures, and researchers have made considerable efforts to incorporate spatial information to enhance its expression ability. Spatial Graph Convolutional Networks [16, 17] utilize 3D coordinates to model the structure of complexes. However, the output of coordinate-based models can be negatively impacted by rotations of the coordinates. This limitation is addressed by distance-aware GNNs [13, 18], which only take distance into account. But these models may not suffice to accurately model 3D structures for binding affinity predictions. Directional message passing-based GNNs [2, 19] have been proposed to address this limitation. These models incorporate angle and distance information, which has been shown to be crucial in empirical potentials for molecules [20]. While these models offer improved prediction performance, their accuracy have a great potential to be further improved. Since the protein-ligand binding affinity is determined by its absolute binding free energy [21], which is primarily specified by curvature [22], incorporating curvature information into the graph representation is necessary to enhance prediction accuracy.The concept of curvature is closely related to the geometry of a manifold, and some efforts have been made to generalize curvatures for a graph [23, 24]. Based on this generalization, two different curvature-based graph neural networks [25, 26] have been proposed, and they perform well on baseline datasets. Biomolecules often exhibit hierarchical and multiscale structures, which require a multiscale representation to accurately characterize their interactions [27]. It implies multiscale curvature for graph is more suitable. However, incorporating multiscale curvature into GNNs for predicting binding affinity remains an open research question.

Moreover, many studies have recognized the heterogeneity of protein-ligand complex graphs and endeavored to incorporate this heterogeneity into their graph neural networks [2, 28, 29]. Nevertheless, it is often disregarded that the graph is not strictly homophilic, as neighboring nodes may not be similar. Graph neural networks based on the homophily assumption cannot effectively learn heterophily, which is the property where linked nodes have dissimilar features [30, 31]. Therefore, previous studies on binding affinity have failed to capture heterophily.

To address above challenges, we propose a novel Curvature-based Adaptive Graph Neural Network (CurvAGN) for predicting protein-ligand binding affinity. The CurvAGN comprises a curvature block and an adaptive attention guided neural block (AGN). The curvature block assigns edge attributes to include multiscale curvature, and AGN is inspired by SIGN [2] and consists of two parts. The first part, called the polar-inspired adaptive graph attention block (PAGA), uses an adaptive graph attention mechanism [32] to model the 3D spatial structure of the protein-ligand complex by incorporating distance, angle, and curvature information. The adaptive attention mechanism addresses the heterophily in the protein-ligand complex graph. The second part is the pooling block, which is described in [2] and includes the pairwise interactive pooling (PiPool) for leveraging long-range interactions and the output pooling layer for predicting the protein-ligand binding affinity.

Our work makes three main contributions:

- We propose the curvature block that utilizes multiscale curvature to encode edge attributes of biomolecule graphs, effectively capturing the multiscale structure of these biomolecules.
- We find the distance-based complex interaction graph is a heterophilic graph, and further propose the adaptive attention guided neural model (AGN) to capture the heterophily and geometric structure of angles and distances, and and long-range molecular interactions.
- We combine the curvature-based graph neural network and AGN to propose the Curvature-based Adaptive Graph Neural Network (CurvAGN).
- We apply CurcAGN to predicting the protein-ligand binding affinity. We train and validate our model on the publicly available standard PDBbind-v2016 dataset, and show that it outperforms SIGN [2] by 7.5% in RMSE and 9.4% in MAE.

Related work

3D structure GNNs for binding affinity prediction

3D structural GNNs have been used to integrate the 3D structure of protein-ligand complexes into high-level representations, thereby improving the accuracy of binding affinity prediction. Atom coordinate-based GNNs [17] use atomic coordinates directly as node attributes, but they often fail to recognize the same protein-ligand complex due to coordinate variations in different coordinate systems. Distance-based GNNs [13, 33–35] overcome this deficiency by utilizing atomic distances. Angle and distance-based GNNs [2, 19] can enrich geometric information and enhance complex modeling capabilities.

Ricci curvature for graphs

Ricci curvature is a geometric object that measures the curvature of a Riemannian manifold [36, 37]. Intuitively, if the Ricci curvature is positive, the manifold curves more like a sphere, while negative Ricci curvature results in a more saddle-like curve. In recent years, there has been growing interest in the study of graph curvature, which is a discrete analogue of Ricci curvature. There are two main types of graph curvature: Ollivier Ricci curvature (ORC) and Forman Ricci curvature (FRC). ORC is based on optimal transport theory and captures the geometric properties of a graph [23, 38–43], while FRC is based on the graph Laplacian and captures the algebraic topological properties of a graph [24, 44]. In general, ORC is a more recent and sophisticated measure of curvature than FRC. However, FRC is more widely used because it is easier to compute.

Persistent graph-curvature-descriptors

Xia et al. propose a persistent graph curvature descriptor to characterize molecular features based on the observation that biomolecules have a hierarchical and multiscale structure [27, 43]. They first filter the edges of the graph by length to remove short edges that are less relevant to the hierarchical structure, and then construct a sequence of subgraphs, where each subgraph is a subset of the next one. They then define a permutation-invariant descriptor function for each subgraph that is related to curvature. This function is designed to be invariant to the order in which the nodes are arranged, so that it can be used to characterize the molecular features of the graph regardless of how the graph is represented. Finally, they arrange the descriptors of each subgraph in sequence, to form the persistent graph curvature descriptor.

Heterophily-based GNNs

Heterophilic graphs refer to graphs where linked nodes exhibit heterophily, meaning that they have dissimilar features and different class labels [29]. Many real-world graphs, such as transaction networks [45], exhibit heterophily. Recent studies have shown that GNNs do not perform well on heterophilic graphs [46–49]. This is because GNNs are typically designed to learn from homophilic graphs, where linked nodes have similar features and class labels. To address this issue, several GNN designs have been proposed that are specifically tailored for heterophilic graphs. These designs include MixHop [50], MM-DAN [51], BeyondGNN [32], AdaGNN [52], Beyond-GCN [53], and Geom-GCN [54].

Persistent curvature descriptors have been shown to be effective in representing protein-ligand complexes, but they rely on prior knowledge. To overcome this limitation, we developed a multiscale curvature graph neural network that incorporates the multiscale curvature of edges as edge attributes. In addition to the curvature information, the interactions between molecules play a critical role in binding affinity. When modeling a protein-ligand complex as a graph, protein atoms and ligand atoms are connected based on distance, but short distance does not necessarily mean similar features. This leads to the graph not having strict homophily. To capture this important feature, it is natural to utilize heterophily-based models. However, to the best of our knowledge, no heterophily-based GNNs have been used for modeling this complex yet. Therefore, we propose incorporating the adaptive graph attention mechanism [32] into our network.

