Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

CICN: Higher-order link prediction with clustering mutual information of common neighbors

Yabing Yao [a,b,*], Ziyu Ti [a], Zhipeng Xu [a], Yangyang He [a], Zeguang Liu [c], Wenxiang Liu [b], Xiangzhen He [d,e], Fuzhong Nian [a], Jianxin Tang [a]

[a] School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
[b] School of Information Engineering, Gansu Minzu Normal University, Hezuo 747000, China
[c] Department of Information Technology, Qinghai Open University, Xining 810000, China
[d] Key Laboratory of Linguistic and Cultural Computing Ministry of Education, Northwest Minzu University, Lanzhou 730030, China
[e] Key Laboratory of Minzu Languages and Cultures Intelligent Information Processing, Northwest Minzu University, Lanzhou 730030, China

ARTICLE INFO

Dataset link: https://snap.stanford.edu/data/, http://networkrepository.com/networks.php

Keywords:
Complex networks
Higher-order link prediction
Higher-order clustering coefficient
Mutual information

ABSTRACT

In complex networks, traditional link prediction focuses on pairwise interactions between node pairs to predict missing links and identify spurious interactions, which has a wide range of applications in the real world. However, with the continuous expansion of network scale, interactions within the network occur not only between pairs of nodes but also involve higher-order interactions among multiple nodes. However, traditional link prediction methods face challenges in directly predicting these higher-order structures. In this paper, we propose a novel higher-order link prediction method based on **C**lustering mutual **I**nformation of **C**ommon **N**eighbors (CICN) for the prediction of 3-cliques (triangles). CICN employs node information entropy and calculates the impact of common neighbors by integrating different-order clustering coefficients to predict 3-cliques in the network. Experiments on 9 empirical networks show that the higher-order clustering patterns of nodes can significantly improve the accuracy of predicting 3-cliques. Additionally, we investigate the stability of the proposed algorithm and the results indicate that the performance of the CICN remains favorable across training sets of varying sizes. The source codes of our method are publicly available at: https://github.com/yabingyao/CICN4HigherOrderLinkPrediction.

1. Introduction

In recent years, the study of complex networks has attracted extensive attention from researchers in different fields. Real-world complex systems, like online social networks, transportation systems, and biological networks, consist of intricate relationships among numerous nodes [1,2]. As a core research area, link prediction in complex networks aims to predict unknown links or identify missing links by utilizing observed data in the network [3]. It is of great significance for understanding the evolution process of complex networks. Furthermore, link prediction has a wide range of applications in various fields. In drug discovery, predicting the interactions between drug components can help to guide the development of novel drugs [4,5]. In the recommendation systems, link prediction can recommend potentially interesting products to users based on their purchase records, which can effectively enhance the personalized user experience [6,7]. Additionally, in the field of network security, predicting potential links in a network can aid in detecting network attacks and anomalous behavior, thereby strengthening network security defenses [8].

The traditional link prediction problem revolves around pairwise interaction between two nodes in the network [9,10]. Nevertheless, mounting evidence indicates that real-world complex systems often exhibit collective behaviors involving multiple entities [11,12]. For example, in a collaboration network, collaborative relationships among authors occur not only between two individuals but also involve three or more people [13]. In biological networks, metabolic processes are often regulated by collaboration among multiple protein molecules [14]. The study of higher-order interactions has significantly advanced the dynamical analysis of network systems. By transcending the limitations of traditional pairwise interaction analysis, it offers novel perspectives on understanding complex networks. Investigating higher-order interactions can refine our modeling capabilities for complex systems, enabling a deeper understanding, analysis, and prediction of their dynamic behavior [15,16]. However, traditional link prediction focuses solely on predicting pairwise interactions between two nodes and cannot directly predict higher-order interactions in the network.

[*] Corresponding author.
E-mail addresses: yaoyabing@lut.edu.cn (Y. Yao), 222085404094@lut.deu.cn (Z. Ti).

https://doi.org/10.1016/j.jocs.2024.102513
Received 25 July 2024; Received in revised form 4 December 2024; Accepted 18 December 2024
Available online 26 December 2024

To predict higher-order interactions in the network, R. Benson et al. [17] propose pairwise link prediction that directly targets the prediction of triangle structures. Specifically, it predicts which nodes in the network may form triangle structures with a given seed edge. Chavan and Potika [18] introduce a triangle embedding-based algorithm, which combines node2vec [19], graph2vec [20], and graph neural network techniques, to address higher-order link prediction, specifically focusing on predicting the closure of open triangles. Papamarkou et al. [21] have conducted an in-depth exploration of higher-order concepts within topological deep learning (TDL), encompassing the evolution of higher-order network applications, a variety of methodological models and their potential applications. To address the issues of insufficient utilization of higher-order interactions and graph structural information in the heterogeneous graph link prediction task, the neighborhood overlap-aware heterogeneous hypergraph neural network (NOH) [22] combines the heterogeneous hypergraph variational autoencoder and the neighborhood overlap-aware graph neural network to learn higher-order semantics and utilize graph structural information. Traditional Graph Neural Networks (GNNs) are effective in handling graph data but have limitations in modeling higher-order interactions. Simplicial complexes (SCs), which can represent complex relationships among multiple entities, provide a novel framework for studying higher-order networks [23,24]. Liu et al. [25] propose two similarity measures, namely, simplex decomposition weight and closure ratio weight, based on local information to predict potential higher-order interactions in simplex networks. Existing higher-order link prediction methods primarily extend traditional link prediction algorithms, often neglecting the impact of higher-order clustering mutual information on the formation of higher-order network structures.

In this paper, we use cliques to represent higher-order structures in a network [26]. The probability of nodes forming higher-order cliques in a network can be assessed by the higher-order clustering structures. For example, nodes with higher 2-order clustering coefficients in the network are more likely to form 3-cliques (triangles) with neighboring nodes. Inspired by this, we explore a novel approach on the basis of pairwise link prediction and strive to solve the higher-order link prediction problem. Given a l-clique in the network, our goal is to identify which 1-cliques are most likely to interact with the given l-clique, and forming an $(l+1)$-clique. In this paper, we focus on the prediction of 3-cliques, as illustrated in Fig. 1(c). Specifically, given a 2-clique (edge) in the network, our objective is to identify which 1-cliques (nodes) are most likely to interact with this 2-clique, forming a 3-clique (triangle). To address this issue, we propose a higher-order link prediction with clustering mutual information of common neighbors (CICN), considering the higher-order clustering structures in the network, it amalgamates different higher-order clustering coefficients as conditional probabilities to estimate the probability of forming a 3-clique. Additionally, in order to determine the impacts of different common neighbors on 3-cliques, we employ the mutual information theory to distinguish the contributions of neighbors. The experimental results demonstrate that the CICN method is superior to other comparative algorithms. Moreover, the different order higher-order clustering coefficients of nodes can better improve the accuracy of higher-order link prediction algorithms. The contributions of this research are described as follows:

1. We propose a higher-order link prediction method with clustering mutual information of common neighbors. This method leverages node information entropy and combines different-order clustering coefficients to measure the contribution of common neighbors, which can accurately predict 3-cliques in the network.
2. We compare CICN with 13 benchmarks on 9 empirical networks, and the results show that higher-order clustering patterns of nodes can significantly improve the accuracy of predicting 3-cliques, especially in dense networks.
3. We propose a new dataset split method for 3-cliques prediction, and use negative sampling to address the imbalanced problem of positive and negative test samples in higher-order link prediction. We conduct experiments on 4 networks with different test sample ratios, validating the feasibility of our method.

The rest of this paper is organized as follows: Section 2 defines the higher-order link prediction problem studied in this paper. Section 3 elaborates on the CICN framework for higher-order link prediction that we proposed. Section 4 introduces the benchmarks and evaluation metrics. Section 5 presents the experimental results. Section 6 concludes the paper.

2. Problem definition

In traditional link prediction, given an unweighted undirected simple network $G = (V, E)$, where V represents the set of nodes and E represents the set of edges. To test the accuracy of the algorithm, the observed links in E are divided into a training set E^T and a test set E^P. Specifically, $E^T \cap E^P = \emptyset$ and $E^T \cup E^P = E$. For links that do not exist in E^T, the traditional link prediction task is to identify the links in the test set E^P as much as possible. For the higher-order link prediction in this paper, we describe the network as $G_n = (K_1, K_2, K_3, \dots, K_n)$, where K_1, K_2, K_3, …, K_n are the sets of 1-clique (node), 2-clique (edge), 3-clique (triangle), …, n-clique (polygon) in the network, n represents the highest order structure present in the network. Similarly, in order to test the accuracy of the higher-order algorithm, we divide the observed higher-order structure K_l into the training set K_l^T and the test set K_l^P. For higher-order structures that do not exist in K_l^T, the higher-order link prediction task is to identify the higher-order structures in the test set K_l^P as much as possible.

Therefore, traditional link prediction can be considered as given one node $k_1^i \in K_1$, finding which node $k_1^j \in K_1$ is more likely to generate links with it. We denote (k_1^i, k_1^j) as a candidate k_2, and measure its possibility of forming clique k_2 (edge) through algorithms, as shown in Fig. 1(a). Higher-order link prediction can be considered as given a clique k_l, finding which node k_1 is more likely to form k_{l+1} with it. We denote (k_1, k_l) as a candidate k_{l+1}. When $l = 2$, the higher-order structure to be predicted is a clique k_3, as shown in Fig. 1(c), and so on.

In this article, we focus on the prediction problem of cliques k_3. Specifically, in a network, for a given clique k_2 (edge), we predict which clique k_1 (nodes) are most likely to interact with k_2 to form a clique k_3 (triangle). To assess the prediction performance, the observed K_3 in the network is randomly divided into two parts, namely the training set K_3^T and the test set K_3^P. Here, K_3^T is considered as the known observed information, while K_3^P is only used for testing the algorithm. Obviously, $K_3^T \cup K_3^P = K_3$ and $K_3^T \cap K_3^P = \emptyset$.

3. Methods

In this section, we elaborate on the CICN framework for higher-order link prediction, which consists of the following three parts:

Higher-order link prediction framework based on topological mutual information. Given a higher-order interaction event and the known conditions under which this event occurs, the potential formation of higher-order cliques in the network can be assessed by calculating the uncertainty associated with the occurrence of this event.

Higher-order clustering characteristics of nodes. There are different number of higher-order structures around each node in the network, and the degree of aggregation of these higher-order structures can measure the likelihood of higher-order interactions between the node's neighbors. The higher the degree of aggregation, the more likely higher-order interactions are to occur.

Higher-order link prediction with clustering mutual information of common neighbors. Considering the degree of higher-order clustering around the nodes and the likelihood of higher-order interaction events occurring, the similarity score of the candidate k_3 is calculated.

