

Research Article

Curvature-enhanced graph convolutional network for biomolecular interaction prediction

Cong Shen^{a,d}, Pingjian Ding^b, Junjie Wee^d, Jialin Bi^c, Jiawei Luo^{a,*}, Kelin Xia^{d,*}^a College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410000, China^b Center for Artificial Intelligence in Drug Discovery, School of Medicine, Case Western Reserve University, Cleveland, OH 44106, USA^c School of Mathematics, Shandong University, Jinan, 250100, China^d School of Physical and Mathematical Sciences, Nanyang Technological University, 637371, Singapore

ARTICLE INFO

Keywords:

Ollivier-Ricci curvature
Graph convolutional network
Biomolecular interaction

ABSTRACT

Geometric deep learning has demonstrated a great potential in non-Euclidean data analysis. The incorporation of geometric insights into learning architecture is vital to its success. Here we propose a curvature-enhanced graph convolutional network (CGCN) for biomolecular interaction prediction. Our CGCN employs Ollivier-Ricci curvature (ORC) to characterize network local geometric properties and enhance the learning capability of GCNs. More specifically, ORCs are evaluated based on the local topology from node neighborhoods, and further incorporated into the weight function for the feature aggregation in message-passing procedure. Our CGCN model is extensively validated on fourteen real-world biomolecular interaction networks and analyzed in details using a series of well-designed simulated data. It has been found that our CGCN can achieve the state-of-the-art results. It outperforms all existing models, as far as we know, in thirteen out of the fourteen real-world datasets and ranks as the second in the rest one. The results from the simulated data show that our CGCN model is superior to the traditional GCN models regardless of the positive-to-negative-curvature ratios, network densities, and network sizes (when larger than 500).

1. Introduction

Responsible for nearly all the functional properties, including reproducibility, sustainability, and mortality, biomolecules are of fundamental importance to all life forms, ranging from microorganisms and plants to animals. The monomers, oligomers and macromolecules made from amino acids, peptides, proteins, nucleobases, nucleotides, oligonucleotides, nucleic acids (DNA/RNA), monosaccharides, oligosaccharides, polysaccharides, or lipids, are the major building blocks of life [1–3]. Biomolecular interactions can happen between two or more molecules, and involve single or multiple binding sites in a cooperative way [4–6]. The analysis of biomolecular interactions is of great interests to scientists. Experimentally, the study of these interactions is often time-consuming and labor-intensive. This stimulated the development of computational-based models. Mathematically, the analysis of biomolecular interactions can be transformed into node, edge, subgraph, or other graph property prediction problems, resulting in the development of various graph or network models. Recently, graph neural network (GNN) models have been used to learn the information from

the graph data. These GNN models can perform various tasks, including node classification [7], link prediction [8,9], graph classification [10,11], and graph property prediction [12–14]. They have demonstrated great potential in biomolecular graph data analysis [15–17].

The two essential components of all GNNs are node neighborhood and feature aggregation. For node neighborhood, its most commonly-used definition is that for a certain node (or vertex), all the other nodes that directly connected (through one edge) with this specific node are known as its neighbors. This definition is widely used in GNNs, including Graph Convolutional Network (GCN) [18], Graph Attention Network (GAT) [19], and Graph Isomorphism Network (GIN) [20]. The neighborhood of a node can also be defined from random walk methods, in which all the nodes in the path of a random walk are regarded as the neighbors of the initial starting node. This definition is used in GNNs, such as HetGNN [8]. Feature aggregation, which is key to message passing, is to systematically aggregate the node features (i.e., feature vectors) to update node representations. In general, there are two types of feature aggregation. First, features are aggregated with equal importance. This approach is widely used in models, including

* Corresponding authors.

E-mail addresses: luojiawei@hnu.edu.cn (J. Luo), xiakelin@ntu.edu.sg (K. Xia).<https://doi.org/10.1016/j.csbj.2024.02.006>

Received 22 November 2023; Received in revised form 7 February 2024; Accepted 7 February 2024

Available online 15 February 2024

2001-0370/© 2024 The Authors. Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

GIN [20], GraphSAGE [21], and Neural FPs [22]. Second, features are aggregated with different weights. In GCN [18], the weights are determined by node degrees. In GAT [19], the weights are evaluated through an attention mechanism, in which feature vectors of the node and its neighbors are multiplied to calculate the weight (or importance) of the neighboring nodes to the specific node.

Geometric deep learning models have been proposed to incorporate geometric information into deep learning architectures [23–25]. As one of the fundamental concepts in differential geometry, Ricci curvature characterizes the intrinsic properties of manifold surfaces [26,27]. Recently, discrete Ricci curvature models, including Ollivier-Ricci curvature (ORC) [28–30] and Forman Ricci curvature (FRC) [31–33], have been developed and used in various applications, such as internet topology [34], community detection [35,36], market fragility and systemic risk [37], cancer networks [38], brain structural connectivity [39], and biomolecular systems [40,41]. In particular, discrete Ricci curvatures have been used in the characterization of “over-squashing” phenomenon [42], which happens at the bottleneck region of a network when the messages propagated from distant nodes distort significantly. Due to the low efficiency of information exchange in the bottleneck region, special attention should be paid to this part in GNN models. Further, ORC has been used in protein engineering, drug discovery and cell engineering models. For example, Wee et al. used Forman-Ricci curvature, Ollivier Ricci curvature, simplicial complexes and machine learning to predict protein-ligand interactions [40,41]. Murgas et al. investigated Forman-Ricci curvature, along with network entropy, to explore the relationship of the two quantities as they occur in gene networks [43]. More recently, curvature-based graph neural network models have been developed by the incorporation of ORCs into GNN models [44,45]. These models have achieved great success in various synthetic and real-world graphs, from social networks, coauthor networks, citation networks, and Amazon co-purchase graph. The curvature graph network model can significantly outperform state-of-the-art (SOTA) when the underlying graphs are of large-sized and dense.

Here we propose a Curvature-enhanced Graph Convolutional Network (CGCN) model for biomolecular interaction prediction. In our CGCN model, the ORC is calculated for edges of biomolecular interaction graphs. An ORC-based multilayer perception model is trained and its output value is used as the weight in message-passing module. In this way, our CGCN model is aware of the local geometric information and is robust for “over-squashing”. Our model has been systematically compared with eight SOTA models on fourteen commonly used molecular interaction datasets. It has been found that the proposed model can outperform all SOTA models. Further simulation tests are employed to explore the applicability of our CGCN model. It has been found that the CGCN model consistently delivers better results than traditional GCN model. This performance is highly robust to both network densities and ratios between positive ORCs and negative ORCs. Further, consistently with previous results, our ORC-based CGCN model has a better performance on molecular interaction graphs of medium or large sizes (i.e., >500 nodes).

2. Related works

2.1. Ollivier-Ricci curvature for graph data analysis

Ollivier-Ricci curvature is a discrete Ricci curvature model that is developed for the analysis of graph data. ORC has been combined with deep learning models and demonstrated great advantages. A major reason is that Ricci curvature is found to be related to “over-squashing” phenomenon in message aggregation process and can be used to alleviate information distort in message-passing-based GNNs [42]. RicciNet has been developed to identify the salient computational paths with Ricci curvature-guided pruning [46]. A Ricci flow process, which is parameterized by a reinforcement learning controller, is employed to

deform the discrete space of the graph by the systematical removing of edges with negative Ricci curvatures. Curvature Graph Network (CurvGN) has been proposed to incorporate the Ricci curvature information into graph convolutional network so that it can adapt to different local structural topology [44]. An ORC-based message-passing operator is developed by the aggregation of node representations with an ORC-related weight factor, which is obtained through a multi-layer perceptron (MLP) with ORC as its input. Further, Curvature Graph Neural Network (CGNN) has been developed to increase topological adaptivity of GNNs [45]. Similar to CurvGN, ORC information is transformed into the weights.