Preliminaries

In this section, we introduce some key definitions that will be used in our model and formulas..

Definition 1 (Complex Interaction Graph [2, 35]) For an protein-ligand complex, let $\mathcal{V}^{\mathrm{L}} := \{a_1^L, a_2^L, \ldots, a_n^L\}$ be the ligand atom set, $\mathcal{V}^{\mathrm{P}} := \{a_1^P, a_2^P, \ldots, a_m^P\}$ be the protein atom set. We define the complex interaction graph as a direction graph $\mathcal{G}_I := (\mathcal{V}, \mathcal{E})$, where the node set is

$$\mathcal{V} := \mathcal{V}^L \cup \{a_i \in \mathcal{V}^{\mathrm{P}} : \exists\, a_j^L \in \mathcal{V}^{\mathrm{L}}, \|c(a_i) - c(a_j^L)\| \leq d\},$$

and the edge set is

$$\mathcal{E} := \{(a_i, a_j) \in \mathcal{V} \times \mathcal{V} : \|c(a_i) - c(a_j)\| \leq d\}.$$

Here $c(\cdot)$ sents each atom to is 3D coordinate, $\|\cdot\|$ is an Euclidean distance, and d is a cutoff distance.

Definition 2 (Edge-oriented Neighbors [2]) In the complex interaction graph $\mathcal{G}_I$, for an atom node a_i or a directed edge e_{ij} (i.e., $a_i \rightarrow a_j$), the edge-oriented neighbors N_e of a_i or e_{ij} are defined as the sets of directed edges $\{e_{ki}, \ldots, e_{li}\}$ which point to the target atom a_i or the target edge e_{ij}.

Definition 3 (Ollivier Ricci Curvature [42]) For a graph $\mathcal{G} := (\mathrm{V}, \mathrm{E})$, given a $\alpha \in [0, 1]$, α-Ricci-curvature k_α of nodes a_i and a_j is defined to be

$$k_\alpha(a_i, a_j) := 1 - \frac{\mathrm{W}(m^\alpha_{a_i}, m^\alpha_{a_j})}{d(a_i, a_j)},$$

where $d(a_i, a_j)$ is the graph distance between two vertices a_i and a_j, m^α_a is a probability measure defined as

$$m^\alpha_a(a_k) := \begin{cases} \alpha & a_k = a \\ \frac{1-\alpha}{\deg(a)} & a_k \in \mathrm{N}(a) \\ 0 & otherwise, \end{cases}$$

and $\mathrm{W}(\cdot, \cdot)$ is the transportation distance between two probability distributions m_1 and m_2, is defined by

$$\mathrm{W}(m_1, m_2) := \inf_{\mathrm{A}} \sum_{a_1, a_2 \in V} \mathrm{A}(a_1, a_2) d(a_1, a_2).$$

Here $\deg(\cdot)$ sents each node to its degree, $\mathrm{N}(a)$ is the neighbors of node a, and the map $\mathrm{A} : \mathrm{V} \times \mathrm{V} \rightarrow [0, 1]$ is a coupling between m_1 and m_2 such that

$$\sum_{a_2 \in \mathrm{V}} \mathrm{A}(a_1, a_2) = m_1(a_1) \qquad \text{and} \qquad \sum_{a_1 \in \mathrm{V}} \mathrm{A}(a_1, a_2) = m_2(a_2).$$

Definition 4 (Foramn Ricci Curvature [24]) When a graph $\mathcal{G} := (\mathrm{V}, \mathrm{E})$ is composed of nodes, edges and triangles, otherwise, Forman-Ricci-curvature F of an edge $(a_1, a_2) \in \mathrm{E}$ is defined to be

$$F(a_1, a_2) := 4 - \deg(a_1) - \deg(a_2) + 3\Delta_{a_1 a_2},$$

otherwise, it defined to be

$$F(a_1, a_2) := 4 - \deg(a_1) - \deg(a_2),$$

where $\Delta_{a_1 a_2}$ is the number of triangular containing the edge (a_1, a_2).

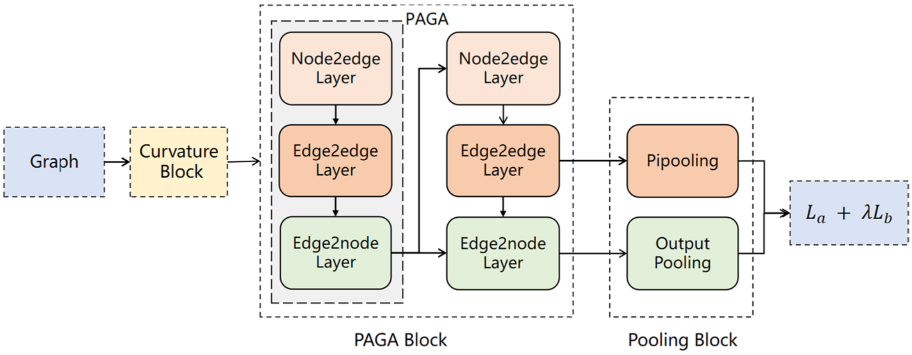

Fig. 1 Illustration of the proposed CurvAGN framework. CurvAGN is composed of a curvature block, a PAGA block, and a pooling block. The curvature block encodes multiscale curvature structure and PAGA block incorporates the geometric information including distance, angle, and multiscale curvature, and the heterophily of protein-ligand complex graph into the representation of the complex. The pooling block outputs the co-occurrent frequency of atom pairs by the Pipooling layer and the prediction of the binding affinity by the output pooling layer

Curvature-based adaptive graph neural networks

In this section, we present our model, called CurvAGN (Curvature-based Adaptive Graph Neural Network). We begin by giving an overview of the framework, followed by a detail description of each component.

Overview

The overall framework of CurvAGN is shown in Fig. 1. It takes a complex interaction graph $\mathcal{G}_I$ as input and is made up of three blocks: a curvature block, a PAGA block, and a pooling block. The first two blocks, namely the curvature and PAGA block, use a 3D model to capture the geometric structure of the protein-ligand complex interaction graph. Specifically, the curvature block captures the multiscale curvature information of the graph, while PAGA learns the spatial distance and angle information. The pooling block then gets the prediction of the binding affinity and the co-occurrent frequency of atom pairs, such as the Carbon-Carbon co-occurrence frequency.