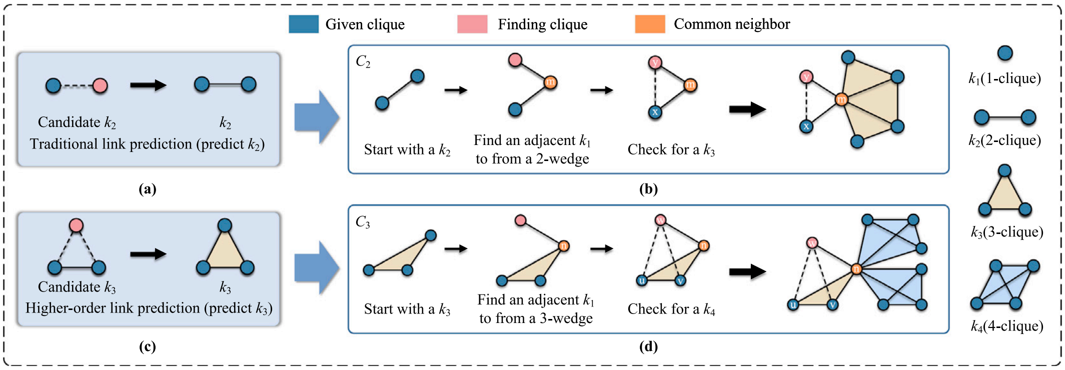


Fig. 1. This brief description outlines the problem of higher-order link prediction using common neighbor clustering mutual information and the prediction task of this paper. (a) Traditional link prediction. For an unconnected pair of nodes, the possibility of generating a connection is predicted (i.e., given a clique k_1, we find which other clique k_1 can form a candidate k_2 with it, and then predict the possibility of the candidate k_2 forming clique k_2). (b) The 2-order clustering coefficient C_2 evaluates the probability that clique k_2 is closed with its neighboring clique k_1. (c) Higher-order link prediction (predict k_3). Given a clique k_2, find which clique k_1 can form a candidate k_3 with it, and predict the possibility of the candidate k_3 forming clique k_3. (d) The 3-order clustering coefficient C_3 evaluates the probability that clique k_3 is closed with its neighboring clique k_1. Therefore, C_l evaluates the probability that clique k_l is closed with its neighboring clique k_1, which means that there are $l-1$ possible edges between clique k_l and its neighboring clique k_1 to form clique k_{l+1}.

3.1. Higher-order link prediction framework based on topological structure mutual information

In the information theory [27], information entropy can be used to measure the uncertainty of the occurrence of an event. Specifically, an event X with higher information entropy implies greater uncertainty regarding its occurrence, and this, in turn, indicates a lower probability of its happening [28]. Its information entropy $I(X)$ can be defined as:

$$I(X) = \log \frac{1}{P(X)} = -\log(P(X)) \tag{1}$$

According to Eq. (1), we can obtain the conditional self-information:

$$I(X_1|X_2) = -\log(P(X_1|X_2)) \tag{2}$$

where $P(X_1|X_2)$ represents the conditional probability of event X_1 occurring given that event X_2 has already occurred.

In higher-order link prediction (e.g., predicting a clique k_3), the uncertainty of interaction links to form a clique can be expressed using information entropy. In a network, given the candidate k_3, the event that it can form clique can be expressed as $L^{k_3}_{(k_1,k_2)}$. More specifically, the information entropy quantifies the likelihood of this clique actually forming. A lower entropy indicates a higher likelihood of the clique forming, and vice versa.

The research shows that different structural features can be used to improve the prediction ability of the model [29,30]. For example, common neighbors and community structure are two representative examples of network structural features that can significantly enhance prediction performance. From an information entropy perspective, these structural features can substantially reduce the uncertainty associated with higher-order interaction events. The more informative the structural features, the greater the reduction in uncertainty about higher-order interactions, leading to a higher likelihood of their occurrence. Correspondingly, for a given topological feature set Ω, the similarity method utilizing information entropy is defined as follows:

$$S^{\Omega}_{(k_1,k_2)} = -I(L^{k_3}_{(k_1,k_2)} \mid \Omega) \tag{3}$$

In the equation, $I(L^{k_3}_{(k_1,k_2)} \mid \Omega)$ is conditional self-information, which represents the uncertainty of whether cliques k_1 and k_2 will interact to form a clique k_3, given a set of known topological features Ω. Higher uncertainty between the two cliques indicates a lower similarity score and a decreased likelihood of interaction. Assuming elements in the set Ω are mutually independent, for a characteristic element ω (i.e., $\omega \in \Omega$), according to mutual information theory, $I(L^{k_3}_{(k_1,k_2)} \mid \Omega)$ is defined by:

$$\begin{aligned} I(L^{k_3}_{(k_1,k_2)}|\Omega) &= I(L^{k_3}_{(k_1,k_2)}) - I(L^{k_3}_{(k_1,k_2)};\Omega) \\ &= I(L^{k_3}_{(k_1,k_2)}) - \sum_{\omega\in\Omega} I(L^{k_3}_{(k_1,k_2)};\omega) \\ &= I(L^{k_3}_{(k_1,k_2)}) - \sum_{\omega\in\Omega} (I(L^{k_3}_{(k_1,k_2)}) - I(L^{k_3}_{(k_1,k_2)}|\omega)) \end{aligned} \tag{4}$$

Combining Eqs. (3) and (4), the topological structural similarity index can be defined for the given candidate k_3 as follows:

$$\begin{aligned} S^{\Omega}_{(k_1,k_2)} &= -I(L^{k_3}_{(k_1,k_2)}|\Omega) \\ &= \sum_{\omega\in\Omega} I(L^{k_3}_{(k_1,k_2)};\omega) - I(L^{k_3}_{(k_1,k_2)}) \\ &= \sum_{\omega\in\Omega} (I(L^{k_3}_{(k_1,k_2)}) - I(L^{k_3}_{(k_1,k_2)}|\omega)) - I(L^{k_3}_{(k_1,k_2)}) \end{aligned} \tag{5}$$

Eq. (5) indicates that each characteristic ω of candidate k_3 helps to reduce the uncertainty of the interaction between clique k_1 and clique k_2 to form clique k_3. The more informative a characteristic is, the greater its contribution to the likelihood of candidate k_3 forming. In addition to the structure of common neighbors, we integrate higher-order clustering characteristics of each common neighbor into a mutual information-based prediction model. The specific process is detailed in the following subsections.

3.2. Higher-order clustering characteristics of nodes

Traditional link prediction methods based on network structure utilize the local clustering coefficient of common neighbor nodes to quantify the interaction between node pairs [31]. As shown in Fig. 1(b), given a node pair (namely, candidate k_2) and their common neighbor m, the core principle of these methods is to evaluate the likelihood of forming a closed triangle (clique) among m's neighboring nodes, which is in line with the definition of local clustering coefficients (i.e., 2-order clustering coefficient). In other words, the higher the 2-order clustering coefficient of m, the greater the likelihood of forming an edge between x and y. Furthermore, when utilizing the higher-order clustering attributes of nodes, we observe that as the 3-order clustering coefficient of a node increases, the likelihood of its neighbors forming clique k_3 becomes higher. As shown in Fig. 1(d), the node w represents a clique k_1, and the edge (u, v) represents a clique k_2, for common neighbor node n of clique k_1 and clique k_2, a higher 3-order clustering coefficient for n indicates a greater likelihood of cliques k_1 and clique k_2 forming clique k_3. It can be inferred that when predicting the interaction between clique k_{l-1} and clique k_1 to form clique k_l, the closure probability of candidate k_l can be measured by the l-order clustering coefficient C_l of their common neighbors. The higher the clustering coefficient is, the greater the possibility of closure is, and the greater the probability that the clique k_l ($l \geq 2$) is formed. Therefore, it is a logical step towards enhancing prediction accuracy to incorporate the higher-order clustering structure into higher-order link prediction problem. Next, we will explore in more detail the influence of higher-order cluster topological features on higher-order link prediction, focusing on the perspective of clustering coefficients.

Generally, giving a node θ, its 2-order clustering coefficient quantifies the likelihood of forming a clique k_3 between this node and its neighboring nodes. A higher coefficient indicates a greater likelihood of forming a clique k_3. Specifically, based on this, the 2-order clustering coefficient of node θ can be defined as [32]:

$$C_2(\theta) = \frac{2|K_3(\theta)|}{|W_2(\theta)|} \tag{6}$$

here, $K_3(\theta)$ denotes the set of cliques containing node θ with size k_3, and $W_2(\theta)$ represents the set of wedges centered around node θ. The meaning of Eq. (6) is the probability of forming a clique k_3 between the node θ and its neighbors. A higher 2-order clustering coefficient implies a greater likelihood of clique formation. Similarly, the formation process of a clique (the size of k_4) can be measured using the 3-order clustering coefficient [32]. Specifically, the 3-order clustering coefficient quantifies the likelihood of forming a clique between the node θ and its neighbors nodes, which can be defined as:

$$C_3(\theta) = \frac{3|K_4(\theta)|}{|W_3(\theta)|} \tag{7}$$

by the same way, the 4-order clustering coefficient can be defined by:

$$C_4(\theta) = \frac{4|K_5(\theta)|}{|W_4(\theta)|} \tag{8}$$

the l-order clustering coefficient can be defined by:

$$C_l(\theta) = \frac{l|K_{l+1}(\theta)|}{|W_l(\theta)|} \tag{9}$$

where $K_{l+1}(\theta)$ denotes the set of k_{l+1} containing node θ, and $W_l(\theta)$ represents the set of l-order wedges centered on node θ.

The higher-order clustering coefficient describes the degree of aggregation of higher-order structures around a node. Considering both the topological structure mutual information theory in Section 3.1 and the higher-order clustering features of common neighbor nodes in Section 3.2, we propose a higher-order link prediction method with clustering mutual information of common neighbors.

3.3. Higher-order link prediction with clustering mutual information of common neighbors

For an unconnected node pairs (x, y), the traditional link prediction always focuses on the probability of potential link between these two nodes. Therefore, the occurring probability of the event $L^{k_2}_{(x,y)}$ without considering any prior structure can be calculated as [29]

$$\begin{aligned} P\left(L^{k_2}_{(x,y)}\right) &= \frac{N^{v_1} + N^{v_2} + N^{v_3} + \cdots + N^{v_{|V|}}}{|V|(|V|-1)} \\ &= \frac{2\,|E|}{|V|(|V|-1)} \end{aligned} \tag{10}$$

where N^{v_i} represents the degree of node v_i, $|V|$ and $|E|$ are the number of nodes and the edges in the network, respectively. Given a clique k_2 and a candidate seed node k_1, with respect to the higher-order problem of predicting the clique k_3, the occurrence probability of the event $L^{k_3}_{(k_1,k_2)}$ without taking any prior structure into account can be extended as:

$$\begin{aligned} P\left(L^{k_3}_{(k_1,k_2)}\right) &= \frac{N^{e_1} + N^{e_2} + N^{e_3} + \cdots + N^{e_{|E|}}}{|E|(|V|-2)} \\ &= \frac{3\,|K_3|}{|E|(|V|-2)} \end{aligned} \tag{11}$$

where N^{e_i} represents the total number of clique k_3 formed by edge e_i, $|K_3|$ is the total number of clique k_3 in the network.