Curvature has also been employed in the characterization of embedding spaces. Curvature Graph Generative Adversarial Network (CurvGAN) has been proposed to better preserve the topological properties and alleviate topological distortions [47]. In CurvGAN, global topology of the graph data is approximated by a Riemannian geometric space with constant curvature and local heterogeneous topology is characterized by ORCs. Hyperbolic Curvature Graph Neural Network (HCGNN) integrates discrete and continuous curvature together to enhance hyperbolic geometric learning [48]. Similar to CurvGAN, global topology is characterized by constant curvature manifold and local heterogeneous topology by ORCs. However, in HCGNN, the embedding space is modeled by a hyperbolic space with constant curvature, and ORCs is incorporated into message passing operator though hyperbolic curvature-aware message propagation and ORC-based homophily constraint. Other discrete curvature models have also been employed in learning models, including curvature-informed multi-task learning for graph networks [49], mixed-curvature multi-relational graph neural network for knowledge graph completion [50], adaptive curvature exploration hyperbolic graph neural network (ACE-HGNN) [51], etc.

2.2. Graph neural network for molecular interaction prediction

Recently, the application of graph neural networks in multifarious molecular interaction prediction tasks has received increasing attention. For instance, SkipGNN [52] utilizes a skip graph neural network to predict molecular interactions. MR-GNN [53] infers the interaction between two entities via a dual graph neural network. CSGNN [54] uses a contrastive self-supervised graph neural network to predict molecular interactions. Besides, some graph neural network models are applied on some specific molecular interactions. KGNN [55] is a knowledge graph neural network and MIRACLE [56] is a multi-view graph contrastive representation learning model, both used to predict drug-drug interactions. KGE_NFM [57] a unified framework for drug-target interaction prediction by combining knowledge graph and recommendation system. IDDkin [58] is a network-based influence deep diffusion model for kinase inhibitors prediction. InteractionGraphNet [59] is a novel deep graph representation learning framework for accurate protein-ligand interaction predictions.

3. Methods

3.1. Mathematical notations

Here we used uppercase letters for matrices (e.g. $W \in \mathbb{R}^{m \times n}$) and lowercase letters to denote vectors (e.g. $h \in \mathbb{R}^d$). We used an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ to represent an interaction network, where \mathcal{V} is the set of vertices and \mathcal{E} is the set of edges. Here $v_i \in \mathcal{V}$ is i -th node and $e_{ij} \in \mathcal{E}$ is the edge between i -th node and j -th node. The edge is formed only when there exists a certain interaction between the two nodes. Furthermore, $c(v_i, v_j)$ represents the Ollivier-Ricci curvature on edge $e_{v_i v_j}$.

3.2. Ollivier-Ricci curvature

Ricci curvature measures the growth of volumes of distance balls, transportation distances between balls, divergence of geodesics, and

meeting probabilities of coupled random walks [32]. Ricci curvature equals to the classical Gauss curvature on two dimensional manifold. Two discrete Ricci curvature forms, i.e., Ollivier Ricci curvature (ORC) [60,61,29,30] and Forman Ricci curvature (FRC) [62,31], have been developed. Among them, the most widely used one is ORC, which was originally proposed on metric spaces [60,29] and further applied to graphs [63,64]. ORC is defined on graph edges. It measures the difference between the edge “distance” (or length of edge) and transportation distance of two probability distributions, which are defined respectively on the two neighborhoods from the edge-related two vertices. Roughly speaking, positive edge ORC means that there are strong connections (or short “distance”) between the two respective neighborhoods, and negative edge ORC indicates weak connections (or long “distance”). It has been found that ORC is also related to various graph invariants, ranging from local measures, such as node degree and clustering coefficient, to global measures, such as betweenness centrality and network connectivity [34].

Mathematically, for a node v_i in a graph $G = \{\mathcal{V}, \mathcal{E}\}$, its neighbors can be expressed as $\mathcal{N}(v_i) = \{v_{i_1}, v_{i_2}, \dots, v_{k_{v_i}}\}$, and the total number of neighbors is k_{v_i} , which is the degree of node v_i . A probability distribution m_{v_i} is defined as,

$$m_{v_i}(v_j) = \begin{cases} \alpha & \text{if } v_j = v_i \\ (1 - \alpha)/k_{v_i} & \text{if } v_j \in \mathcal{N}(v_i) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where parameter $\alpha \in [0, 1]$. Here we use $\alpha = 0.5$, which is the most commonly used value [63,64]. Note that m_{v_i} is a discrete probability distribution function. On the center node v_i , it is defined to be α , and on the neighboring nodes v_j , it is defined to be $(1 - \alpha)/k_{v_i}$.

If there is an edge $e_{v_1 v_2}$ between node v_1 and v_2 , a measure $\xi \in \Pi(m_{v_1}, m_{v_2})$ between two probability distributions m_{v_1} and m_{v_2} defines a transportation plan from m_{v_1} to m_{v_2} . This measure is mass-preserving, i.e., $\sum_{v_j \in \mathcal{V}} \xi(v_1, v_j) = m_{v_1}$ and $\sum_{v_i \in \mathcal{V}} \xi(v_i, v_2) = m_{v_2}$. The amount of mass moved from v_i to v_j is $\xi(v_i, v_j)$, and $d(v_1, v_2)$ is the distance between node v_1 and node v_2 . The L_1 -Wasserstein distance between m_{v_1} and m_{v_2} , which is minimum average traveling distance and represented by $w_{L_1}(m_{v_1}, m_{v_2})$ can be computed,

$$w_{L_1}(m_{v_1}, m_{v_2}) = \inf_{\xi} \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} d(v_i, v_j) \xi(v_i, v_j). \quad (2)$$

The Ollivier-Ricci curvature on edge $e_{v_1 v_2}$, denoted as $c(v_1, v_2)$, is defined as follows,

$$c(v_1, v_2) = 1 - \frac{w_{L_1}(m_{v_1}, m_{v_2})}{d(v_1, v_2)}. \quad (3)$$

Computationally, linear programming (LP) is utilized to calculate Wasserstein distance. Let $\rho(v_i, v_j) \geq 0$ represent the fraction of “mass” transported for node v_i to v_j , the LP formulation can be expressed as follows,

$$\min \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} d(v_i, v_j) \rho(v_i, v_j) m_{v_i}(v_i) \quad (4)$$

$$\text{s.t. : } \sum_{v_j \in \mathcal{V}} \rho(v_i, v_j) = 1, \quad 0 \leq \rho(v_i, v_j) \leq 1$$

$$\sum_{v_i \in \mathcal{V}} \rho(v_i, v_j) m_{v_i}(v_i) = m_{v_j}(v_j).$$

Note that ORCs are calculated on edges of graphs. The node ORCs are usually defined as the average of edges ORCs. That is for a node v_i , its ORC value is

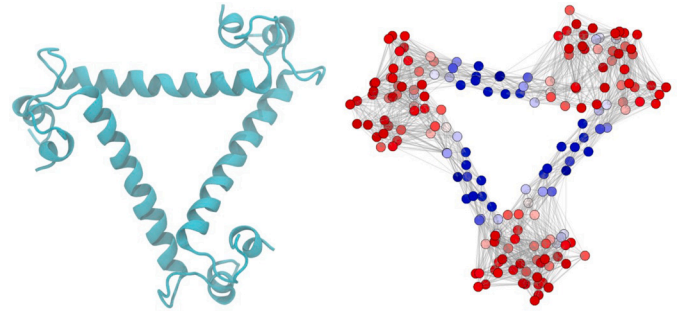


Fig. 1. The illustration of Ollivier-Ricci curvature on a biomolecular structure network. The colors represent the ORC values on the nodes. The red color indicates positive ORC values, while the blue color represents negative values. It can be seen that the negative ORC values all concentrated in the bottleneck or linkage regions. In the well-connected clusters or communities, ORC values are all positive.

$$c(v_i) = \frac{1}{\deg(v_i)} \sum_{v_j \in \mathcal{N}(v_i)} c(v_i, v_j), \quad (5)$$

where $\mathcal{N}(v_i)$ is the neighbors of node v_i , $\deg(v_i)$ is the node degree of v_i .