The PAGA block is composed of multiple PAGA layers, where each layer has a node2edge layer, an edge2edge layer, and an edge2node layer. The node2edge layer utilizes the graph attention mechanism (GAT) to fuse the attribute information of the nodes at both ends of an edge into the edge attributes. The edge2edge layer uses the adaptive GAT to convert the angle information and edge attributes obtained from the first layer into edge representations. Lastly, the edge2node layer employs the adaptive graph attention mechanism to learn the node representations.

The pooling block consists of an output pooling layer and a Pipooling layer. The former generates the binding affinity prediction, while the latter produces the co-occurrent frequency of atom pairs.

The curvature block

Ricci curvature measures the extent to which a smooth object deviates from being flat. Two different discrete forms of Ricci curvature, Ollivier and Forman, have been incorporated into graph neural networks [25, 26]. Biomolecules often have hierarchical and multiscale structures, requiring a multiscale Ricci curvature to accurately characterize these structures and interactions. Such curvature has been proposed in [27, 43]. However, their curvature descriptor for protein-ligand complexes relies on prior knowledge such as the average and variance of all curvatures, and therefore, is not universally applicable. In contrast, we propose a multiscale curvature for each edge of the graph, making it a more versatile and flexible solution.

Let $\mathrm{dc} : \mathcal{E} \to \mathbb{R}$ be a discrete curvature function defined on the edge set $\mathcal{E}$ of a complex interaction graph, where $\mathbb{R}$ denotes the set of real numbers. The curvature of an edge e_{ij} of the graph is denoted by $\mathrm{dc}(e_{ij})$ or simply dc_{ij}.

To define the multiscale curvature, we first select a sequence of filtration values $\{l_i : l_0 < l_1 < \cdots < l_{n-1}\}$, then for each l_k, construct a subgraph $\mathcal{G}^{(k)}$ by removing edges with weight greater than l_k from the original graph and compute the curvature $\mathrm{dc}_{ij}^{(k)}$ of each edge e_{ij} in the subgraph.

The multiscale curvature fc_{ij} for an edge e_{ij} in the original graph is defined by concatenating the curvatures of the edge in subgraphs according to the order of the sequences, as follows:

$$\mathrm{fc}_{ij} := \|_{0 \le k < n} \mathrm{dc}_{ij}^{(k)},$$

where $\|$ represents concatenation. If the edge e_{ij} is not in the subgraph $\mathcal{G}^{(k)}$, we set its curvature dc_{ij} as zero.

We then apply a dense layer to obtain a multiscale curvature embedding:

$$f_{ij} := \mathrm{Softmax}\big(\mathrm{LeakyRelu}(\mathrm{W}_f \cdot \mathrm{fc}_{ij})\big), \tag{1}$$

where W_f is a transformation matrix.

As described in [2], we set a one-hot vector x_{ij} for the weight of edge e_{ij} by taking its integer part. Then the distance embedding for the edge is giving by

$$d_{ij} := \mathrm{W}_d x_{ij}, \tag{2}$$

where $\mathrm{W}_d \in \mathbb{R}^{n_w \times m}$ is a transformation matrix, and n_w represents the dimension of the embedding.

Finally, we define the curvature block as follows:

$$\mathrm{crt}_{ij} := \mathrm{ReLU}\big(W_{fd} \cdot [d_{ij} | f_{ij}]\big), \tag{3}$$

where W_{fd} is a transformation matrix and d_{ij} is the distance embedding in Eq. 2.

The polar-inspired adaptive graph attention block

PAGA is an adaptive graph attention network that models the 3D structure of the complex interaction graph. Compared to PGAL [2], which uses a polar-inspired graph attention block, PAGA focuses on the adaptive graph attention mechanism and the varying dependency of different attributes of a node on a neighboring node. PAGA decomposes the layer into node2edge, edge2edge, and edge2node layers, which allows for a more granular understanding of the structural information.

The node2edge layer

The node2edge layer passes node information to its edges in order to get the edge representation. In the case of PAGA, we need to add angle information to the 3D model, which requires the transportation of node information to edges. This is done by defining the l-th layer of the node2edge layer as follows:

$$h_{e_{ij}}^{(l)} := \mathrm{ReLU}\left(\mathrm{W}_{ab}^{(l)}[h_{a_i}^{(l-1)} \| h_{a_j}^{(l-1)} \| \mathrm{crt}_{ij}]\right), \tag{4}$$

here, $\mathrm{W}_{ab}^{(l)}$ is a transformation matrix and $h_{a_i}^{l-1}$ is the $(l-1)$-layer node representation of the node a_i.

The edge2edge layer

The edge2edge layer uses the adaptive graph attention mechanism to update the edge information based on the angles. To apply angle information, we construct a directed line graph and get subgraphs of the line graph by classifying the angles between edges in the original graph.

The directed line graph of the complex interaction graph is a dual graph where the nodes, node attributes, and edge-oriented neighbors of the nodes correspond respectively to the edges, the edge representations, and edge-oriented neighbors of the edges in the original graph. The weight of a directed edge between nodes in the dual is defined as the angle between the corresponding to edges in the complex interaction graph.

To get the subgraph of the line graph, we set N angle domains, denoted as $(\frac{180^\circ*(q-1)}{N}, \frac{180^\circ*q}{N}]$, for $q = 1, 2, \ldots, N$. The q-th subgraph is the subgraph of the line graph that retains all nodes but only edges of weights in the q-th angle domain. We denote the neighbors of a node e_{ij} in the q-th subgraph by $\mathrm{N}_e^q(e_{ij})$. The aggregation process for the q-th local node representation is defined as follows:

$$m_{ij,q}^{(l)} := \sum_{e_{ki} \in \mathrm{N}_e^q(e_{ij})} \alpha_{ki,q}^{(l)} \odot h_{e_{ki}}^{(l)} + h_{e_{ij}}^{(l)}, \tag{5}$$

$$\alpha_{ki,q}^{(l)} := \tanh(\mathrm{W}_{e,q}^{(l)} \cdot [h_{e_{ij}}^{(l)} \| h_{e_{ki}}^{(l)}] + b_{e,q}^{(l)}), \tag{6}$$

where the operator $\odot$ is the Hadamard product, $\mathrm{W}_{e,q}^{(l)}$ is a learnable transformation matrix, and $b_{e,q}^{(l)}$ is a learnable vector. Equation 6 applies the adaptive graph attention mechanism to get an attention vector which is viewed as the concatenation of coefficients of attributes between nodes. And $m_{ij,q}^{(l)}$ in Eq. 5 is the q-th local node representation at the l-th layer. To obtain the complete node representation, all the local aggregated node representations are combined:

$$h_{e_{i,j}}^{(l)} := [m_{ij,1}^{(l)} \| m_{ij,2}^{(l)} \| \cdots \| m_{ij,\mathrm{N}}^{(l)}].$$

For the dual, representation $h_{e_{ij}}^{(i)}$ is also the edge representation in the complex interaction graph.