According to the definition of information entropy, the entropy of the event $L^{k_3}_{(k_1,k_2)}$ can be depicted as:

$$\begin{aligned} I(L^{k_3}_{(k_1,k_2)}) &= -\log(P(L^{k_3}_{(k_1,k_2)})) \\ &= -\log(\frac{3\,|K_3|}{|E|(|V|-2)}) \end{aligned} \tag{12}$$

The common neighbor structure between node pairs is always utilized to address the traditional link prediction problem. Similarly, we also take into account the common neighbors surrounding a candidate clique k_3 in order to enhance the prediction ability of higher-order interactions. Given a clique k_1 and one clique k_2, their common neighbor set $\Gamma_{(k_1,k_2)}$ can be defined as:

$$\Gamma_{(k_1,k_2)} = \Gamma_{k_1} \cap \Gamma_{k_2} \tag{13}$$

where Γ_{k_1} and Γ_{k_2} represents the neighbor set of clique k_1 and clique k_2, respectively.

Suppose $\omega \in \Gamma_{(k_1,k_2)}$ is one of the common neighbors of clique k_1 and k_2. When considering the l-order clustering property of ω, we use $P_l(\omega)$ to denote the probability that clique k_1 and k_2 interact to form clique k_3, and then the probability that they do not interact can be expressed by $\overline{P_l(\omega)}$. Since the higher-order clustering coefficient $C_l(\omega)$ represents the probability that these two adjacent cliques form clique k_3, taking into account the l-order clustering structure of node ω, it exists:

$$\begin{aligned} &C_l(\omega) \approx P_l(\omega) \\ &\overline{P_l(\omega)} = 1 - P_l(\omega) \approx 1 - C_l(\omega) \end{aligned} \tag{14}$$

Suppose that the different-order clustering coefficients of node are mutually independent. In accordance with probability theory and taking into account the l-order cluster structure, the probability that the clique k_1 and k_2 interact to form clique k_3 can be further defined as:

$$\begin{aligned} P(L^{k_3}_{(k_1,k_2)}|\omega) &= 1 - \overline{P_3(\omega)} * \overline{P_4(\omega)} * \cdots * \overline{P_l(\omega)} \\ &= 1 - (1 - P_3(\omega)) * (1 - P_4(\omega)) * \cdots * (1 - P_l(\omega)) \\ &\approx 1 - (1 - C_3(\omega)) * (1 - C_4(\omega)) * \cdots * (1 - C_l(\omega)) \\ &= 1 - \prod_{j=3}^{l}(1 - C_j(\omega)) \end{aligned} \tag{15}$$

Combining Eqs. (1) and (15), given one of the common neighbors ω between cliques k_1 and k_2 and considering the different l-order clustering coefficients of node ω, its contribution to the formation of clique k_3 can be defined by the conditional entropy of node ω, which is depicted as:

$$\begin{aligned} I(L^{k_3}_{(k_1,k_2)}|\omega) &= -\log(P(L^{k_3}_{(k_1,k_2)}|\omega)) \\ &\approx -\log(1 - \prod_{j=3}^{l}(1 - C_j(\omega))) \end{aligned} \tag{16}$$

According to Eq. (3), given the common neighbor set $\Gamma_{(k_1,k_2)}$ of the clique k_1 and clique k_2, the similarity considering l-order clustering properties of all common neighbors is defined by:

$$\begin{aligned} S^{\Omega}_{(k_1,k_2)} &= -I(L^{k_3}_{(k_1,k_2)}|\Gamma_{(k_1,k_2)}) \\ &= \sum_{\omega\in\Gamma_{(k_1,k_2)}} (I(L^{k_3}_{(k_1,k_2)}) - I(L^{k_3}_{(k_1,k_2)}|\omega)) - I(L^{k_3}_{(k_1,k_2)}) \\ &\approx \sum_{\omega\in\Gamma_{(k_1,k_2)}} (-\log(\frac{3\,|K_3|}{|E|(|V|-2)}) - (-\log(1 - \prod_{j=3}^{l}(1 - C_j(\omega))))) \\ &\quad - (-\log(\frac{3\,|K_3|}{|E|(|V|-2)})) \\ &= \sum_{\omega\in\Gamma_{(k_1,k_2)}} (\log(1 - \prod_{j=3}^{l}(1 - C_j(\omega))) - \log(\frac{3\,|K_3|}{|E|(|V|-2)})) \\ &\quad + \log(\frac{3\,|K_3|}{|E|(|V|-2)}) \end{aligned} \tag{17}$$

Due to the complexity of the calculation, we only consider the contributions of the 3-order, 4-order, and 5-order clustering coefficients ($3 \le l \le 5$), therefore, the CICN index is defined as follows:

$$\begin{aligned} S^{CICN}_{(k_1,k_2)} &= -I(L^{k_3}_{(k_1,k_2)} | \Gamma_{(k_1,k_2)}) \\ &= \sum_{\omega \in \Gamma_{(k_1,k_2)}} (I(L^{k_3}_{(k_1,k_2)}) - I(L^{k_3}_{(k_1,k_2)} | \omega)) - I(L^{k_3}_{(k_1,k_2)}) \\ &\approx \sum_{\omega \in \Gamma_{(k_1,k_2)}} (-\log(\frac{3|K_3|}{|E|(|V|-2)}) - (-\log(1 - \prod_{j=3}^{5}(1 - C_j(\omega))))) \\ &\quad - (-\log(\frac{3|K_3|}{|E|(|V|-2)})) \\ &= \sum_{\omega \in \Gamma_{(k_1,k_2)}} (\log(1 - \prod_{j=3}^{5}(1 - C_j(\omega))) - \log(\frac{3|K_3|}{|E|(|V|-2)})) \\ &\quad + \log(\frac{3|K_3|}{|E|(|V|-2)}) \\ &= \sum_{\omega \in \Gamma_{(k_1,k_2)}} (\log(1 - (1 - C_3(\omega)) * (1 - C_4(\omega)) * (1 - C_5(\omega))) \\ &\quad - \log(\frac{3|K_3|}{|E|(|V|-2)})) + \log(\frac{3|K_3|}{|E|(|V|-2)}) \end{aligned} \tag{18}$$

4. Benchmarks and evolution metrics

4.1. Benchmarks

In this paper, we choose 13 benchmarks to compare with CICN, including 8 higher-order extensions of traditional link prediction algorithms and 5 higher-order link prediction methods for predicting clique k_3, as follows.

4.1.1. Higher-order extensions of traditional link prediction algorithms

Common neighbor (CN) [33] For a given clique k_2 (edge) and the recommended clique k_1 (node), if they share more common neighbors, they are more likely to interact to form clique k_3 (triangle).

$$S^{CN}_{(k_1,k_2)} = |\Gamma_{k_1} \cap \Gamma_{k_2}| \tag{19}$$

where Γ_{k_1} and Γ_{k_2} represent the neighbor sets of clique k_1 and clique k_2, respectively.

Adamic–Adar (AA) [34] improves CN by giving higher weight to common neighbors with smaller degrees.

$$S^{AA}_{(k_1,k_2)} = \sum_{z \in \Gamma_{k_1} \cap \Gamma_{k_2}} \frac{1}{\log(|\Gamma_z|)} \tag{20}$$

where z is the common neighbor of clique k_1 and clique k_2, and $|\Gamma_z|$ is the degree of node z.

In addition, for a given clique k_2 and the recommended clique k_1, such as edge (u, v) and node w, **AA-MAX** [17] represents the maximum value between $S^{AA}_{(w,u)}$ and $S^{AA}_{(w,v)}$. **AA-MUL** [17] denotes the product between $S^{AA}_{(w,u)}$ and $S^{AA}_{(w,v)}$, as the similarity score, which are defined as:

$$\begin{aligned} S^{AA-MAX}_{(k_1,k_2)} &= \max(S^{AA}_{(w,u)}, S^{AA}_{(w,v)}) \\ &= \max(\sum_{\delta \in \Gamma_w \cap \Gamma_u} \frac{1}{\log(|\Gamma_\delta|)}, \sum_{\eta \in \Gamma_w \cap \Gamma_v} \frac{1}{\log(|\Gamma_\eta|)}) \end{aligned} \tag{21}$$

$$\begin{aligned} S^{AA-MUL}_{(k_1,k_2)} &= S^{AA}_{(w,u)} \bullet S^{AA}_{(w,v)} \\ &= \sum_{\delta \in \Gamma_w \cap \Gamma_u} \frac{1}{\log(|\Gamma_\delta|)} \bullet \sum_{\eta \in \Gamma_w \cap \Gamma_v} \frac{1}{\log(|\Gamma_\eta|)} \end{aligned} \tag{22}$$

where δ is the common neighbor of node w and u, and η is the common neighbor of node w and v.

Jaccard Similarity (JS) [35] normalizes the common neighbor size of clique k_1 and clique k_2. It is defined by:

$$S^{JS}_{(k_1,k_2)} = \frac{|\Gamma_{k_1} \cap \Gamma_{k_2}|}{|\Gamma_{k_1} \cup \Gamma_{k_2}|} \tag{23}$$

Similarly, for edge (u, v) and node w, **JS-MAX** [17] represents the maximum value between $S^{JS}_{(w,u)}$ and $S^{JS}_{(w,v)}$. **JS-MUL** [17] denotes the product between $S^{JS}_{(w,u)}$ and $S^{JS}_{(w,v)}$, which are defined as:

$$\begin{aligned} S^{JS-MAX}_{(k_1,k_2)} &= \max(S^{JS}_{(w,u)}, S^{JS}_{(w,v)}) \\ &= \max(\frac{|\Gamma_w \cap \Gamma_u|}{|\Gamma_w \cup \Gamma_u|}, \frac{|\Gamma_w \cap \Gamma_v|}{|\Gamma_w \cup \Gamma_v|}) \end{aligned} \tag{24}$$

$$\begin{aligned} S^{JS-MUL}_{(k_1,k_2)} &= S^{JS}_{(w,u)} \bullet S^{JS}_{(w,v)} \\ &= \frac{|\Gamma_w \cap \Gamma_u|}{|\Gamma_w \cup \Gamma_u|} \bullet \frac{|\Gamma_w \cap \Gamma_v|}{|\Gamma_w \cup \Gamma_v|} \end{aligned} \tag{25}$$

Degree of Gravity for Link Prediction (DGLP) [36] combines degree centrality, common neighbors, and distance between candidate k_3 to mitigate the cold start problem for higher-order link prediction. It is defined by:

$$S^{DGLP}_{(k_1,k_2)} = \frac{|\Gamma(k_1)| + |\Gamma(k_2)|}{d_{(k_1,k_2)} + 1} + \sum_{z \in \Gamma(k_1)\Gamma(k_2)} k_z \tag{26}$$

where $d_{(k_1,k_2)}$ is the shortest distance between k_1 and k_2.