Fig. 1 illustrates Ollivier-Ricci curvature (for nodes) for molecular structure analysis. It can be seen that positive ORCs appear in densely-connection regions, while negative ORCs are found in link or bottleneck regions. The “over-squashing” issue tends to happen at these narrow bottleneck regions, where the transportation of information tends to be distorted.

3.3. Curvature graph convolutional network

In our CGCN model, the ORC information is incorporated into message passing process by using ORC-related edge weights. To alleviate the “over-squashing” effects that usually happen at regions with negative ORC values, we propose an edge weight function that is inversely related to edge ORC values. More specifically, for edge $e_{v_i v_j}$ with ORC $c(v_i, v_j)$, we define an ORC-related vector,

$$g_c(v_i, v_j) = \left(\frac{1 + e^{-c(v_i, v_j)}}{2}, \frac{1 + e^{-2c(v_i, v_j)}}{2}, \dots, \frac{1 + e^{-n_c c(v_i, v_j)}}{2} \right)^T, \quad (6)$$

here n_c is a positive integer and represents the dimension of $g_c(v_i, v_j)$. The edge weight function is defined as follows,

$$f_c(v_i, v_j) = \sigma(w_{\text{MLP}} \cdot g_c(v_i, v_j) + b) \quad (7)$$

here $\sigma(\cdot)$ represents activate function, w_{MLP} is an weight vector with size $n_c \times 1$, and b is the bias. Essentially, we use an MLP to learn the edge weight function from the ORC-related vector.

The weight function is then incorporated into the message-passing process, in which node representation is updated by the aggregation of node features from all its neighbors. In our CGCN, the contribution from neighboring node features is not aggregated with equal weight, instead it is scaled by the edge weighted function as follows,

$$h_v^{(l+1)} = \sigma \left(\sum_{v_j \in \mathcal{N}(v) \cup \{v\}} \frac{1}{\sqrt{\deg(v)} \sqrt{\deg(v_j)}} f_c(v, v_j) W_{\text{GCN}} h_{v_j}^{(l)} \right) \quad (8)$$

where $\mathcal{N}(v)$ is the neighbors of node v , $\deg(v)$ represents the degree of node v , $h_v^{(l+1)}$ and $h_{v_j}^{(l)}$ are the node features of v and v_j after $l + 1$ and l message-passing iterations respectively. Fig. 2 illustrates the ORC-based message-passing process in our CGCN model.

After the message-passing process, node representations h_{v_i} and h_{v_j} are obtained for nodes v_i and v_j . The probability that there is an interaction between two nodes v_i and v_j can be evaluated through a MLP and a hidden layer based score function as follows,

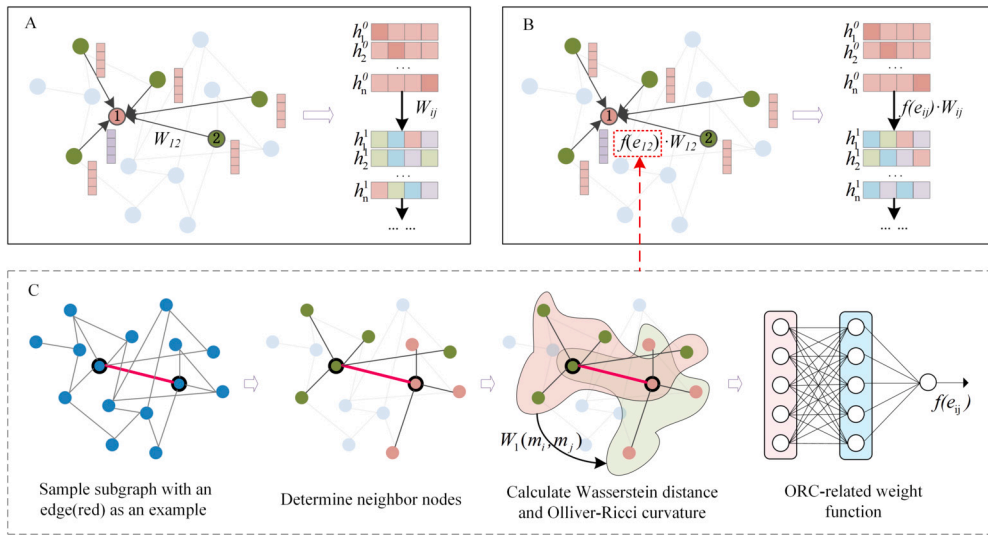


Fig. 2. An illustration of the ORC-based feature aggregation in message-passing procedure. **A** A normal message-passing procedure, in which node feature representation is updated by using all feature vectors from its neighboring nodes with the same weight. **B** ORC-based message-passing procedure in our CGCN model. The neighboring feature vectors are aggregated with ORC-related weights. **C** The calculation of ORCs and ORC-related weights. The edge ORC is calculated by using the Wasserstein distance between two probability distributions defined on neighboring nodes. The neighbors of the centering green node (enclosed by a black circle) are the other green nodes together with the centering red node (enclosed by a black circle), while the neighbors of the centering red node are the other red nodes together with the green centering node. The ORC-related weight is calculated through a MLP.

$$\hat{p}_{v_i v_j} = \text{MLP} \left(\left\| \left(h_{v_i} + h_{v_j}, h_{v_i} \odot h_{v_j}, h_{v_i}, h_{v_j} \right) \right\| \right), \quad (9)$$

where \odot is element-wise product, $\|(\cdot)$ means the concatenation and the output is a vector, MLP is a multi-layer perceptron, and $\hat{p}_{v_i v_j}$ is the prediction of the relationship between two nodes v_i and v_j .

4. Results and discussions

4.1. Datasets and model setup

Datasets In this study, two types of graph datasets, i.e., 14 real-world graph datasets and 77 simulated graph datasets, are employed. The 14 datasets include ChCh-Miner, ChG-Miner, DCh-Miner, PPT-Ohmnet, DG-AssocMiner, HuRI-PPI, PP-Decagon, PP-Pathways, CPI_human, CPI_celegans, Drugbank_DTI, Drugbank_DDI, AdverseDDI, and DisGeNET. These datasets cover various types of biomolecular interaction, including drug-drug interaction networks (ChCh-Miner and Drugbank_DDI, AdverseDDI), drug-gene interaction networks (ChG-Miner), disease-drug interaction networks (DCh-Miner), protein-protein interaction networks (PPT-Ohmnet, HuRI-PPI, PP-Decagon, and PP-Pathways), disease-gene interaction networks (DG-AssocMiner and DisGeNET), compound-protein interaction network (CPI_human and CPI_celegans), and drug-target interaction network (Drugbank_DTI). Among them, ChCh-Miner, ChG-Miner, DCh-Miner, PPT-Ohmnet, DG-AssocMiner, and HuRI-PPI are obtained from Ref [65]. PP-Decagon and PP-Pathways are from Ref [66]. CPI_human and CPI_celegans are from Ref [67]. Drugbank_DTI, Drugbank_DDI, and AdverseDDI are from Ref [68]. AdverseDDI is taken from Ref [69] and DisGeNET is from Ref [70]. The details of these 14 networks are shown in Table 1, including the number of nodes and edges, average degree, density, and ratio of positive and negative curvature.

Baselines. We compare our CGCN model with 8 state-of-the-art methods, which can be categorized into two classes, i.e., GNN models and network embedding models. Four GNN models are considered, including Graph Convolutional Network (GCN) [18], Graph Attention Network (GAT) [19], CSGNN [54] and SkipGNN [52]. Network embedding models are to represent a high-dimensional, sparse vector space with a low-dimensional, dense vector space. They are widely used for network representation learning. Four classical network embedding

Table 1
Statistics for 14 biomolecular interaction networks.