The edge2node layer

The node2edge layer incorporates angle information into the edge representation. To further inject the distance and multiscale curvature information into the node representation, we design the edge2node layer based on the adaptive attention mechanism. This is in contrast to the GAT-based distance-aware attention mechanism in [2], which cannot capture heterophily.

Since the feature spaces of edges and nodes are different, we first use learnable parameter matrices $\mathrm{W}_e^{(l)}$ and $\mathrm{W}_a^{(l)}$ convert the representations of nodes and edges to the same space as follows:

$$\tilde{h}_{e_{ij}}^{(l)} := \mathrm{W}_e^{(l)} \cdot h_{e_{ij}}^{(l)}, \tag{7}$$

$$\tilde{h}_{a_j}^{(l)} := \mathrm{W}_a^{(l)} \cdot h_{a_j}^{(l-1)}, \tag{8}$$

Then we define the attention of e_{ij} with respect to a_j as

$$\beta_{ij}^{(l)} := \tanh(v_l^T \cdot [\tilde{h}_{e_{ij}}^{(l)} \| \tilde{h}_{a_j}^{(l)} \| \mathrm{W}_{dr}^{(l)} \mathrm{crt}_{ij}]), \tag{9}$$

where v_l^T is a parameter vector at the l-th layer, and $\mathrm{W}_{dr}^{(l)}$ is the learnable parameter matrix. Finally, we get the multi-head attention version of our edge2node layer by aggregating over all edges $e_{ij} \in \mathrm{N}_e(a_j)$ as follows:

$$h_{a_j}^{(l)} := \frac{1}{C} \sum_{c=1}^{C} \sum_{e_{ij} \in \mathrm{N}_e(a_j)} \beta_{ij,c}^{(l)} \cdot \tilde{h}_{e_{ij},c}^{(l)} + \tilde{h}_{a_j,c}^{(l)}, \tag{10}$$

where C is the number of attention heads and $\mathrm{N}_e(a_j)$ is the edge-oriented neighbors of node a_j.

Assuming PAGA has L polar-inspired adaptive graph attention layers, it yields the node representation $a_j^{(L)}$ for atom a_j and the edge representation $e_{ij}^{(L)}$ between atoms a_i and a_j.

The pooling block

As illustrated in [2], the pooling block is composed of a PiPooling layer and an output pooling layer. The PiPooling layer is designed to capture the long-range intermolecular interactions between the protein and ligand and output poling layer to predict the affinity.

The PiPooling layer

The PiPooling layer first divides the edges into $|\mathrm{S}_P| \times |\mathrm{S}_L|$ components, where S_P and S_L be atomic type(number) sets of the protein and its ligand, respectively. For the (T_k, T_l) -component, the pooling of edge representations is defined as

$$h_{kl} := \sum_{e_{ij} \in \mathcal{E}_I} \delta(\tau(a_i), T_k) \delta\big(\tau(a_j), T_l\big) \mathrm{W}_h e_{ij}^{(L)}, \tag{11}$$

where W_h is a shared parameter, $T_k \in \mathrm{S}_P$, $T_l \in \mathrm{S}_L$, the map τ sents each node to its atomic number, δ is a Kronecker delta function, and $\mathcal{E}_I$ is the set containing all the intermolecular edges in the complex $\mathcal{G}_I$. The output of PiPool is given by

$$\tilde{Z}_{kl} := \frac{\exp(q^T h_{kl})}{\sum_{ij} \exp(q^T h_{ij})}, \tag{12}$$

where q^T is a learnable parameter. And $\tilde{Z}_{kl}$ can be considerd an approximation of interaction matrix

$$Z_{kl} := \frac{n(T_k, T_l)}{\sum_{(a_i,a_j)\in \mathrm{S}_P \times \mathrm{S}_L} \Theta(d_\rho - d_{ij})}, \tag{13}$$

where $n(T_k, T_l) := \sum_{a_i \in \mathrm{S}_P} \sum_{a_j \in \mathrm{S}_L} \delta(\tau(a_i), T_k)\delta\big(\tau(a_j), T_l\big)\Theta(d_\rho - d_{ij})$, d_ρ is the interaction cutoff distance, and $\Theta(\cdot)$ a Heaviside step function which sents positive number to 1, and non-positive to 0.

The output pooling layer

The output pooling layer is based on a graph-level representation. We pool the node representations for the graph embedding, first. Then we apply the embedding for the affinity prediction. That is,

$$\hat{y} := \mathrm{MLP}\left(\sum_i a_i^{(L)}\right). \tag{14}$$

where $a_i^{(L)}$ is the node representation for atom i at the last layer of the PAGA model.

Optimization objective

The optimization objective of PAGA is to minimize the loss between the predicted interaction matrix $\tilde{Z}$ and the ground truth interaction matrix Z, as well as the loss between the predicted affinity $\tilde{y}$ nd the ground truth affinity y [2].

The loss function for interaction matrix is given by

$$\mathcal{L}_b := \sum_{\mathcal{G}_I \in \mathcal{D}} \|\mathrm{F}(\tilde{Z}) - \mathrm{F}(Z)\|, \tag{15}$$

where $\mathrm{F}(\cdot)$ is the flatten operation for matrix and $\mathcal{D}$ is the training set. The loss function for addinity prediction is

$$\mathcal{L}_a := \sum_{\mathcal{G}_I \in \mathcal{D}} |\hat{y} - y|. \tag{16}$$

Then the overall optimization formulated as

$$\mathcal{L} := \mathcal{L}_a + \lambda \mathcal{L}_b,$$

where λ is a hyper-parameter that controls the trade-off between the two loss terms.