4.1.2. Higher-order link prediction algorithms

Pair-seeded PageRank (PSP) [17] The flow of information from seed nodes to other nodes in the network is modeled by Markov chains, and the smooth distribution of the chains provides the scores on the nodes. Pair-seeded PageRank is equivalent to the sum of Single-seeded PageRank (SSP) on each node, at most a scalar multiple. It can be extended to consider the maximum similarity score of node u and node v with other nodes **(SSP-MAX)**, and the product of similarity scores **(SSP-MUL)**.

Triangle reinforced PageRank (TRPR) [37] assigns higher weights to edges involved in many clique k_3 to enhance the influence of clique k_3. Although TRPR introduces higher weights for edges involved in many k_3 by forming a new adjacency matrix $\hat{X} + A$, these weights are usually dominated by the weights in the adjacency matrix A. In order to make a fair contribution to these edges, the **weighted version of TRPR (TRPRW)** introduces a scalar multiple to $\hat{X}$. Any scalar multiplied by $\hat{X}$ produces a fair contribution of weights between $\hat{X}$ and A.

4.2. Evaluation metrics

To measure the performance of different higher-order link prediction methods in this paper, we utilize the AUC and AUPR metrics.

4.2.1. AUC

AUC [38] (Area Under the ROC Curve) represents the probability that a randomly selected clique k_3 from the test cliques set K^P_3 has a higher score than a randomly selected clique from the unobserved cliques set $(U_3 - K_3)$, where U_3 represents all possible clique k_3 in the network. The value of AUC can be calculated as:

$$\text{AUC} = \frac{n' + 0.5 \times n''}{n} \tag{27}$$

where n represents the total number of comparisons. n' denotes the number of times that the clique k_3 from K^P_3 have higher similarity scores than clique k_3 from $(U_3 - K_3)$, and n'' is the number of times that two cliques k_3 have the same score.

4.2.2. AUPR

The link prediction problem can be regarded as a binary classification task [39]. Therefore, evaluation metrics commonly used in binary classification task can be applied to evaluate link prediction performance. In a binary classification task, we represent all samples as positive and negative examples. The relationship between the model's predictions for these samples and their true classes can be captured using a confusion matrix [40]. True Positive (TP) represents the situation where both the model prediction and the actual result are positive; True Negative (TN) represents the situation where both the model prediction

Table 1
Topology information of nine empirical networks. $| V |$ and $| E |$ represent the number of nodes and edges in the network, respectively, the average degree is represented as $\langle k \rangle$, the network density is represented as S. C_3, C_4 and C_5 represent the 3-order, 4-order and 5-order clustering coefficients respectively. $| K_3 |$ and $| K_4 |$ represent the number of 3-cliques and 4-cliques in the network, respectively.

Datasets	$\vert V \vert$	$\vert E \vert$	$\langle k \rangle$	S	C_3	C_4	C_5	$\vert K_3 \vert$	$\vert K_4 \vert$
Macaque	93	2262	48.645	0.529	0.820	0.806	0.797	29638	257607
Journals	124	5972	96.323	0.783	0.814	0.777	0.752	169978	3357799
Scc-enron	146	9828	134.630	0.929	0.937	0.922	0.911	425267	13409478
145Bit	1002	11252	22.459	0.022	0.047	0.011	0.001	13656	4288
Power	1174	8687	14.799	0.013	0.346	0.305	0.299	140724	2279121
Hamsterster	2000	16097	16.097	0.008	0.421	0.334	0.256	52651	131905
162Bit	3606	37069	20.560	0.006	0.023	0.001	0.000	25725	4180
Chebyshev3	4101	24582	11.988	0.003	0.799	0.699	0.571	61431	575866
Ca-GrQc	5241	14484	5.527	0.001	0.290	0.154	0.097	48260	329297

and the actual result are negative; False Positive (FP) represents the situation where the model prediction result is positive, but the actual result is negative; False Negative (FN) represents the situation where the model prediction result is negative, but the actual result is positive. From the confusion matrix, we can derive the values of precision and recall metrics as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{28}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{29}$$

AUPR [41] is composed of Recall on the X-axis and Precision on the Y-axis.

5. Experiment result

In this section, we evaluate the performance of CICN on higher-order link prediction on 9 empirical networks by answering the following questions:

- **Q1**: Which higher-order clustering structure information can improve the prediction accuracy of clique k_3?
- **Q2**: How does the prediction performance of CICN proposed in this paper compare with other benchmarks?
- **Q3**: How robust is CICN with different proportional training sets?
- **Q4**: How to solve the imbalance of positive and negative samples in higher-order link prediction?

5.1. Datasets

In this work, CICN makes predictions on 9 empirical networks from different fields. Table 1 provides the topological features and detailed information of these networks. (1) Macaque [42]: is a brain network, referring to the connections between neurons in the brain. (2) Journals [42]: is a miscellaneous network. (3) Scc-enron [42]: is a temporal reachability network, where its links are only active at certain time points. Each link carries information about when it is active, as well as other possible features. (4) 145Bit [42]: is a network formed by decomposing 145 bit numbers. (5) Power [42]: refers to networks used for transmission and distribution of electricity. These networks include components such as power plants, substations, transmission lines, and distribution lines. (6) Hamsterster [42]: is a social network where individuals can exchange, share information, and even collaborate in some cases. (7) 162Bit [42]: is a network formed by decomposing 162 bit numbers. (8) Chebyshev3 [42]: is in the collection of Miscellaneous Networks. (9) Ca-GrQc [42]: is in the category of Collaboration Networks. All of the above networks can be downloaded from the following websites: https://networkrepository.com/index.php and https://snap.stanford.edu/data/

5.2. Divide the datasets

The traditional link prediction task focuses on predicting clique k_2 (edge). To evaluate the model's performance, it is common to randomly remove a certain proportion of edges from the original network as the training set and use the removed edges as the test set. Since the higher-order link prediction task in this paper involves predicting which clique k_1 (node) in the network will interact with a given clique k_2 (edge) to form clique k_3 (triangle), we extend the traditional dataset partitioning method.

Specifically, let K_3 denote the set of all already existing cliques k_3 (triangles) in the network. We first randomly select a certain proportion of cliques k_3 from K_3. Then, for each selected clique k_3, we randomly remove its wedge structure from the original network. Finally, the remaining network serves as the training set, and the removed wedge structures constitute the test set. To assess the predictive ability of the higher-order model, for each clique k_2 (edge) in the training set, the higher-order model assigns a score to every clique k_1 (node) that did not interact with it. The higher the score, the higher the probability of forming clique k_3 (triangle).

5.3. Prediction results for different cluster structures (Q1)

According to the CICN definition given above, CICN quantifies the contributions of various higher-order clustering coefficients associated with common neighbor nodes using information entropy. In this experiment, we set the training set ratio to 90% and conduct 50 independent experiments on 9 empirical networks. The AUC and AUPR results are shown in Figs. 2 and 3. Specifically, CICN_o^3, CICN_o^4, and CICN_o^5 represent the cases where only the 3-order, 4-order, and 5-order clustering coefficient are considered individually. CICN_f^{34} represents the fusion of 3-order and 4-order clustering coefficients, and CICN_f^{345} represents the fusion of 3-order, 4-order, and 5-order clustering coefficients.

From Figs. 2 and 3, the utilization of higher-order clustering structures can substantially enhance predictive performance. As evidenced by CICN_o^3 superior AUC scores on Macaque, 145Bit, Hamsterster, 165Bit, and Chebyshev3 networks, this finding aligns with our previous theoretical explanations. Specifically, for both the Journals, Scc-enron, and Chebyshev3 networks, CICN_o^3 outperforms CICN_o^4 and CICN_o^5 in terms of both AUC and AUPR, possibly due to the significance of the high-order clustering structure. As shown in Table 1, these three networks have higher 3-order and 4-order clustering coefficients, and also have more 3-cliques and 4-cliques.

Based on the experimental results mentioned above, it is evident that incorporating higher-order cluster structure information within the network enhances prediction accuracy. Specifically, we found that the 3-order cluster structure information is more favorable to the prediction of 3-clique, and other higher-order cluster information can also improve the prediction performance to a certain extent. However, as the higher-order clustering coefficients of nodes increase, the computation process also becomes more complex. To address this trade-off between prediction accuracy and computational efficiency, this paper carefully selects CICN_o^3 and CICN_f^{34} for subsequent experiments. CICN_o^3 considers only the 3-order clustering coefficient, while CICN_f^{34} fuses the 3-order and 4-order clustering coefficients (i.e., $\text{CICN}_f^{3/34}$). By focusing on these specific higher-order clustering coefficients, we aim to strike

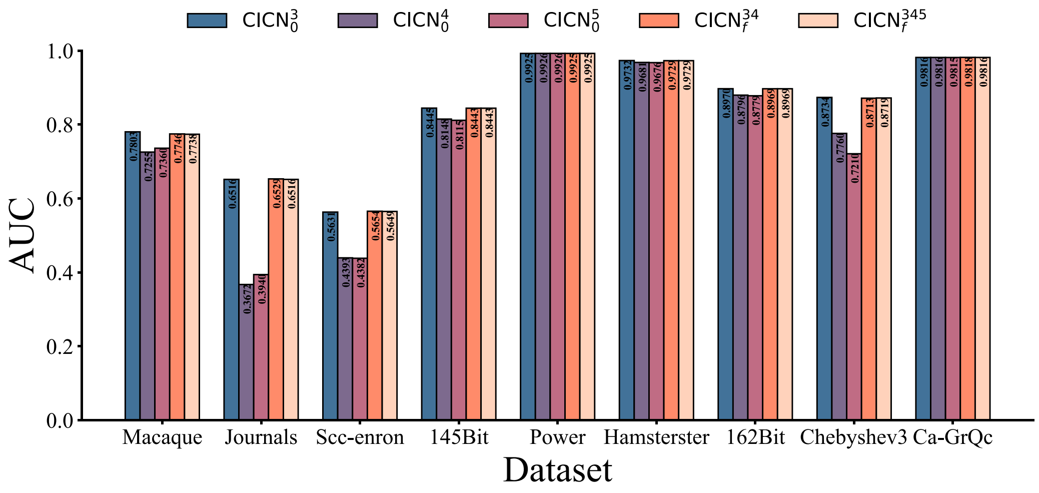


Fig. 2. The impact of different higher-order clustering structures on AUC. The X-axis represents different Datasets, and the Y-axis is the AUC value of the method on the dataset. The random sampling rate of the training set is set to 90%, $CICN_o$ represents considering only one higher-order clustering structure, while $CICN_f$ represents the fusion of two or more higher-order clustering structures.