Datasets	Nodes	Edges	Degree	Density	Ratio ^a
ChCh-Miner [65]	1514	48514	64.09	4.24%	1.66
ChG-Miner [65]	7341	15138	4.12	0.06%	0.61
DCh-Miner [65]	7197	466656	129.65	1.80%	2.33
PPT-Ohmnet [65]	4510	70338	31.19	0.69%	0.20
DG-AssocMiner [65]	7813	21357	5.47	0.07%	0.24
PP-Decagon [66]	19081	715612	75.01	0.39%	0.50
PP-Pathways [66]	21557	342353	31.76	0.15%	0.19
HuRI-PPI [65]	5604	23322	8.32	0.15%	0.18
CPI_human [67]	2013	2633	2.62	0.13%	1.21
CPI_celegans [67]	1782	2659	2.98	0.17%	0.94
Drugbank_DTI [68]	12566	18866	3.00	0.02%	0.98
Drugbank_DDI [68]	1977	563438	569.99	28.85%	383.79
AdverseDDI [69]	393	12480	63.51	16.20%	87.24
DisGeNET [70]	19783	81746	8.26	0.04%	0.20

^a The ratios between positive ORCs and negative ORCs.

models are selected, including DeepWalk [71], LINE [72], Node2Vec [73] and SDNE [74]. For the graph neural network methods GSGNN and SkipGNN, we use the default parameter settings of the original paper. We use a two-layer neural network to run GCN and GAT methods. The codes of the other four network embedding models (DeepWalk, LINE, Node2Vec and SDNE) are also adopted from the original paper, and we use the default parameters. The number of layers in our CGCN is two.

Implementation details. In this study, we randomly select as many negative samples as there are positive samples, and the whole data set was divided into training set, validation set and test set according to the ratio of 7:1:2. The area under the receiver operating characteristic curve (AUC) and the area under the precision-recall curve (AUPR) are used to evaluate the performance of model. The initial vector of each node of the CGCN model is one-hot encoding. We run each test 10 times, and use the average values as final results. For the simulated network, in order to have a more reasonable topological structure, five disjoint communities are created. The probability of node connection within the community is p , and the probability of node connection between communities is q . Three types of tests are conducted. First, we fix the node number to be 1000 and systematically change p and q to generate a series of networks with various ratios of positive to negative

Table 2

The comparison between CGCN and four GNN methods.

Datasets	GCN		GAT		CSGNN		SkipGNN		CGCN	
	AUC	AUPR	AUC	AUPR	AUC	AUPR	AUC	AUPR	AUC	AUPR
ChCh-Miner [65]	0.8984	0.8791	0.8786	0.8502	0.9350	0.9210	0.8819	0.8594	0.9426	0.9329
ChG-Miner [65]	0.9352	0.9409	0.9514	0.9499	0.9258	0.9307	0.9526	0.9524	0.9644	0.9627
DCh-Miner [65]	0.9966	0.9961	0.9966	0.9959	0.9914	0.9903	0.8446	0.8606	0.9972	0.9967
PPT-Ohmnet [65]	0.8937	0.8988	0.8798	0.8806	0.9031	0.9055	0.8091	0.7896	0.9143	0.9174
DG-AssocMiner [65]	0.9930	0.9906	0.9936	0.9916	0.9919	0.9896	0.8585	0.6679	0.9945	0.9925
PP-Decagon [66]	0.9138	0.9126	0.8836	0.8740	NA ^a	NA ^a	0.8892	0.8819	0.9397	0.9402
PP-Pathways [66]	0.9394	0.9370	0.9225	0.9177	NA ^a	NA ^a	0.9263	0.9228	0.9487	0.9453
HuRI-PPI [65]	0.9164	0.9189	0.8994	0.8965	0.9228	0.9269	0.9119	0.9182	0.9327	0.9333
CPI_human [67]	0.9423	0.9554	0.9578	0.9653	0.9696	0.9708	0.6232	0.6245	0.9738	0.9770
CPI_celegans [67]	0.9552	0.9661	0.9722	0.9774	0.9839	0.9852	0.7217	0.6995	0.9886	0.9891
Drugbank_DTI [68]	0.9234	0.9371	0.9476	0.9533	0.9737	0.9730	0.8946	0.6764	0.9750	0.9730
Drugbank_DDI [68]	0.9009	0.8949	0.9448	0.9514	0.9537	0.9495	0.8144	0.7772	0.9655	0.9678
AdverseDDI [69]	0.9492	0.9445	0.9381	0.9325	0.9540	0.9508	0.8450	0.7610	0.9466	0.9411
DisGeNET [70]	0.9723	0.9785	0.9829	0.9849	0.9869	0.9880	0.9145	0.9271	0.9895	0.9901

^a NA indicates that the CSGCN model requires too much memory on the PP-Decagon and PP-Pathways datasets.**Table 3**

The comparison between CGCN and four network embedding methods.

Datasets	CGCN		SDNE		Node2Vec		LINE		DeepWalk	
	AUC	AUPR	AUC	AUPR	AUC	AUPR	AUC	AUPR	AUC	AUPR
ChCh-Miner [65]	0.8560	0.8375	0.8668	0.8431	0.8436	0.8424	0.6881	0.6736	0.9426	0.9329
ChG-Miner [65]	0.6108	0.6114	0.9144	0.8943	0.7312	0.7354	0.7623	0.8159	0.9644	0.9627
DCh-Miner [65]	0.7769	0.7889	0.8020	0.8077	0.7494	0.7461	0.6279	0.6243	0.9972	0.9967
PPT-Ohmnet [65]	0.8652	0.8694	0.7608	0.7675	0.7118	0.7411	0.6274	0.6409	0.9143	0.9174
DG-AssocMiner [65]	0.5831	0.5797	0.8461	0.8272	0.6277	0.6323	0.7011	0.7201	0.9945	0.9925
PP-Decagon [66]	0.8812	0.8810	0.8309	0.8306	0.8159	0.8258	0.6279	0.6216	0.9397	0.9402
PP-Pathways [66]	0.9115	0.9116	0.7678	0.7786	0.8253	0.8283	0.6300	0.6280	0.9487	0.9453
HuRI-PPI [65]	0.9243	0.9324	0.8286	0.8300	0.7179	0.7400	0.6707	0.6866	0.9327	0.9333
CPI_human [67]	0.9613	0.9714	0.9523	0.9441	0.7600	0.7798	0.8620	0.8876	0.9738	0.9770
CPI_celegans [67]	0.9793	0.9826	0.9706	0.9697	0.8331	0.8558	0.8388	0.8303	0.9886	0.9891
Drugbank_DTI [68]	0.7109	0.6988	0.9634	0.9481	0.5725	0.6037	0.8351	0.8819	0.9750	0.9730
Drugbank_DDI [68]	0.8048	0.7776	0.8085	0.7804	0.7785	0.7523	0.7265	0.6926	0.9655	0.9678
AdverseDDI [69]	0.8945	0.8630	0.8954	0.8810	0.8758	0.8482	0.7829	0.7460	0.9466	0.9411
DisGeNET [70]	0.6831	0.6520	0.8821	0.8725	0.6801	0.6688	0.6995	0.7210	0.9895	0.9901

ORCs. Second, we systematically change the number of nodes from 200 to 20,000 (while keeping p and q to be 0.1 and 0.0001, respectively). Third, we fix the number of nodes to be 400 and 4000, and systematically change the network density, i.e., the ratio of edge number to the number of all possible edges (in a complete graph).