Experiment

The publicly available standard PDBbind-v2016 dataset[1] is used to train and validate our module. This dataset contains a total of 13,283 protein-ligand complexes, with experimental binding affinities expressed as the negative logarithm pk_a of the determined value (e.g, $-\log K_d$, $-\log K_i$, $-\log IC_{50}$). The dataset is hierarchically structured into three nested sets: the General set, the Refined set, and the Core set, with 13,283, 4057, and 290 complexes, respectively. The Core set is used as the test set, a randomly selected subset of 1000 complexes from the difference between the Refined set and the Core set is used as the validation set. The remaining 11,993 complexes in the General set are used as the training set [11, 55].

Evaluation metrics

To evaluate the performance of our model, we employe four metrics that are widely adopted in computational biology to quantify the accuracy and precision of predictive models: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Pearson's correlation coefficient (R), and the standard deviation (SD) in regression [2, 11, 35]. RMSE and MAE provide measures of the average error between predicted and actual values, whereas R and SD are used to assess the correlation and dispersion of the predicted values, respectively. The detail is introduced in Additional file 1. We selecte these metrics to comprehensively evaluate the performance of our model on the test data.

Baselines

To demonstrate the effectiveness of our CurvAGN model, we compare it against several representative methods from different categories, including free-spatial structure methods, 3D coordinate-based methods, distance-based methods, and angle-distance based methods.

- Free-spatial structure methods: only consider the topological structure of protein-ligand complexes and neglect the spatial structure and interaction information.
  - GraphDTA [56] includes four different variants based on different types of GNNs (GCN, GAT, GIN, and GAT-GCN).
- 3D coordinate-based methods:directly utilize atomic coordinates based on GNNs.
  - SGCN [17]: is based on GCN.
- Distance-based methods: learn graph representation by employing distance information.
  - MAT [33]: learns graph representation by employing a molecule-augmented attention mechanism with the inter-atomic distances.
  - CMPNN [34]: is an edge-oriented model that strengthens the message interactions between edges (bonds) and nodes (atoms) while propagating the distance information.

[1] http://www.pdbbind.org.cn/.

Table 1 The list of atom features

Atom type	**C, N, O, S, P, I, Cl, B, F (onehot)**
Atom hybridization	1, 2, 3 (integer)
Number of heavyatoms attached	(integer)
Number of heteroatoms attached	(integer)
SMARTS patterns	Hydrophobic, aromatic, acceptor, donor and ring (onehot)
Partial charge	(float)

  - GNN-DTI [13]: leverages GAT to represent a protein-ligand complex graph constructed by the distance between atoms.
  - ELGN [35]: considers distance information and long-distance interaction information between molecules, as well as the topology information of bonds

- Angle-distance based methods: mploy angle and distance information in GNNs.

  - DimeNet [19]: employs the angle and distance information in graph neural network.
  - SIGN [2]: improves GNNs to model the 3D-structure of a protein-ligand complex by not only encoding angle and distance information, but also handling interactions in the complex.

Implementation details

Let protein set be $S_P := \{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{S}\}$ and ligand set be $S_L := \{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{S}, \mathrm{P}, \mathrm{I}, \mathrm{Cl}, \mathrm{B}, \mathrm{F}\}$, we construct the complex interaction graph and interaction matrix by setting cutoff-threshold $d_\theta = 5$ and the interaction cutoff distance $d_\rho = 12$ as previous work [8, 57].

For initial node features, we follow the approach in [2, 11], where an atom is represented by an 18-dimensional vector (refer to Table 1 in previous work). To distinguish between ligand and protein atoms, we encode an atom using a 36-dimensional vector, where the first half represents raw features and the second half are all zeros for a ligand and vice versa for a protein atom. The initial edge features consist of vectors of 26 dimensions, where the 26th dimension represents the Euclidean distance between the atoms of the edge, and the first 50 dimensions are filter Forman curvatures with filtration values set as $\{0.1 * i : i = 0, 1, 2, \ldots, 49\}$.

The distance and curvature embedding dimensions are both set to 128. Each vector undergoes transformation matrix action, resulting in an embedding of 128 dimensions. To train the model, The Adam optimizer with a learning rate of 0.001 and the batch size of 32 is used to train the model. The dropout rate is set to 0.2, and the hyper-parameter γ is set to $= 1.75$. In the PAGA layers, there are 8 attention heads and 6 angle domains. We list the all settings as following Table 2.

All the experiments are conducted on one NVIDIA GeForce RTX 2080 Ti GPU and Inter Xeon Gold 5218 16-Core Processor. And the performance of all the baselines refers to [2].

Table 2 The parameter setting for our Curv-SIAGN model

Learning rate	Batch size	Hyper-parameter	Cutoff for graph	Dim of Node_emb
0.001	32	$\lambda = 1.75$	$\theta_d = 5$	128
Heads	Dim of Edge_emb	Cutoff for matrix	Filtrations	Angle domains
4	128	$\theta_p = 12$	50	6
Dropout	Blocks	Dim of curvature		
0.2	4	128		

Table 3 The performace comparision on PDBbind-v2016 core set

Methods		RMSE↓	MAE↓	SD↓	R↑
Free-spatial	GCN	1.735 (0.034)	1.343 (0.037)	1.719 (0.027)	0.613 (0.016)
	GAT	1.765 (0.026)	1.354 (0.033)	1.740 (0.027)	0.601 (0.016)
Methods	GIN	1.640 (0.044)	1.261 (0.044)	1.621 (0.036)	0.667 (0.018)
	GAT-GCN	1.562 (0.022)	1.191 (0.016)	1.558(0.018)	0.697 (0.008)
Coordinate	SGCN	1.583 (0.033)	1.250 (0.036)	1.582 (0.320)	0.686 (0.015)
	MAT	1.457 (0.037)	1.154 (0.037)	1.445 (0.033)	0.747 (0.013)
Distance	GNN-DTI	1.492 (0.025)	1.192 (0.032)	1.471 (0.051)	0.736 (0.021)
Methods	CMPNN	1.408 (0.028)	1.117 (0.031)	1.399 (0.025)	0.765 (0.009)
	ELGN	1.285 (0.027)	1.013 (0.022)	1.263(0.026)	0.810 (0.012)
Angle	DimeNet	1.453 (0.027)	1.138 (0.026)	1.434 (0.023)	0.752 (0.010)
Methods	SIGN	1.316 (0.031)	1.027 (0.025)	1.312(0.035)	0.797 (0.012)
Ours	CurvAGN	**1.217 (0.012)**	**0.930(0.014)**	**1.191(0.015)**	**0.8305 (0.004)**

Source: We present the average (standard deviation) across 5 random runs, highlighting the best results. Note that the upward arrow ↑ indicates that a higher value is better, while the downward arrow ↓ indicates that a higher value is worse

Performance evaluation

We conduct a comparison of our CurvAGN model and baseline models on the PDBbind v2016 core set. The average and standard deviation of four indicators for testing performance, obtained from five random runs, are presented in Table 3. Overall, the results show that the CurvAGN model outperforms all other models in the dataset.