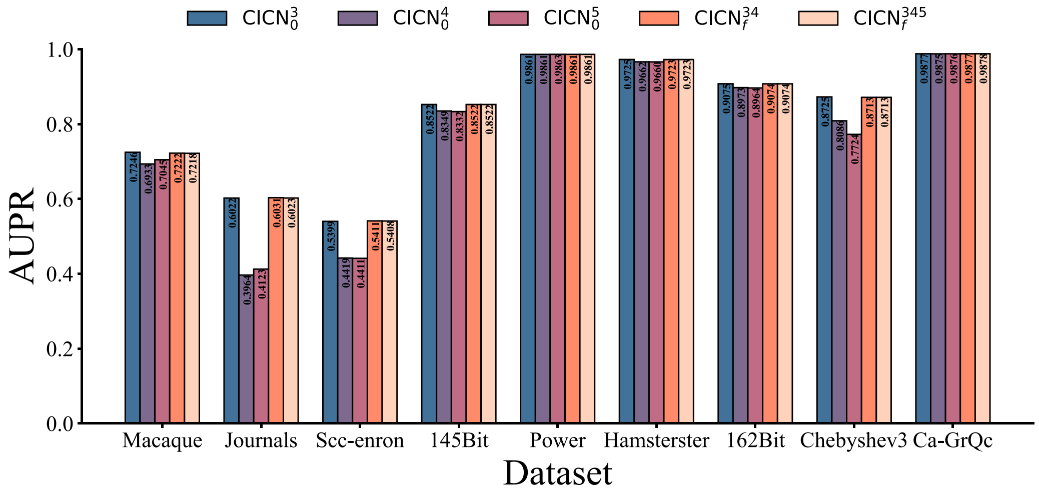


Fig. 3. The impact of different higher-order clustering structures on AUPR. The X-axis represents different Datasets, and the Y-axis is the AUPR value of the method on the dataset. The random sampling rate of the training set is set to 90%, $CICN_o$ represents considering only one higher-order clustering structure, while $CICN_f$ represents the fusion of two or more higher-order clustering structures.

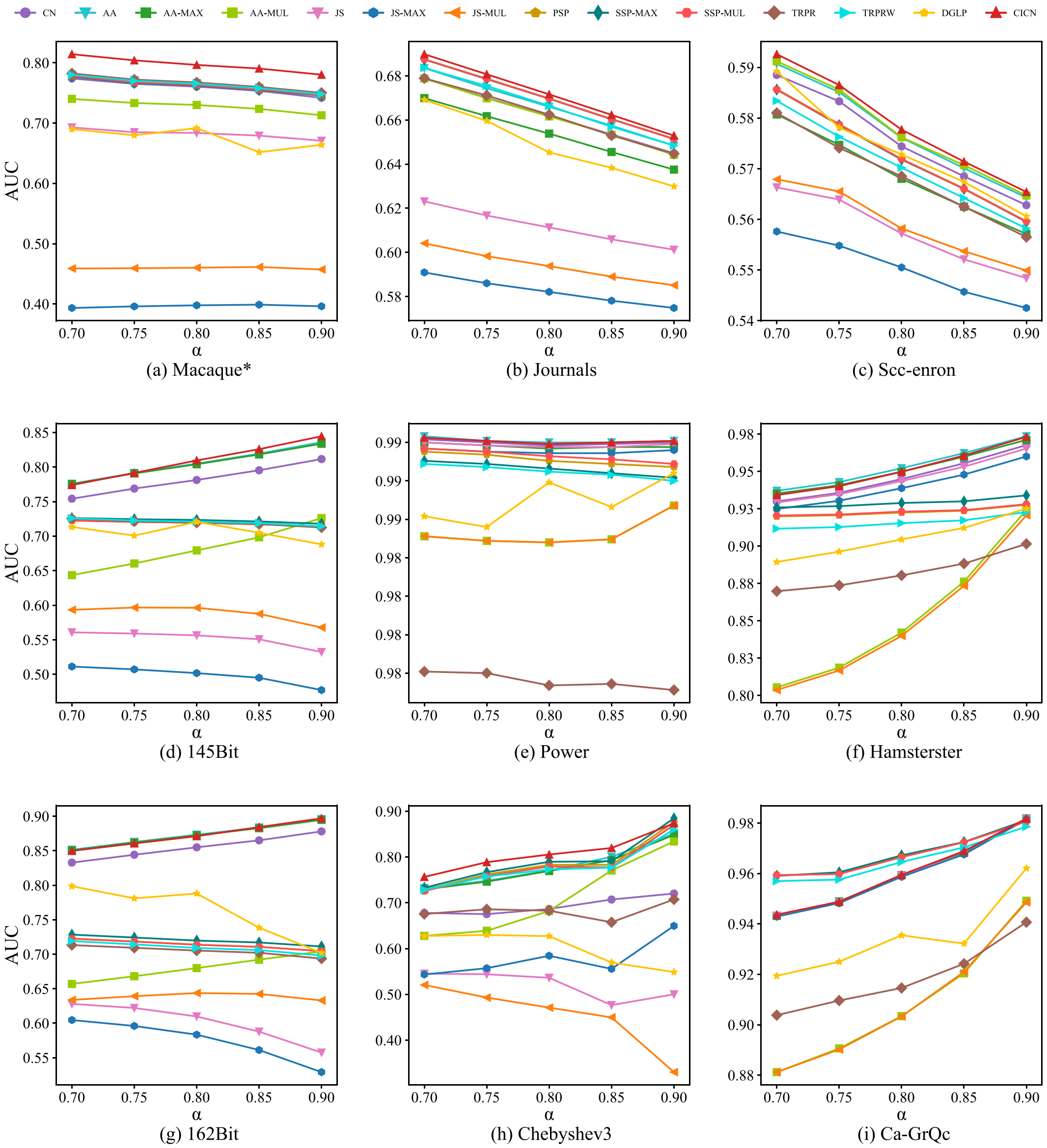
a balance between achieving high prediction accuracy and maintaining low computational complexity. This approach allows us to leverage the valuable information provided by higher-order cluster structures without sacrificing computational efficiency.

5.4. Comparative analysis of CICN and benchmarks (Q2)

In this section, the comparison of the prediction performance of the CICN method with 13 other benchmarks (i.e., CN, AA, AA-MAX, AA-MUL, JS, JS-MAX, JS-MUL, PSP, SSP-MAX, SSP-MUL, DGLP, TRPR and TRPRW) is mainly studied. In this experiment, we also set the training set ratio to 90% and conduct 50 independent experiments. The average AUC and AUPR results are shown in Tables 2 and 3, respectively, in which bold indicates the best results on each network, and the suboptimal are underlined.

For AUC, as can be seen in Table 2, $CICN_o^3$ has the best prediction results on three networks (i.e., Macaque 145Bit and 165Bit), and the suboptimal results on Hamsterster and Chebyshev3 networks. Moreover, in addition to considering the 3-order clustering coefficient, it can be found that $CICN_f^{34}$ performs best on both the Journals and Scc-enron networks, and $CICN_o^5$ achieves suboptimal results on the Power network. And similar results in AUPR, we find that CICN achieves the best results on five networks and the suboptimal results on the remaining three networks. Experimental results demonstrate that incorporating higher-order clustering structures can significantly improve the performance of predictive models, showing a clear advantage over baseline methods.

For the benchmarks, in traditional link prediction methods, AA achieves the best results on the Power, Hamsterster and Ca-GrQc networks. While both AA and CN measure similarity based on the number of common neighbors, AA consistently outperforms CN across various datasets. This advantage is particularly evident in networks with higher clustering coefficients. The reason is that in such networks, transforming more triples into triangles yields higher weights and scores. In Higher-order link prediction methods, SSP-MAX achieves optimal performance on Chebyshev3 networks due to its high computational efficiency, enabling it to operate effectively on large-scale networks and excel in link prediction tasks.



(a) Macaque*

(b) Journals

(c) Scc-enron

(d) 145Bit

(e) Power

(f) Hamsterster

(g) 162Bit

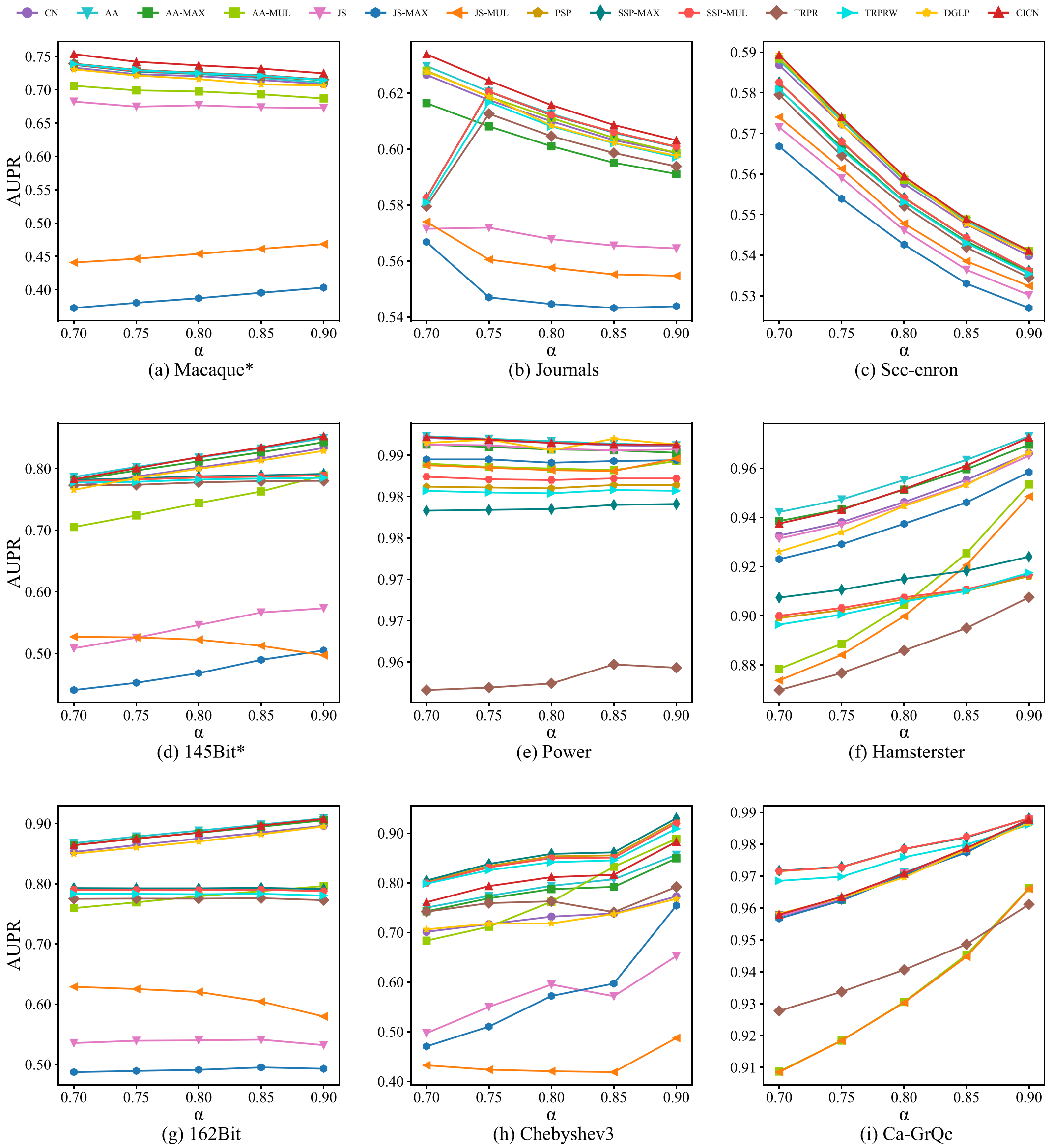
(h) Chebyshev3

(i) Ca-GrQc

Fig. 4. The predictive performance of 14 methods on 9 different network training ratios is measured by AUC. Each data point represents the average of 50 independent experiments. The X-axis is the training set ratio from 0.7 to 0.9, and the Y-axis is the AUC value. The parts of the network marked with * in the figure have lower AUC values for some methods, which affects the visual effect and are not shown (Macaque: JS-MUL, PSP).