We use batch size 128 with Adam optimizer of learning rate 5e-4 and run CGCN model in PyTorch. For training, we use a server with 2 Intel(R) Core (TM) I9-10900X 3.70 GHz CPUs, 64 GB RAM and 2 NVIDIA GeForce RTX 2070 GPUs. For more detailed parameter introduction of the model, please refer to the source code.¹

4.2. CGCN for biomedical network analysis

In this section, we conduct tests to compare CGCN with all the baselines on 14 real-world biomedical datasets. The AUC and AUPR of various methods in link prediction tasks are shown in Table 2 and Table 3. Table 2 lists the results of graph neural network methods and Table 3 lists the results of network embedding methods. Our CGCN model performs well on most datasets, achieving the best predictive performance on 13 of 14 datasets. Although these 14 datasets differ greatly in terms of network size, average node degree, density, and ratio of positive to negative curvature, our CGCN shows a consistent good performance. In particular, our CGCN model demonstrates great superiority on datasets of ChCh-Miner, ChG-Miner, Drugbank_DTI and Drugbank_DDI, in which the results of CGCN are better than GCN model

by 4.9%, 3.1%, 4.8% and 7.2% respectively. Although on AdverseDDI dataset, the performance of our CGCN is not the best, it is only inferior to CSGCN and better than all other models. In general, CGCN shows strong superiority in comparison with both graph neural network models and network embedding methods.

4.3. CGCN performance analysis

In order to further verify the performance of CGCN under various datasets and analyze the limitations of CGCN, we design multiple test cases based on three types of graph properties, including the ratio between positive and negative curvature, network size, and network density. The networks are generated by using the probability of node connection within the community (p) and the probability of node connection between communities (q). AUC is used as the metric for the evaluation of the performance. We systematically compare our CGCN model with GCN model [18]. The difference between the AUCs in three types of graph property tests is calculated and the results are illustrated in Fig. 3. Note that y-axis represents the difference between the AUCs of CGCN and GCN in all four subfigures, i.e., $AUC_{CGCN} - AUC_{GCN}$.

First, we analyze the effect of ratios between positive and negative curvature. We generated 34 random networks with a relatively continuous distribution of positive-to-negative-curvature ratios ranging from 0.004 to more than 600. The results are shown in Fig. 3A. It can be seen that no matter what the positive-to-negative ratios are, our CGCN performance is always better than that of GCN model, which shows that the performance of CGCN is robust to positive-to-negative ratios. In particular, graphs with small positive-to-negative ratios usually have a network topology close to a tree, while large positive-to-negative ra-

¹ A reference implementation of CGCN may be found at <https://github.com/CS-BIO/CGCN>.

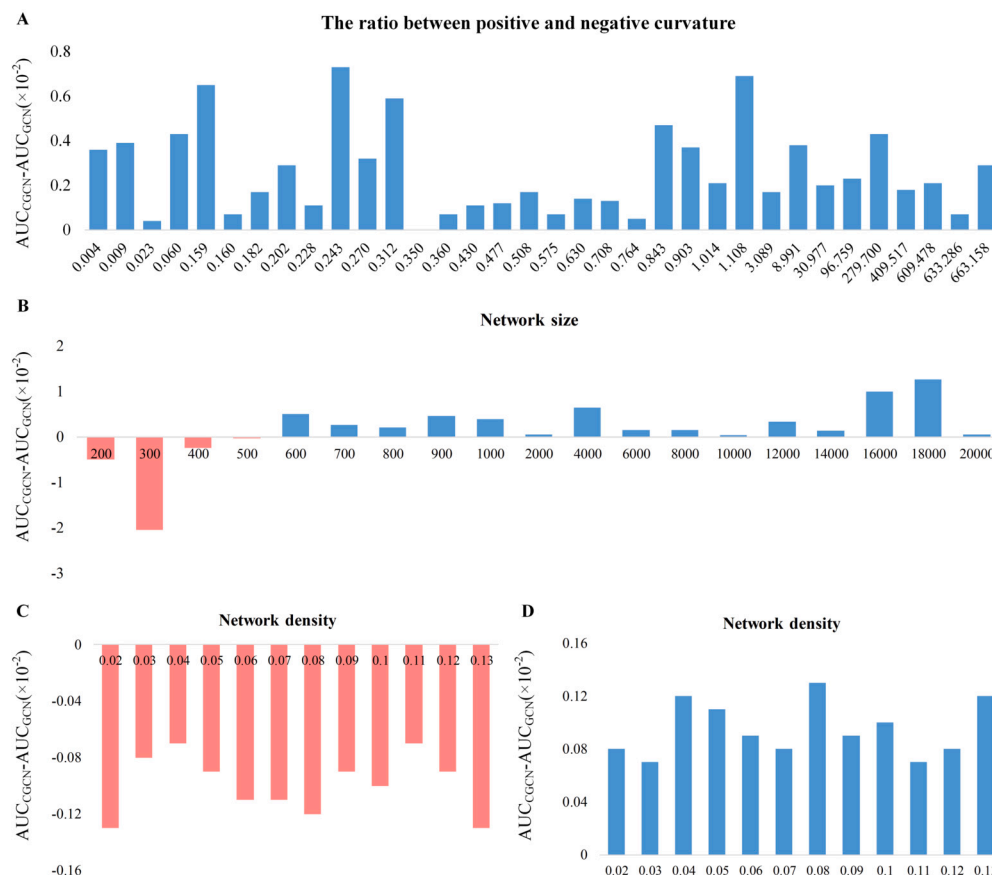


Fig. 3. The comparison of the results from CGCN and traditional GCN on simulated datasets. **A** Performance comparison between CGCN and GCN on different ratios of positive to negative curvatures. **B** Performance comparison between CGCN and GCN on different network sizes. **C** Performance comparison between CGCN and GCN on small-size (400) networks with different network densities. **D** Performance comparison between CGCN and GCN on large-size (4000) networks with different network densities.

tios are associated with complete graphs. The better performance of our CGCN indicates that it is suitable for all kinds of network topologies.

Second, we explore the impact of network sizes on model performance. We set the number of nodes in the simulated network to increase from 200 to 20,000 sequentially. The performance of CGCN and GCN models is shown in Fig. 3B. It can be seen that when the number of nodes in the network is greater than 500, the performance of CGCN is better than that of GCN. When the number of nodes is less than 500, CGCN cannot show obvious superiority, as indicated by the red bars. This indicates that our CGCN model is more suitable for large-sized networks, i.e., nodes larger than 500. These results are consistent with the ones from 14 real-world biomolecular datasets. In fact, our CGCN model is only inferior to traditional GCN model on the AdverseDDI dataset, whose number of nodes is 393 (<500).

Third, we analyze the impact of network densities on model performance. Fig. 3C and 3D show the prediction performance of the CGCN model under different network densities when the number of network nodes is 400 and 4000, respectively. It can be seen that the AUC value of the CGCN model is consistently smaller than that of the GCN in the network with 400 nodes and consistently larger than that of the GCN in the network with 4000 nodes. The results show that our CGCN model is relatively stable under different network densities, and the performance of our CGCN is more related to graph sizes.

Fourth, we analyze the impact of the ORC-based vector in Eq. (6) on the performance of CGCN. In particular, we systematically change parameter n_c from 1 to 10 and study the performance of our CGCN models. The results for the 14 biomolecular interaction datasets under different n_c values are displayed in Fig. 4 A. The average values of AUC for all 14 systems are illustrated in Fig. 4 B. It can be seen that

even though for some cases, there are fluctuations of AUC values, in general, with the increase of parameter n_c , the performance will slightly increase.

4.4. CGCN for representation learning

In this section, we explore the capabilities of the CGCN model in terms of representation learning. We extract the representation vectors of each node in the test datasets and use t-SNE [75] to project the high-dimensional representation vectors into 2D space. The whole datasets are considered. Our CGCN model is compared with four network embedding models (DeepWalk, LINE, Node2Vec and SDNE), which usually are used to network representation learning in various tasks. The results are shown in Fig. 5 and 6, in which the red and blue points represent node pairs without link relationship and node pairs with link relationship, respectively. It can be seen from the results that the proposed CGCN model is significantly better than the other four network embedding methods in distinguishing node pairs with links and node pairs without links. Quantitatively, in terms of the Davies Bouldin index (DBI) [76] (the smaller, the better), which is a metric to evaluate the clustering results, CGCN clearly outperforms other models. These result shows that the CGCN model has a good capability in representation learning.