According to [2, 35], the performance of protein-ligand binding affinity prediction models is heavily influenced by their ability to utilize the spatial structure of protein-ligand complexes. GraphDTA models, which do not use spatial structure, perform poorly. SGCN, which leverages atom coordinates, performs better than the GCN, a variant of GraphDTA. However, SGCN's performance suffers because its coordinate operations are not rotation invariant. GNN-DTI, with distance information, clearly improves performance over GAT. Among distance-based methods, ELGN and CMPNN focus more on message communication between nodes and edges, resulting in better performance than MAT and GNN-DTI. ELGN leverages long-range intermolecular interactions and incorporates the topology information of bonds, resulting in the best performance among these distance-based methods.

DimeNet is capable of learning the angle and distance structure and outperforms SGCN marginally. SIGN, although it considers angle information, lacks the topology of edges, which could be the main reason for its weaker performance compared to ELGN. Our proposed CurvAGN, on the other hand, captures more spatial information in the

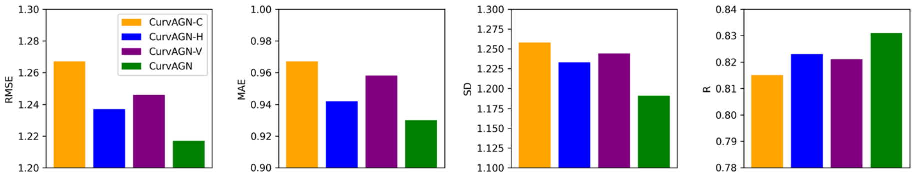

Fig. 2 The variants of the CurvAGN model. Different colors mark different models. CurvAGN (green) performs the best, followed by CurvAGN-H (blue) and CurvAGN-V (purple). CurvAGN-C (orange) performs the worst, which suggests that curvature features have a significant impact on protein-ligand binding

form of curvature and utilizes an adaptive graph attention mechanism, resulting in superior performance compared to SIGN.

Ablation analysis

To validate the importance of multi-scale curvature, heterophily, and multi-head GAT on predicting protein-ligand binding affinity, we compare CurvAGN and its variants on the test data.

- CurvAGN-C: uses the adaptive GAT layer without curvature information.
- CurvAGN-H: uses the vanilla multi-head GAT layer.
- CurvAGN-V: uses the adaptive GAT layer.

As can be observed in the Fig. 2, CurvAGN performs best among all the variants, proving the necessity of curvature information, heterophily and multi-head GAT in predicting protein-ligand binding affinity. Specifically, CurvAGN-C performs worse than CurvAGN because it fails to capture the curvature information. CurvAGN-H suffers from the lack of heterophily, which leads to a performance drop. The different attributes of nodes have varying impacts on the interactions between neighboring nodes. CurvAGN-V fails to capture this, resulting in a decrease in performance. CurvAGN-C has a larger prediction error than CurvAGN-H and CurvAGN-V, indicating that curvature information plays a greater role in improving the model's performance.

To check whether the gains made by our method are uniformly distributed across all these 290. We compare the average absolute prediction error of the SIGN and CurvAGN models on the test set across 5 random runs, and the distribution of the difference in absolute prediction error between SIGN and CurvAGN on these complexes is shown in the Fig. 3. In the Fig. 3, the x-axis represents complexes, the y-axis denotes average absolute prediction error, and the area under the curve represents the difference in the total sum of absolute errors between SIGN and CurvAGN on the test set. It is easy to see that the area under the curve above the x-axis (70.67) is greater than the area under the curve below the x-axis (41.50). This implies that CurvAGN performs better than SIGN on average. However, the gains of CurvAGN are not consistent across all complexes, as there are 127 samples with negative y-coordinates.

We compare well-performing complexes with poorly-performing complexes and find our model performs better for complexes with a high ratio of the number of ligand-protein atom pairs with a distance less than 4.8Å to the total number of ligand-protein atom pairs. This may suggest that intramolecular interactions within the protein and the

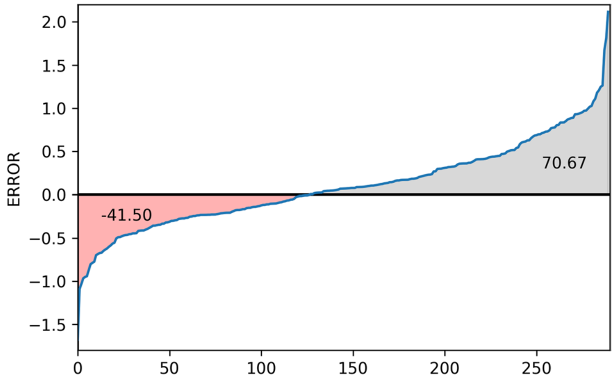

Fig. 3 Gains made by CurvAGN on each complex in the test set. The *x*-axis denotes the complexes, and the *y*-axis denotes the error between the difference in absolute prediction error between SIGN and CurvAGN on each complex. The area under curve represents the total gains made by CurvAGN on the test set. The figure shows that our method is only effective for some specific complexes

ligand interfere with the prediction. Further research and analysis is introduced in Additional file 1.

Conclusion

In this work, we propose CurvAGN, a curvature-based GNN model to predict protein-ligand binding affinity with improved performance, through incorporating the fine-grained geometric information, interaction information among atoms, and heterophily in the complex graph for enhanced representation learning. We first design a curvature block that encodes multiscale curvature information. We then introduce a polar-inspired adaptive graph attention block (PAGA) to capture the heterophily in the complex graph and also the angle and distance information. Additionally, since node attributes rely on the graph structure differently, we use vector attention in the edge2edge layer of PAGA which allows the model to learn different attention weights for different attributes in the node. Additionally, since node attributes rely on the graph structure differently, we use vector attention in the edge2edge layer of PAGA which allows the model to learn different attention weights for different attributes in the node. We train the model on the standard PDBbind-v2016 dataset and its experimental result outperforms SIGN by 7.5% in RMSE and 9.4% in MAE which confirms that the proposed CurvAGN model is effective in improving protein-ligand binding affinity prediction.