Table 2
Comparison of prediction accuracy under the AUC indicator of 14 methods on 9 empirical datasets. The training set is split independently at a ratio of 9:1 and averaged 50 times. The best-performing method is marked in bold on each dataset, and the suboptimal results are underlined. Among them, CICN* is the best result among all methods using higher-order information. $\text{CICN}_f^{3/34}$ is the result using 3-order alon or fusing the 3-order and 4-order information.

Datasets	CN	AA	AA-MAX	AA-MUL	JS	JS-MAX	JS-MUL	PSP	SSP-MAX	SSP-MUL	DGLP	TRPR	TRPRW	$\text{CICN}_f^{3/34}$	CICN*
Macaque	0.7419	0.7468	0.7461	0.7130	0.6707	0.3961	0.4572	0.7466	0.7457	0.7466	0.6639	0.7502	0.7482	**0.7803**	$\mathbf{0.7803}_o^3$
Journals	0.6449	0.6484	0.6375	0.6442	0.6012	0.5748	0.5851	0.6515	0.6514	0.6514	0.6299	0.6447	0.6484	**0.6529**	$\mathbf{0.6529}_f^{34}$
Scc-enron	0.5628	0.5643	0.5571	0.5647	0.5484	0.5425	0.5499	0.5596	0.5595	0.5596	0.5606	0.5565	0.5582	**0.5654**	$\mathbf{0.5654}_f^{34}$
145Bit	0.8115	0.8360	0.8337	0.7261	0.5322	0.4772	0.5677	0.7137	0.7179	0.7135	0.6880	0.7125	0.7143	**0.8445**	$\mathbf{0.8445}_o^3$
Power	0.9925	**0.9927**	0.9922	0.9884	0.9924	0.9920	0.9884	0.9909	0.9902	0.9911	0.9905	0.9764	0.9900	0.9925	0.9926_o^5
Hamsterster	0.9677	**0.9736**	0.9712	0.9242	0.9654	0.9601	0.9207	0.9277	0.9340	0.9281	0.9252	0.9015	0.9228	0.9732	0.9732_o^3
162Bit	0.8781	0.8958	0.8948	0.7041	0.5574	0.5291	0.6330	0.7047	0.7113	0.7045	0.7020	0.6937	0.6981	**0.8970**	$\mathbf{0.8970}_o^3$
Chebyshev3	0.7202	0.8508	0.8482	0.8339	0.5002	0.6495	0.3302	0.8663	**0.8844**	0.8700	0.5487	0.7073	0.8601	0.8734	0.8734_o^3
Ca-GrQc	0.9814	**0.9819**	0.9817	0.9492	0.9816	0.9815	0.9486	0.9810	0.9812	0.9806	0.9621	0.9407	0.9786	0.9818	0.9818_f^{34}



(a) Macaque*

(b) Journals

(c) Scc-enron

(d) 145Bit*

(e) Power

(f) Hamsterster

(g) 162Bit

(h) Chebyshev3

(i) Ca-GrQc

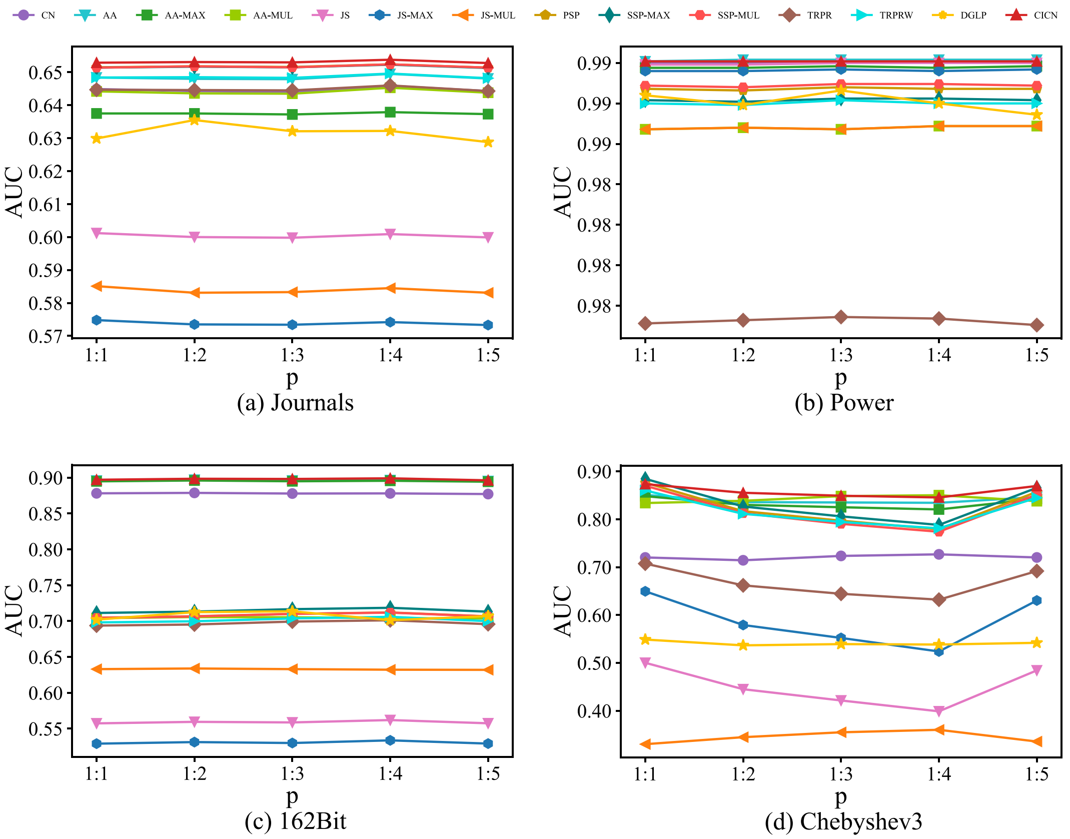
Fig. 5. The predictive performance of 14 methods on 9 different network training ratios is measured by AUPR. Each data point represents the average of 50 independent experiments. The X-axis is the training set ratio from 0.7 to 0.9, and the Y-axis is the AUPR value. The parts of the network marked with * in the figure have lower AUPR values for some methods, which affects the visual effect and are not shown (Macaque: JS-MUL, PSP, 145Bit: JS, JS-MAX, JS-MUL).

It can be seen that although CICN* often demonstrates superior predictive performance as evidenced in Tables 2 and 3, its computational complexity may be a concern in certain scenarios. To address this trade-off, we utilize $\text{CICN}_f^{3/34}$ in the subsequent sections, which offers a reasonable balance between efficiency and accuracy.

5.5. Algorithm stability verification (Q3)

In order to verify the stability of CICN, this subsection uses CICN to represent $\text{CICN}_f^{3/34}$, and conducts experiments under different training set ratios, and the AUC and AUPR results are shown in Figs. 4 and 5, respectively.

The result is shown in Figs. 4 and 5, as the proportion of the training set increases, the structural information available to the network becomes increasingly rich. Consequently, compared to the benchmarks, the AUC and AUPR results of CICN have improved. For our proposed method, CICN, combining node information entropy and clustering coefficients of different orders, accurately evaluates the contribution of common neighbors and effectively predicts 3-cliques in networks, significantly improving prediction accuracy, especially on the network Macaque, Journals, Scc-enron and 145Bit. The stability of CICN on the Macaque, Journals, and Scc-enron networks decreased as the training ratio increased. This might be due to the high density of these networks. When the training set proportion is excessively high, the prediction model may overfit the training data, leading to poor performance on the test data. However, for the other six networks, the model prediction accuracy generally improve with increasing training proportion, suggesting that this method is more suitable for scenarios with lower network density. When the training ratio is relatively low, the predictive performance of our proposed method undergoes significant changes. Removing too many 3-cliques greatly disrupts the networks higher-order structure, resulting in reduced influence from the higher-order clustering coefficient, which in turn affects CICN



(a) Journals

(b) Power

(c) 162Bit

(d) Chebyshev3

Fig. 6. The predictive performance of 14 methods on 4 different network test sample ratios is measured by AUC. Each point represents the average of 50 independent experiments. The X-axis represents the proportion of positive test samples to negative test samples, and the Y-axis is the AUC value.

Table 3
Comparison of prediction accuracy under the AUPR indicator of 14 methods on 9 empirical datasets. The training set is split independently at a ratio of 9:1 and averaged 50 times. The best-performing method is marked in bold on each dataset, and the suboptimal results are underlined. Among them, CICN* is the best result among all methods using higher-order information. $\mathrm{CICN}_f^{3/34}$ is the result using 3-order alon or fusing the 3-order and 4-order information.