5. Conclusions

The proper incorporation of geometric information into deep learning architectures plays a key role in geometric deep learning models. As one of the fundamental concepts in different geometry, Ricci curvature characterizes the intrinsic properties of manifold surfaces. The

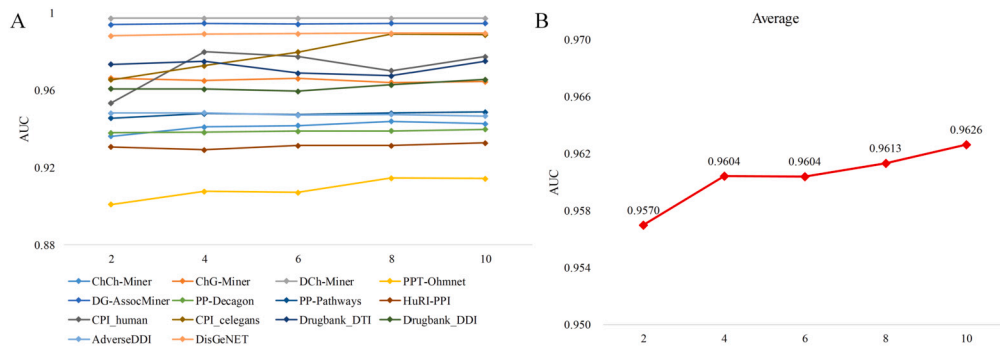


Fig. 4. Effect of hyperparameter n_c . **A** Effect of hyperparameter n_c on different systems. **B** Average performance of the 14 datasets under different n_c values.

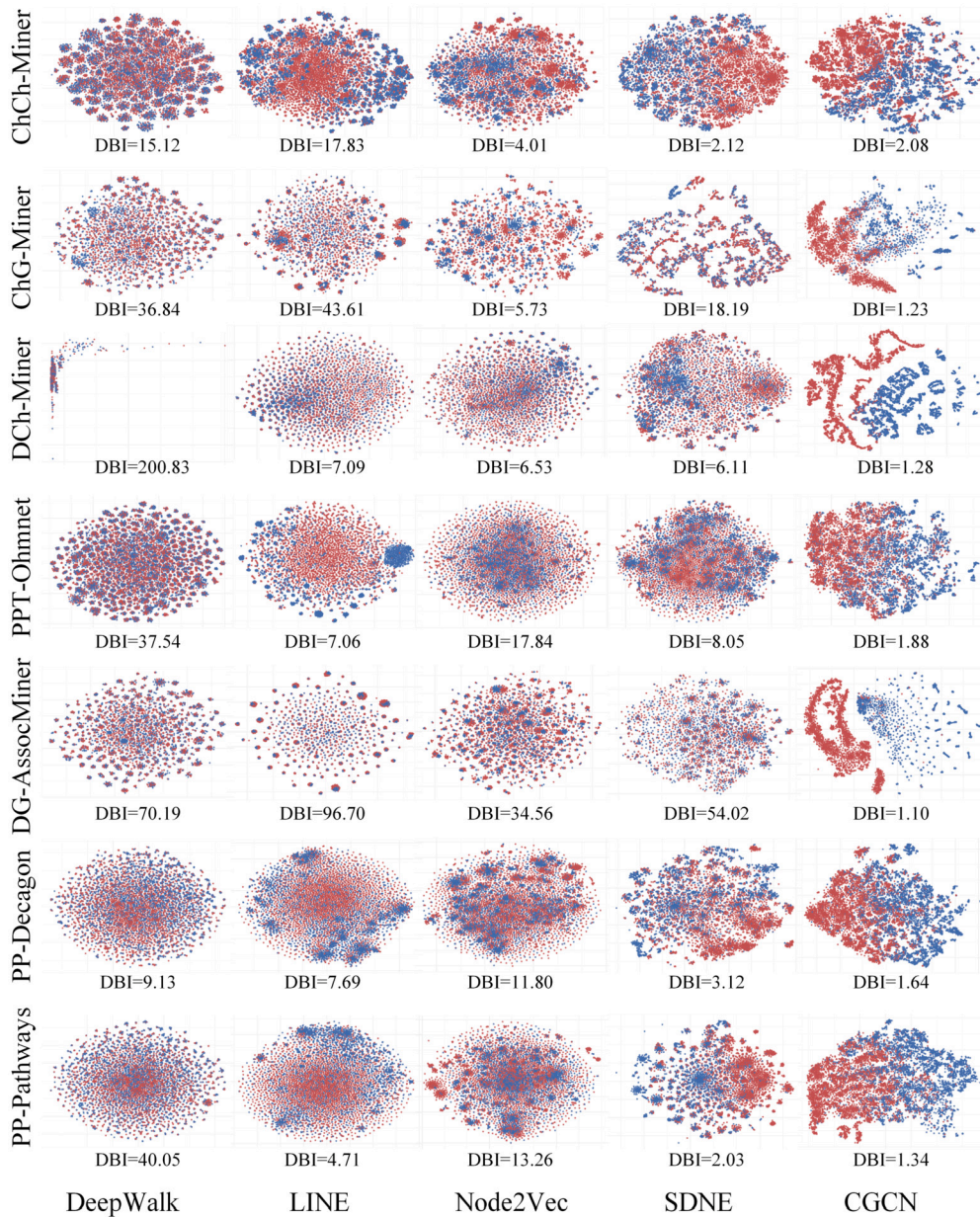


Fig. 5. The performance of representation learning on ChCh-Miner, ChG-Miner, DCh-Miner, PPT-Ohmnet, DG-AssocMiner, PP-Decagon and PP-Pathways network datasets. The representation vectors of each node in the test datasets are projected into 2D spaces by t-SNE. The red and blue points represent node pairs without link relationship and node pairs with link relationship, respectively. Four network embedding methods are considered in our comparison.