For protein-ligand binding affinity prediction, the accuracy of the prediction is important for the design and development of drugs, understanding protein function and interaction mechanisms, etc. Therefore, even if the lift in RMSE is small, our method can improve the accuracy of the prediction and provide more reliable and useful results.

Future research

We believe that further exploration is warranted to address the issue that our model may not improve prediction accuracy for all protein-ligand complexes. This investigation cannot only reveal the applicability range of our model but also provide new insights for its further improvement. Additionally, we aim to incorporate the overall geometric information of the complexes, such as topological information, into our network structure.

Finally, we aspire to apply our model to other areas of biology, such as miRNA-disease association prediction [58] and drug repositioning [59].

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-023-05503-w.

Additional file 1. Supplemental provides details of valuation metrics used in this work and the relation between complex structure and model performance.

Acknowledgements
The authors thank the anonymous reviewers for their valuable suggestions.

Author contributions

J. Wu designed research; J. Wu, H. Chen, M. Cheng and H. Xiong performed research; J. Wu and H. Chen analyzed data; and J. Wu, H. Chen, M. Cheng and H. Xiong wrote the paper.

Funding
This work is supported by the National Key Research and Development Program of China (2022YFB4500300) and, in part, by key research project of Zhejiang Lab (No. 2022PI0AC01).

Availibility of data and materials

We use the publicly available standard PDBbind-v2016 dataset http://www.pdbbind.org.cn/.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Code available

Code can be found from this link https://github.com/tumancao/CurvAGN.

Competing interest

No competing interest is declared.

Received: 5 July 2023 Accepted: 28 September 2023
Published online: 05 October 2023

References

1. Liu X, Huitao F, Jie W, Kelin X. Persistent spectral hypergraph-based machine learning (PSH-ML) for protein-ligand binding affinity prediction. Brief Bioinform. 2021;22(5)
2. Li S, Zhou J, Xu T, Huang L, Wang F, Xiong H, Huang W, Dou D, Xiong H. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining; 2021. p. 975–985
3. Allen WJ, Balius TE, Mukherjee S, Brozell SR, Moustakas DT, Lang PT, Case DA, Kuntz ID, Rizzo RC. Dock 6: impact of new features and current docking performance. J Comput Chem. 2015;36(15):1132–56.
4. Jain AN. Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. J Med Chem. 2003;46(4):499–511.
5. Vina A. Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading trott, oleg; olson, arthur j. J Comput Chem. 2010;31(2):455–61.
6. Wang DD, Chan M-T. Protein-ligand binding affinity prediction based on profiles of intermolecular contacts. Comput Struct Biotechnol J. 2022;20:1088–96.
7. Kinnings SL, Liu N, Tonge PJ, Jackson RM, Xie L, Bourne PE. A machine learning-based method to improve docking scoring functions and its application to drug repurposing. J Chem Inf Model. 2011;51(2):408–19.
8. Ballester PJ, Mitchell JB. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. Bioinformatics. 2010;26(9):1169–75.
9. Dong S, Wang P, Abbas K. A survey on deep learning and its applications. Comput Sci Rev. 2021;40: 100379.
10. Ragoza M, Hochuli J, Idrobo E, Sunseri J, Koes DR. Protein-ligand scoring with convolutional neural networks. J Chem Inf Model. 2017;57(4):942–57.
11. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. Bioinformatics. 2018;34(21):3666–74.
12. Wallach, I., Dzamba, M., Heifets, A. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. 2015 arXiv preprint arXiv:1510.02855