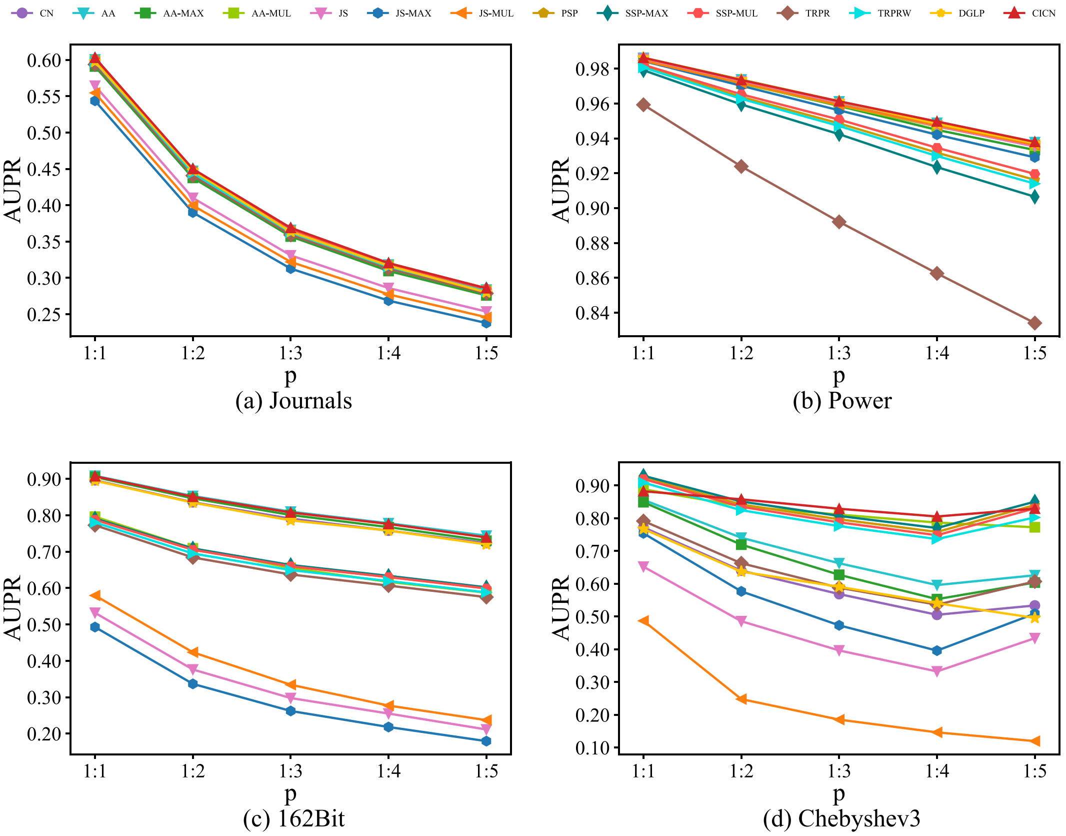
Datasets	CN	AA	AA-MAX	AA-MUL	JS	JS-MAX	JS-MUL	PSP	SSP-MAX	SSP-MUL	DGLP	TRPR	TRPRW	$\mathrm{CICN}_f^{3/34}$	CICN*
Macaque	0.7082	0.7106	0.7144	0.6869	0.6725	0.4032	0.4684	0.7152	0.7153	0.7153	0.7063	0.7153	0.7145	**0.7246**	$\mathbf{0.7246}_o^3$
Journals	0.5986	0.6007	0.5911	0.5987	0.5645	0.5438	0.5547	0.6009	0.6008	0.6008	0.5976	0.5938	0.5971	**0.6031**	$\mathbf{0.6031}_f^{34}$
Scc-enron	0.5398	0.5406	0.5359	0.5410	0.5302	0.5270	0.5324	0.5362	0.5362	0.5362	0.5406	0.5345	0.5354	**0.5411**	$\mathbf{0.5411}_f^{34}$
145Bit	0.8332	0.8492	0.8425	0.7887	0.5732	0.5048	0.4971	0.7891	0.7911	0.7890	0.8285	0.7799	0.7848	**0.8522**	$\mathbf{0.8522}_o^3$
Power	0.9861	0.9862	0.9853	0.9843	0.9857	0.9844	0.9846	0.9814	0.9791	0.9822	0.9863	0.9593	0.9807	0.9861	$\mathbf{0.9863}_o^5$
Hamsterster	0.9661	**0.9730**	0.9696	0.9534	0.9651	0.9584	0.9485	0.9161	0.9240	0.9167	0.9664	0.9075	0.9175	0.9725	0.9725_o^3
162Bit	0.8964	**0.9086**	0.9052	0.7961	0.5320	0.4928	0.5796	0.7879	0.7911	0.7878	0.8951	0.7727	0.7803	0.9075	0.9075_o^3
Chebyshev3	0.7724	0.8563	0.8492	0.8886	0.6522	0.7540	0.3870	0.9242	**0.9298**	0.9199	0.7674	0.7916	0.9089	0.8725	0.8725_o^3
Ca-GrQc	0.9876	**0.9878**	0.9877	0.9662	0.9876	0.9874	0.9660	0.9861	0.9871	0.9861	0.9866	0.9611	0.9861	0.9877	0.9877_o^3

overall performance. The experimental results show that this method maintains better predictive performance on most networks, indicating the reliability of the CICN.

For the baselines, we find that AA and SSP-MUL methods exhibit excellent performance on the Hamsterster and Ca-GrQc networks, respectively, even surpassing the CICN method. In addition, the performance of both methods show a steady upward trend as the training proportion increased. Both methods excel in higher-order networks primarily due to their ability to more effectively capture complex relationships between nodes. The AA method is particularly adept at adapting to the heterogeneity of networks, while the SSP-MUL method can better capture complex relationships between nodes by considering the influence of multiple seed nodes simultaneously, thereby improving prediction accuracy.

5.6. Constructing data using negative sampling method (Q4)

In this study, when calculating the prediction score of the model, there is an imbalance problem that the number of positive samples is small while the number of negative samples is large, which leads to the high complexity of the model. In machine learning, negative sampling is often used to balance the number of positive and negative samples. Inspired by this, we adopt the method of negative sampling for data construction, by generating negative samples with the same number of positive samples to balance their proportions, so as to improve the predictive power of the model [43,44]. The details are as follows: randomly select 90% 3-cliques as the training set, and the test set is composed of the remaining 3-cliques as the positive test samples and the same number of randomly sampled non-existent 3-cliques as the negative test samples. In order to verify the feasibility of this method,



(a) Journals

(b) Power

(c) 162Bit

(d) Chebyshev3

Fig. 7. The predictive performance of 14 methods on 4 different network test sample ratios is measured by AUPR. Each point represents the average of 50 independent experiments. The X-axis represents the proportion of positive test samples to negative test samples, and the Y-axis is the AUPR value.

we conduct experiments on Journals, Power, 162Bit, and Chebyshev3 networks. In the test set, positive and negative samples are divided according to the ratio of 1:1, 1:2, 1:3, 1:4 and 1:5. The AUC and AUPR results on all methods are shown in Figs. 6 and 7, CICN is used to represent $\text{CICN}_{f}^{3/34}$.

As shown in Figs. 6 and 7, the experimental results demonstrate that CICN consistently achieves the best prediction performance on Journals, 162Bit, and Chebyshev3 networks across five distinct positive and negative sample partitioning ratios. Furthermore, CICN's performance on the Power network closely approaches the optimal level. Fig. 6 illustrates that the AUC metric is less sensitive to variations in the proportion of negative samples, exhibiting a stable trend. In contrast, Fig. 7 shows a sharp decline in the precision of the AUPR metric. This is because AUC focuses on the model's ability to rank samples, while AUPR places more emphasis on the model's precision for positive samples. As a result, when the number of negative samples increases, the model's precision is more susceptible to being affected, leading to a decline in the AUPR metric, whereas the AUC metric remains relatively stable. In the AUPR evaluation of the Chebyshev3 network, as the ratio of positive to negative samples increases from 1:3 to 1:5, the rate of performance degradation gradually slows down. This is because when the proportion of negative samples reaches a certain level, the model misclassifies a large number of negative samples as positive ones, and further increasing the number of negative samples has a negligible impact on the model's accuracy.

From the perspective of sample imbalance, in extremely imbalanced datasets, increasing the number of negative samples can lead the model to be biased towards predicting samples as negative, thereby affecting the overall performance of the model. From the perspective of evaluation metrics, AUC and AUPR focus on different performance indicators, resulting in varying sensitivities to sample imbalance. In conclusion, employing negative sampling for data construction, such as maintaining a 1:1 ratio of positive to negative samples, can effectively address the imbalance issue in higher-order link prediction without compromising predictive performance. Moreover, this approach can enhance the computational efficiency of the model.

6. Conclusion

In this paper, we investigate the influence of higher-order clustering coefficients and introduce a CICN method for higher-order link prediction. Our approach integrates various order clustering coefficients using mutual information theory to predict the 3-cliques in the network. By evaluating AUC and AUPR metrics across 9 networks, experimental results demonstrate that CICN outperforms other baselines, exhibiting strong robustness. Additionally, we employ a negative sampling method to construct data, effectively addressing the imbalance issue in higher-order link prediction. The findings suggest that higher-order clustering coefficients are critical for understanding complex network structures and can significantly enhance link prediction accuracy of interactions among multi-nodes. Our method can be applied across various fields, such as social network analysis, biological network modeling, and recommendation systems, where accurate prediction of multi-node interactions is essential. The robustness of CICN also highlights its potential for application in large-scale, real-world networks with high levels of complexity.

Overall, while the CICN method makes significant progress in high-order link prediction, there is still room for improvement. On one hand, the high-order clustering coefficient primarily focuses on the local high-order features of nodes, neglecting the global structure of the network.

Consequently, the CICN method falls short in capturing complex high-order structures such as network motifs and hierarchies, and also fails to fully exploit higher-order interactions and graph structural information in heterogeneous graph link prediction tasks. In the future, we can delve deeper into these higher-order structural characteristics. By designing models capable of effectively capturing higher-order interactions and integrating multi-source information such as node attributes, to further enhance the accuracy of heterogeneous graph link prediction. On the other hand, compared to traditional link prediction methods, uncovering structural features that contribute to the formation of high-order relationships is more challenging and computationally expensive. To address this issue, we can explore various methods such as graph neural networks, topological deep learning, and simplicial complexes to fully exploit the higher-order structural information in network data and apply it to higher-order link prediction tasks.

We declare that there are no conflicts of interest in this work. Throughout the research process, we have not encountered any conflicts of interest that could have influenced the research outcomes. We are committed to upholding the highest standards of research ethics and integrity.

CRediT authorship contribution statement

Yabing Yao: Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Ziyu Ti:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis. **Zhipeng Xu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yangyang He:** Writing – review & editing, Methodology, Investigation, Formal analysis, Conceptualization. **Zeguang Liu:** Software, Methodology, Formal analysis. **Wenxiang Liu:** Supervision, Formal analysis. **Xiangzhen He:** Supervision, Funding acquisition, Formal analysis. **Fuzhong Nian:** Supervision, Formal analysis. **Jianxin Tang:** Supervision, Formal analysis.