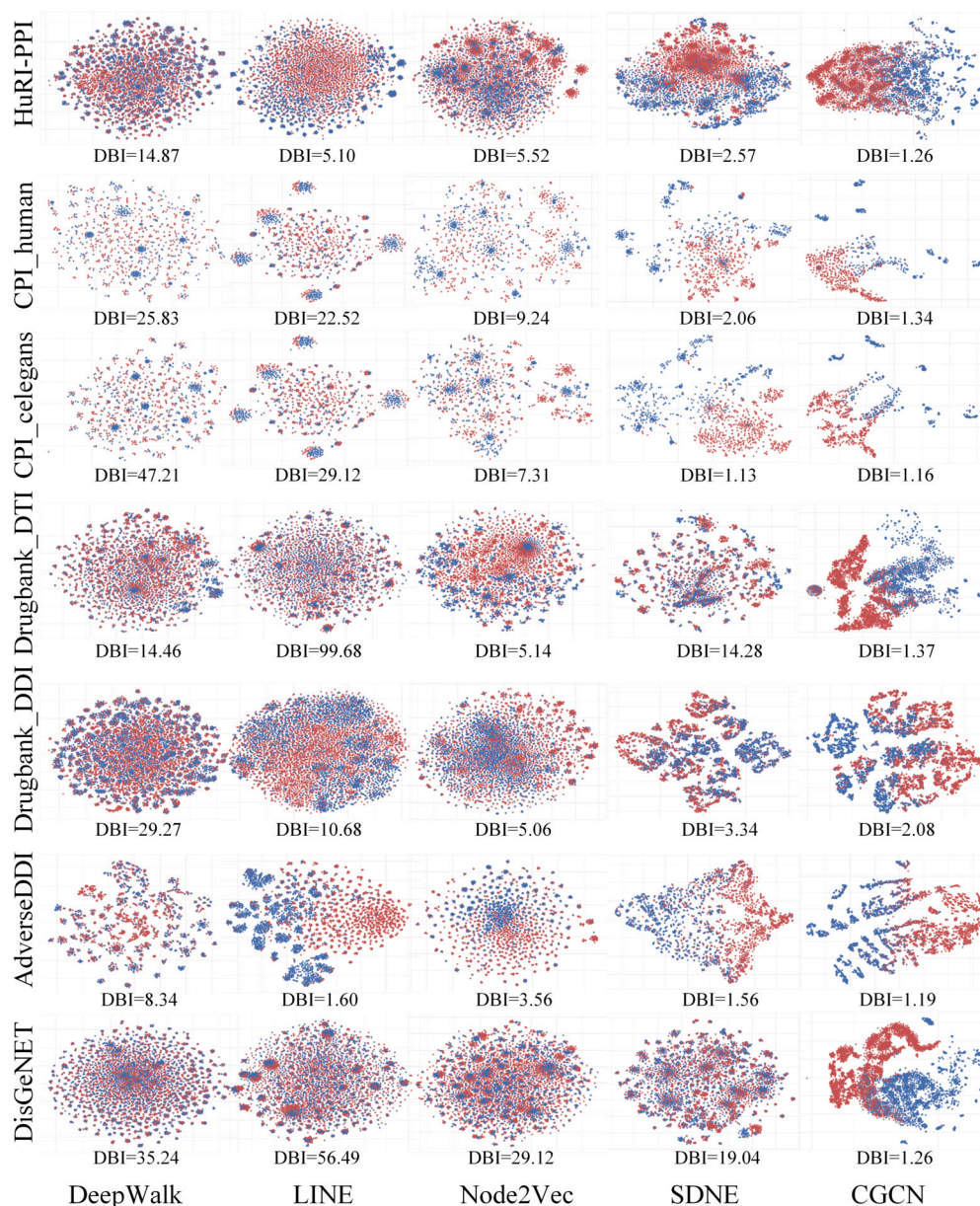


Fig. 6. The performance of representation learning on HuRI-PPI, CPI_human, CPI_celegans, Drugbank_DTI, Drugbank_DDI, AdverseDDI and DisGeNET network datasets. The representation vectors of each node in the test datasets are projected into 2D spaces by t-SNE. The red and blue points represent node pairs without link relationship and node pairs with link relationship, respectively. Four network embedding methods are considered in our comparison.

discrete Ricci curvatures have found various applications in network and graph data analysis. In particular, they have been used in the characterization of “over-squashing” phenomenon. In this paper, we propose a curvature-enhanced graph convolutional network (CGCN) to incorporate the Ollivier-Ricci curvature (ORC) information into node feature aggregation process. With a better characterization of local topological structures through ORCs, our CGCN model has a more efficient message-passing operator. Experimental results show that the proposed model outperforms the competitive methods in 13 out of 14 real-world biomedical datasets and ranks as second in the rest one. In the simulated tests, our CGCN model is superior to the traditional GCN model regardless of the positive-to-negative-curvature ratios, network densities, and network sizes (when larger than 500).

Declaration of competing interest

We declare that there is no conflict of interest or competing interest.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC grant nos. 61873089, 62032007), Nanyang Technological University SPMS Collaborative Research Award 2022, Singapore Ministry of Education Academic Research fund (RG16/23, MOE-T2EP20120-0013, MOE-T2EP20220-0010, MOE-T2EP20221-0003) and China Scholarship Council (CSC grant no. 202006130147).

References

- [1] Malafaya PB, Silva GA, Reis RL. Natural-origin polymers as carriers and scaffolds for biomolecules and cell delivery in tissue engineering applications. *Adv Drug Deliv Rev* 2007;59(4–5):207–33.
- [2] Chen F-M, Liu X. Advancing biomaterials of human origin for tissue engineering. *Prog Polym Sci* 2016;53:86–168.
- [3] Jang J, Park JY, Gao G, Cho D-W. Biomaterials-based 3d cell printing for next-generation therapeutics and diagnostics. *Biomaterials* 2018;156:88–106.