13. Lim J, Ryu S, Park K, Choe YJ, Ham J, Kim WY. Predicting drug-target interaction using a novel graph neural network with 3D structure-embedded graph representation. J Chem Inf Model. 2019;59(9):3981–8.
14. Sun M, Zhao S, Gilvary C, Elemento O, Zhou J, Wang F. Graph convolutional networks for computational drug development and discovery. Brief Bioinform. 2019;21(3):919–35. https://doi.org/10.1093/bib/bbz042.
15. Jiang D, Hsieh C-Y, Wu Z, Kang Y, Wang J, Wang E, Liao B, Shen C, Xu L, Wu J, Cao D, Hou T. Interactiongraphnet: a novel and efficient deep graph representation learning framework for accurate protein-ligand interaction predictions. J Med Chem. 2021;64(24):18209–32. https://doi.org/10.1021/acs.jmedchem.1c01830.
16. Jiang D, Hsieh C-Y, Wu Z, Kang Y, Wang J, Wang E, Liao B, Shen C, Xu L, Wu J. Interactiongraphnet: a novel and efficient deep graph representation learning framework for accurate protein-ligand interaction predictions. J Med Chem. 2021;64(24):18209–32.
17. Danel T, Spurek P, Tabor J, Smieja M, Struski L, Slowik A, Maziarka L. Spatial graph convolutional networks. In: Neural information processing: 27th international conference, ICONIP 2020, Bangkok, Thailand, November 18–22, 2020, Proceedings, Part V. Springer; 2020. p. 668–75.
18. Volkov M, Turk J-A, Drizard N, Martin N, Hoffmann B, Gaston-Mathe Y, Rognan D. On the frustration to predict binding affinities from protein-ligand structures with deep neural networks. J Med Chem. 2022;65(11):7946–58.
19. Gasteiger J, Groß J, Günnemann S. Directional message passing for molecular graphs. In: International conference on learning representations 2019
20. Leach AR. Molecular modelling: principles and applications. London: Pearson Education; 2001.
21. Helms V, Wade RC. Computational alchemy to calculate absolute protein- ligand binding free energy. J Am Chem Soc. 1998;120(12):2710–3.
22. Cao Y, Li L. Improved protein-ligand binding affinity prediction by using a curvature-dependent surface-area model. Bioinformatics. 2014;30(12):1674–80.
23. Chung FR, Yau S-T. Logarithmic Harnack inequalities. Math Res Lett. 1996;3(6):793–812.
24. Forman R. Bochner's method for cell complexes and combinatorial RICCI curvature. Discret Comput Geom. 2003;29(3):323–74.
25. Li H, Cao J, Zhu J, Liu Y, Zhu Q, Wu G. Curvature graph neural network. Inf Sci. 2022;592:50–66.
26. Ye Z, Liu KS, Ma T, Gao J, Chen C. Curvature graph network. In: International conference on learning representations 2019
27. Wee J, Xia K. Ollivier persistent RICCI curvature (OPRC) based molecular representation for drug design; 2020; arXiv preprint arXiv:2011.10281
28. Yu L, Qiu W, Lin W, Cheng X, Xiao X, Dai J. Hgdti: predicting drug-target interaction by using information aggregation based on heterogeneous graph neural network. BMC Bioinform. 2022;23(1):126.
29. Yang Z, Zhong W, Lv Q, Dong T, Yu-Chian Chen C. Geometric interaction graph neural network for predicting protein-ligand binding affinities from 3d structures (GIGN). J Phys Chem Lett. 2023;14(8):2020–33.
30. Yang T, Wang Y, Yue Z, Yang Y, Tong Y, Bai J. Graph pointer neural networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 36; 2022. p. 8832–8839
31. Sun J, Zhang L, Zhao S, Yang Y. Improving your graph neural networks: a high-frequency booster; 2022 arXiv preprint arXiv:2210.08251
32. Bo D, Wang X, Shi C, Shen H. Beyond low-frequency information in graph convolutional networks. In: Proceedings of the AAAI conference on artificial intelligence, vol. 35; 2021. p. 3950–3957
33. Maziarka L, Danel T, Mucha S, Rataj K, Tabor J, Jastrz kebski Sl. Molecule attention transformer. 2020 arXiv preprint arXiv:2002.08264
34. Song Y, Zheng S, Niu Z, Fu ZH, Lu Y, Yang Y. Communicative representation learning on attributed molecular graphs. In: IJCAI, vol. 2020; 2020, p. 2831–2838
35. Yi Y, Wan X, Zhao K, Ou-Yang L, Zhao P. Predicting protein-ligand binding affinity with equivariant line graph network. 2022; arXiv preprint arXiv:2210.16098
36. Jost J, Jost J. Riemannian geometry and geometric analysis, vol. 42005. Berlin: Springer; 2008.
37. Najman L, Romon P. Modern approaches to discrete curvature. Berlin: Springer; 2017.
38. Ollivier Y. Ricci curvature of metric spaces. CR Math. 2007;345(11):643–6.
39. Lott J, Villani C. Ricci curvature for metric-measure spaces via optimal transport. Ann Math. 2009;03–991
40. Ollivier Y. Ricci curvature of Markov chains on metric spaces. J Funct Anal. 2009;256(3):810–64.
41. Bonciocat A-I, Sturm K-T. Mass transportation and rough curvature bounds for discrete spaces. J Funct Anal. 2009;256(9):2944–66.
42. Lin Y, Lu L, Yau S-T. Ricci curvature of graphs. Tohoku Math J Second Ser. 2011;63(4):605–27.
43. Wee J, Xia K. Forman persistent RICCI curvature (FPRC)-based machine learning models for protein-ligand binding affinity prediction. Brief Bioinform. 2021;22(6):136.
44. Sreejith R, Mohanraj K, Jost J, Saucan E, Samal A. Forman curvature for complex networks. J Stat Mech Theory Exp. 2016;2016(6): 063206.
45. Pandit S, Chau DH, Wang S, Faloutsos C. Netprobe: a fast and scalable system for fraud detection in online auction networks. In: Proceedings of the 16th international conference on world wide web; 2007, p. 01–210
46. Du L, Shi X, Fu Q, Ma X, Liu H, Han S, Zhang D. GBK-GNN: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In: Proceedings of the ACM web conference 2022; 2022. p. 1550–1558
47. Fang Z, Xu L, Song G, Long Q, Zhang Y. Polarized graph neural networks. In: Proceedings of the ACM web conference 2022; 2022. p. 1404–1413
48. Jin D, Yu Z, Huo C, Wang R, Wang X, He D, Han J. Universal graph convolutional networks. Adv Neural Inf Process Syst. 2021;34:10654–64.
49. Li Y, Lin B, Luo B, Gui N. Graph representation learning beyond node and homophily. IEEE Trans Knowl Data Eng. 2022;
50. Abu-El-Haija S, Perozzi B, Kapoor A, Alipourfard N, Lerman K, Harutyunyan H, Ver Steeg G, Galstyan A. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In: International conference on machine learning. PMLR; 2019. p. 21–29

51. Bi W, Du L, Fu Q, Wang Y, Han S, Zhang D. MM-GNN: Mix-moment graph neural network towards modeling neighborhood feature distribution. In: Proceedings of the 16th ACM international conference on web search and data mining, 2023. p. 132–140
52. Dong Y, Ding K, Jalaian B, Ji S, Li J. Adagnn: Graph neural networks with adaptive frequency response filter. In: Proceedings of the 30th ACM international conference on information & knowledge management, 2021. p. 392–401
53. Li S, Kim D, Wang, Q. Beyond low-pass filters: adaptive feature propagation on graphs. In: Machine learning and knowledge discovery in databases. research track: European conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21. Springer; 2021. p. 450–465
54. Pei H, Wei, B, Chang KCC, Lei Y, Yang B. Geom-GCN: geometric graph convolutional networks. 2020;arXiv preprint arXiv:2002.05287
55. Zheng L, Fan J, Mu Y. Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein-ligand binding affinity prediction. ACS Omega. 2019;4(14):15956–65.
56. Nguyen T, Le H, Quinn TP, Nguyen T, Le TD, Venkatesh S. Graphdta: predicting drug-target binding affinity with graph neural networks. Bioinformatics. 2021;37(8):1140–7.
57. Muegge I, Martin YC. A general and fast scoring function for protein- ligand interactions: a simplified potential approach. J Med Chem. 1999;42(5):791–804.
58. He Y, Yang Y, Su X, Zhao B, Xiong S, Hu L. Incorporating higher order network structures to improve miRNA disease association prediction based on functional modularity. Brief Bioinform. 2022;24(1):562.
59. Zhao BW, Wang L, Hu PW, Wong L, Su XR, Wang BQ, You ZH, Hu L. Fusing higher and lower-order biological information for drug repositioning via graph representation learning. IEEE Trans Emerg Top Comput. 2023. https://doi.org/10.1109/TETC.2023.3239949.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.