Declaration of competing interest

We declare that there are no conflicts of interest in this work. Throughout the research process, we have not encountered any conflicts of interest that could have influenced the research outcomes. We are committed to upholding the highest standards of research ethics and integrity.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (No. 62366030, 62162040), the Gansu Provincial Natural Science Foundation, China (No. 23JRRA8222), the Higher Education Innovation Fund project of Gansu, China (No. 2022A-022, 24CXGA046), the Open Project of Key Laboratory of Linguistic and Cultural Computing Ministry of Education, China (No. KFKT202304), the Gansu Provincial Science Fund for Distinguished Young Scholars, China (No. 23JRRA766).

Data availability

The datasets analyzed during the experiments conducted in this paper can be obtained at the following URL: https://snap.stanford.edu/data/ and http://networkrepository.com/networks.php.

References

[1] Y. Yao, T. Cheng, X. Li, Y. He, F. Yang, T. Li, Z. Liu, Z. Xu, Link prediction based on the mutual information with high-order clustering structure of nodes in complex networks, Phys. A 610 (2023) 128428.

[2] S. Kumar, A. Mallik, B. Panda, Link prediction in complex networks using node centrality and light gradient boosting machine, World Wide Web 25 (2022) 2487–2513.

[3] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Phys. A 390 (2011) 1150–1170.

[4] K. Abbas, A. Abbasi, S. Dong, L. Niu, L. Yu, B. Chen, S.-M. Cai, Q. Hasan, Application of network link prediction in drug discovery, BMC Bioinform. 22 (2021) 1–21.

[5] K. Han, P. Cao, Y. Wang, F. Xie, J. Ma, M. Yu, J. Wang, Y. Xu, Y. Zhang, J. Wan, A review of approaches for predicting drug–drug interactions based on machine learning, Front. Pharmacol. 12 (2022) 814858.

[6] J. Chen, X. Lin, Y. Wu, Y. Chen, H. Zheng, M. Su, S. Yu, Z. Ruan, Double layered recommendation algorithm based on fast density clustering: Case study on yelp social networks dataset, in: 2017 International Workshop on Complex Systems and Networks, IWCSN, IEEE, 2017, pp. 242–252.

[7] M. Vahidi Farashah, A. Etebarian, R. Azmi, R. Ebrahimzadeh Dastjerdi, A hybrid recommender system based-on link prediction for movie baskets analysis, J. Big Data 8 (2021) 1–24.

[8] A.S. Pope, D.R. Tauritz, M. Turcotte, Automated design of tailored link prediction heuristics for applications in enterprise network security, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019, pp. 1634–1642.

[9] Y. Yao, R. Zhang, F. Yang, J. Tang, Y. Yuan, R. Hu, Link prediction in complex networks based on the interactions among paths, Phys. A 510 (2018) 52–67.

[10] A. Kumar, S.S. Singh, K. Singh, B. Biswas, Link prediction techniques, applications, and performance: A survey, Phys. A 553 (2020) 124289.

[11] M.M. Mayfield, D.B. Stouffer, Higher-order interactions capture unexplained complexity in diverse communities, Nat. Ecol. Evol. 1 (2017) 0062.

[12] D. Centola, J. Becker, D. Brackbill, A. Baronchelli, Experimental evidence for tipping points in social convention, Science 360 (2018) 1116–1119.

[13] A. Patania, G. Petri, F. Vaccarino, The shape of collaborations, EPJ Data Sci. 6 (2017) 1–16.

[14] E. Nasiri, K. Berahmand, M. Rostami, M. Dabiri, A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding, Comput. Biol. Med. 137 (2021) 104772.

[15] D. Shi, G. Chen, W.W.K. Thong, X. Yan, Searching for optimal network topology with best possible synchronizability, IEEE Circuits Syst. Mag. 13 (2013) 66–75, http://dx.doi.org/10.1109/MCAS.2012.2237145.

[16] D. Shi, G. Chen, Simplicial networks: A powerful tool for characterizing higher-order interactions, National Sci. Rev. 9 (2022) nwac038.

[17] H. Nassar, A.R. Benson, D.F. Gleich, Neighborhood and pagerank methods for pairwise link prediction, Soc. Netw. Anal. Min. 10 (2020) 63.

[18] N. Chavan, K. Potika, Higher-order link prediction using triangle embeddings, in: 2020 IEEE International Conference on Big Data, Big Data, IEEE, 2020, pp. 4535–4544.

[19] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.

[20] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, S. Jaiswal, graph2vec: Learning distributed representations of graphs, 2017, arXiv preprint arXiv:1707.05005.

[21] T. Papamarkou, T. Birdal, M.M. Bronstein, G.E. Carlsson, J. Curry, Y. Gao, M. Hajij, R. Kwitt, P. Lio, P. Di Lorenzo, et al., Position: Topological deep learning is the new frontier for relational learning, in: Forty-First International Conference on Machine Learning, 2024.

[22] Y. Lu, M. Gao, H. Liu, Z. Liu, W. Yu, X. Li, P. Jiao, Neighborhood overlap-aware heterogeneous hypergraph neural network for link prediction, Pattern Recognit. 144 (2023) 109818.

[23] Y. Huang, Y. Zeng, Q. Wu, L. Lü, Higher-order graph convolutional network with flower-petals laplacians on simplicial complexes, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, 2024, pp. 12653–12661.

[24] Y. Zeng, Y. Huang, Q. Wu, L. Lü, Influential simplices mining via simplicial convolutional networks, Inf. Process. Manage. 61 (2024) 103813.

[25] B. Liu, R. Yang, L. Lü, Higher-order link prediction via local information, Chaos 33 (2023).

[26] S. Piaggesi, A. Panisson, G. Petri, Effective higher-order link prediction and reconstruction from simplicial complex embeddings, in: Learning on Graphs Conference, PMLR, 2022, 55–1.

[27] R.M. Gray, Entropy and Information Theory, Springer Science & Business Media, 2011.

[28] A. Lesne, Shannon entropy: A rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics, Math. Structures Comput. Sci. 24 (2014) e240311.

[29] B. Zhu, Y. Xia, An information-theoretic model for link prediction in complex networks, Sci. Rep. 5 (2015) 13707.

[30] F. Tan, Y. Xia, B. Zhu, Link prediction in complex networks: A mutual information perspective, PLoS One 9 (2014) e107056.

[31] M.Á. Serrano, M. Boguna, Clustering in complex networks. I. General formalism, Phys. Rev. E 74 (2006) 056114.

[32] H. Yin, A.R. Benson, J. Leskovec, Higher-order clustering in networks, Phys. Rev. E 97 (2018) 052306.

[33] M.E. Newman, Clustering and preferential attachment in growing networks, Phys. Rev. E 64 (2001) 025102.

[34] L.A. Adamic, E. Adar, Friends and neighbors on the web, Soc. Netw. 25 (2003) 211–230.

[35] P.J. Etude, Comparative de la distribution florale dans une portion des alpes et des jura, Bull. Soc. Vaud. Sci. Nat. 37 (1901) 547.

[36] H. Yuliansyah, Z.A. Othman, A.A. Bakar, A new link prediction method to alleviate the cold-start problem based on extending common neighbor and degree centrality, Phys. A 616 (2023) 128546.

[37] H. Nassar, A.R. Benson, D.F. Gleich, Pairwise link prediction, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2019, pp. 386–393.

[38] J.A. Hanley, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1982) 29–36.

[39] M. Al Hasan, V. Chaoji, S. Salem, M. Zaki, Link prediction using supervised learning, in: SDM06: Workshop on Link Analysis, Counter-Terrorism and Security, vol. 30, 2006, pp. 798–805.

[40] C.D. Manning, An Introduction to Information Retrieval, Cambridge University Press, 2009.

[41] J. Davis, M. Goadrich, The relationship between precision–recall and roc curves, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 233–240.

[42] R. Rossi, N. Ahmed, The network data repository with interactive graph analytics and visualization, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, 2015.

[43] P. Patil, G. Sharma, M.N. Murty, Negative sampling for hyperlink prediction in networks, in: Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May, 2020 11–14, Proceedings, Part II 24, Springer, 2020, pp. 607–619.

[44] Z. Fang, S. Tan, Y. Wang, J. Lu, Elementary subgraph features for link prediction with neural networks, IEEE Trans. Knowl. Data Eng. (2021).

Yabing Yao is currently an associate professor at School of computer and communication, Lanzhou University of Technology. He received the Ph.D. degree in the School of Information Science and Engineering at Lanzhou University in 2017. His work has focused on link prediction in complex networks. His research interests include machine learning on graphs, network science.

Ziyu Ti received her B.S. computer science and technology from University of Xianyang Normal in 2021. She is currently a M.S. student in the School of Computer and Communication at Lanzhou University of Technology. Her research interests include link prediction and graph self supervised learning.

Zhipeng Xu received the bachelor's degree from the Lanzhou University of Arts and Science, Lanzhou, China, in 2021. He received the M.S. degree with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China, in 2024. His research interests include link prediction and higher-order link prediction.

Yangyang He received the bachelor's degree from the Chaohu University, Hefei, China, in 2020. He received the M.S. degree with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China, in 2024. He is currently pursuing Ph.D. degree in the School of Computer Science & Technology, Beijing Jiaotong University, Beijing, China. His research interests include complex network analysis, link prediction and calculation acceleration of deep neural network.

Zeguang Liu received the bachelor's degree from the Tianjin Renai college, Tianjin, China, in 2020. He received the M.S. degree with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China, in 2023. He is currently working at Qinghai Open University. His research focuses on link prediction in complex networks, with interests in hypergraphs, hyper-link prediction, knowledge graph completion, network science, and machine learning.

Wenxiang Liu is an associate professor of Internet of Things engineering in Gansu Normal University for Nationalities, graduated from Northwest Minzu University with a master's degree in 2008, mainly engaged in and studying digital protection of intangible cultural heritage.

Xiangzhen He is a professor and doctoral supervisor of Northwest Minzu University, graduated from Northwest Minzu University in 2016. Now he is the deputy director of the Key Laboratory of Linguistic and Cultural Computing Ministry of Education, and the director of the Key Laboratory of Minzu Languages and Cultures Intelligent Information Processing,Gansu Province. His main research field is multimodal human–computer interaction.

Fuzhong Nian received his B.S. degree in Physics from Northwest Normal University, Lanzhou, China, in 1998; and M.S. degree in Control Theoretics & Engineering from Lanzhou University of Technology, Lanzhou, China, in 2004, respectively. He received the Ph.D degree at Dalian University of Technology. He is currently a professor at the School of Computer and Communication, Lanzhou University of Technology. His research interests include nonlinear dynamics and control, complex networks and systems, neural networks.

Jianxin Tang received the Ph.D. degree from the School of Information Science & Engineering, Lanzhou University, Lanzhou, China, in 2019. Since 2012, he has been with the School of Computer Science and Communication Department, Lanzhou University of Technology, Lanzhou, China, where he became an Associate Professor in 2021. His current interests include intelligent algorithm & optimization, social network analysis, social computing.