- [4] El Deeb S, Al-Harrasi A, Khan A, Al-Broumi M, Al-Thani G, Alomairi M, et al. Microscale thermophoresis as a powerful growing analytical technique for the investigation of biomolecular interaction and the determination of binding parameters. *Methods Appl Fluoresc* 2022;10(4):042001.
- [5] Hell SW. Far-field optical nanoscopy. *Science* 2007;316(5828):1153–8.
- [6] Moll D, Prinz A, Gesellchen F, Drewianka S, Zimmermann B, Herberg FW. Biomolecular interaction analysis in functional proteomics. *J Neural Transm* 2006;113:1015–32.
- [7] Wang Y, Wang W, Liang Y, Cai Y, Liu J, Hooi B. Nodeaug: semi-supervised node classification with data augmentation. In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*; 2020. p. 207–17.
- [8] Zhang C, Song D, Huang C, Swami A, Chawla NV. Heterogeneous graph neural network. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*; 2019. p. 793–803.
- [9] Zhang Z, Cai J, Zhang Y, Wang J. Learning hierarchy-aware knowledge graph embeddings for link prediction. In: *Proceedings of the AAAI conference on artificial intelligence*, vol. 34. 2020. p. 3065–72.
- [10] Wu J, He J, Xu J. Net: degree-specific graph neural networks for node and graph classification. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*; 2019. p. 406–15.
- [11] Zhang M, Cui Z, Neumann M, Chen Y. An end-to-end deep learning architecture for graph classification. In: *Proceedings of the AAAI conference on artificial intelligence*, vol. 32. 2018.
- [12] Jiang D, Wu Z, Hsieh C-Y, Chen G, Liao B, Wang Z, et al. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J Cheminform* 2021;13(1):1–23.
- [13] Li X, Yan X, Gu Q, Zhou H, Wu D, Xu J. Deepchemstable: chemical stability prediction with an attention-based graph convolution network. *J Chem Inf Model* 2019;59(3):1044–9.
- [14] Feinberg EN, Sur D, Wu Z, Husic BE, Mai H, Li Y, et al. PotentialNet for molecular property prediction. *ACS Cent Sci* 2018;4(11):1520–30.
- [15] Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY. A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 2020;32(1):4–24.
- [16] Zhang S, Tong H, Xu J, Maciejewski R. Graph convolutional networks: a comprehensive review. *Comput Soc Netw* 2019;6(1):1–23.
- [17] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, et al. Graph neural networks: a review of methods and applications. *AI Open* 2020;1:57–81.
- [18] Welling M, Kipf TN. Semi-supervised classification with graph convolutional networks. In: *International conference on learning representations (ICLR 2017)*; 2016.
- [19] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [20] Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks. In: *ICLR*; 2019.
- [21] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. *Adv Neural Inf Process Syst* 2017;30.
- [22] Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, et al. Convolutional networks on graphs for learning molecular fingerprints. In: *Advances in neural information processing systems*; 2015. p. 2224–32.
- [23] Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Process Mag* 2017;34(4):18–42.
- [24] Bronstein MM, Bruna J, Cohen T, Veličković P. Geometric deep learning: grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.
- [25] Atz K, Grisoni F, Schneider G. Geometric deep learning on molecular representations. *Nat Mach Intell* 2021;3(12):1023–32.
- [26] Jost J, Jost J. *Riemannian geometry and geometric analysis*, vol. 42005. Springer; 2008.
- [27] Najman L, Romon P. *Modern approaches to discrete curvature*, vol. 2184. Springer; 2017.
- [28] Sturm K-T, et al. On the geometry of metric measure spaces. *Acta Math* 2006;196(1):65–131.
- [29] Ollivier Y. Ricci curvature of Markov chains on metric spaces. *J Funct Anal* 2009;256(3):810–64.
- [30] Bonciocat A-I, Sturm K-T. Mass transportation and rough curvature bounds for discrete spaces. *J Funct Anal* 2009;256(9):2944–66.
- [31] Sreejith R, Mohanraj K, Jost J, Saucan E, Samal A. Forman curvature for complex networks. *J Stat Mech Theory Exp* 2016;2016(6):063206.
- [32] Samal A, Sreejith R, Gu J, Liu S, Saucan E, Jost J. Comparative analysis of two discretizations of Ricci curvature for complex networks. *Sci Rep* 2018;8(1):1–16.
- [33] Saucan E, Weber M. Forman's Ricci curvature from networks to hypernetworks. In: *International conference on complex networks and their applications*. Springer; 2018. p. 706–17.
- [34] Ni C-C, Lin Y-Y, Gao J, Gu XD, Saucan E. Ricci curvature of the Internet topology. In: *2015 IEEE conference on computer communications (INFOCOM)*. IEEE; 2015. p. 2758–66.
- [35] Ni C-C, Lin Y-Y, Luo F, Gao J. Community detection on networks with Ricci flow. *Sci Rep* 2019;9(1):1–12.
- [36] Sia J, Jonckheere E, Bogdan P. Ollivier-Ricci curvature-based method to community detection in complex networks. *Sci Rep* 2019;9(1):1–12.
- [37] Sandhu RS, Georgiou TT, Tannenbaum AR. Ricci curvature: an economic indicator for market fragility and systemic risk. *Sci Adv* 2016;2(5):e1501495.
- [38] Sandhu R, Georgiou T, Reznik E, Zhu L, Kolesov I, Senbabaoglu Y, et al. Graph curvature for differentiating cancer networks. *Sci Rep* 2015;5(1):1–13.
- [39] Farooq H, Chen Y, Georgiou TT, Tannenbaum A, Lenglet C. Network curvature as a hallmark of brain structural connectivity. *Nat Commun* 2019;10(1):1–11.
- [40] Wee J, Xia K. Forman persistent Ricci curvature (FPRC)-based machine learning models for protein-ligand binding affinity prediction. *Brief Bioinform* 2021;22(6):bbab136.
- [41] Wee J, Xia K. Ollivier persistent Ricci curvature-based machine learning for the protein-ligand binding affinity prediction. *J Chem Inf Model* 2021;61(4):1617–26.
- [42] Topping J, Di Giovanni F, Chamberlain BP, Dong X, Bronstein MM. Understanding over-squashing and bottlenecks on graphs via curvature. In: *International conference on learning representations (ICLR 2022)*; 2021.
- [43] Murgas KA, Saucan E, Sandhu R. Quantifying cellular pluripotency and pathway robustness through Forman-Ricci curvature. In: *Complex networks & their applications X: volume 2, proceedings of the tenth international conference on complex networks and their applications COMPLEX NETWORKS 2021* 10. Springer; 2022. p. 616–28.
- [44] Ye Z, Liu KS, Ma T, Gao J, Chen C. Curvature graph network. In: *International conference on learning representations*; 2019.
- [45] Li H, Cao J, Zhu J, Liu Y, Zhu Q, Wu G. Curvature graph neural network. *Inf Sci* 2022;592:50–66.
- [46] Glass S, Spasov S, Liò P. RicciNets: Curvature-guided pruning of high-performance neural networks using Ricci flow. 2020.
- [47] Li J, Fu X, Sun Q, Ji C, Tan J, Wu J, et al. Curvature graph generative adversarial networks. In: *Proceedings of the ACM web conference 2022*; 2022. p. 1528–37.
- [48] Yang M, Zhou M, Pan L, King I. Hyperbolic curvature graph neural network. *arXiv preprint arXiv:2212.01793*, 2022.
- [49] New A, Pekala MJ, Le NQ, Domenico J, Piatko CD, Stiles CD. Curvature-informed multi-task learning for graph networks. In: *ICML 2022 2nd AI for science workshop*; 2022.
- [50] Wang S, Wei X, Nogueira dos Santos CN, Wang Z, Nallapati R, Arnold A, et al. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In: *Proceedings of the web conference 2021*; 2021. p. 1761–71.
- [51] Fu X, Li J, Wu J, Sun Q, Ji C, Wang S, et al. ACE-HGNN: adaptive curvature exploration hyperbolic graph neural network. In: *2021 IEEE international conference on data mining (ICDM)*. IEEE; 2021. p. 111–20.
- [52] Huang K, Xiao C, Glass LM, Zitnik M, Sun J. SkipGNN: predicting molecular interactions with skip-graph networks. *Sci Rep* 2020;10(1):1–16.
- [53] Xu N, Wang P, Chen L, Tao J, Zhao J. MR-GNN: multi-resolution and dual graph neural network for predicting structured entity interactions. In: *Proceedings of the twenty-eighth international joint conference on artificial intelligence*; 2019. p. 3968–74.
- [54] Zhao C, Liu S, Huang F, Liu S, Zhang W. CSGNN: contrastive self-supervised graph neural network for molecular interaction prediction. In: *IJCAI*; 2021. p. 3756–63.
- [55] Lin X, Quan Z, Wang Z-J, Ma T, Zeng X. KGN: knowledge graph neural network for drug-drug interaction prediction. In: *IJCAI*, vol. 380. 2020. p. 2739–45.
- [56] Wang Y, Min Y, Chen X, Wu J. Multi-view graph contrastive representation learning for drug-drug interaction prediction. In: *Proceedings of the web conference 2021*; 2021. p. 2921–33.
- [57] Ye Q, Hsieh C-Y, Yang Z, Kang Y, Chen J, Cao D, et al. A unified drug-target interaction prediction framework based on knowledge graph and recommendation system. *Nat Commun* 2021;12(1):1–12.
- [58] Shen C, Luo J, Ouyang W, Ding P, Chen X. IDDKin: network-based influence deep diffusion model for enhancing prediction of kinase inhibitors. *Bioinformatics* 2020;36(22–23):5481–91.
- [59] Jiang D, Hsieh C-Y, Wu Z, Kang Y, Wang J, Wang E, et al. Interactiongraphnet: a novel and efficient deep graph representation learning framework for accurate protein-ligand interaction predictions. *J Med Chem* 2021;64(24):18209–32.
- [60] Ollivier Y. Ricci curvature of metric spaces. *C R Math* 2007;345(11):643–6.
- [61] Lott J, Villani C. Ricci curvature for metric-measure spaces via optimal transport. *Ann Math* 2009;903–91.
- [62] Forman R. Bochner's method for cell complexes and combinatorial Ricci curvature. *Discrete Comput Geom* 2003;29(3):323–74.
- [63] Lin Y, Lu L, Yau S-T. Ricci curvature of graphs. *Tohoku Math J (2)* 2011;63(4):605–27.
- [64] Lin Y, Yau S-T. Ricci curvature and eigenvalue estimate on locally finite graphs. *Math Res Lett* 2010;17(2):343–56.
- [65] Luck K, Kim D-K, Lambourne L, Spirohn K, Begg BE, Bian W, et al. A reference map of the human binary protein interactome. *Nature* 2020;580(7803):402–8.
- [66] Marinka Z, Rok S, Sagar M, Jure L. BioSNAP datasets Stanford biomedical network dataset collection; Aug. 2018.
- [67] Tsubaki M, Tomii K, Sese J. Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics* 2019;35(2):309–18.
- [68] Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res* 2018;46(D1):D1074–82.
- [69] Zhu J, Liu Y, Zhang Y, Chen Z, Wu X. Multi-attribute discriminative representation learning for prediction of adverse drug-drug interaction. *IEEE Trans Pattern Anal Mach Intell* 2021.

- [70] Piñero J, Bravo À, Queralt-Rosinach N, Gutiérrez-Sacristán A, Deu-Pons J, Centeno E, et al. DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Res* 2016:gkw943.
- [71] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*; 2014. p. 701–10.
- [72] Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. Line: large-scale information network embedding. In: *Proceedings of the 24th international conference on world wide web*; 2015. p. 1067–77.
- [73] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*; 2016. p. 855–64.
- [74] Wang D, Cui P, Zhu W. Structural deep network embedding. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*; 2016. p. 1225–34.
- [75] Van der Maaten L, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;9(11).
- [76] Davies DL, Bouldin DW. A cluster separation measure. *IEEE Trans Pattern Anal Mach Intell* 1979;2:224–7.