Subgraph Structure Feature Learning for Triangle Clique Prediction in Complex Networks

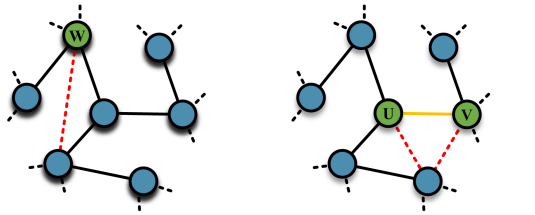
Yabing Yao, Zhiheng Mao, Yangyang He, Zhipeng Xu, Ziyu Ti, Pingxia Guo, Fuzhong Nian, and Ning Ma

***Abstract*—Link prediction is a critical task in network analysis, widely used to infer potential relationships between nodes. While traditional methods focus on pairwise interactions, real-world networks often exhibit higher-order interactions involving multiple nodes, such as 3-clique (triangle), which play a crucial role in understanding tightly-knit groups and complex network dynamics. In this paper, we propose a triangle clique prediction method based on Subgraph Structure Feature Learning (SSFL), which focuses on triangle structures in a network for prediction. In detail, it extracts the one-hop neighborhood around a target 3-clique, encodes it as a enclosing subgraph, and represents its structural features as a vector. These feature vectors are then processed using a fully connected neural network to predict 3-clique formations effectively. Experimental results show that the proposed method outperforms similarity-based link prediction methods and demonstrates comparable performance to embedding-based and machine learning-based approaches across various datasets. Our work can not only directly predict 3-clique structures in a network, but also provides insights into better understanding the evolution mechanism of networks.**

***Index Terms*—Complex networks, link prediction, clique structure, subgraph, neural network, network evolution.**

I. Introduction

LINK prediction is a fundamental problem in network analysis, aiming to predict missing links or potential new relationships between nodes in a network [1], [2], [3], [4]. Due to its strong predictive performance, it is widely applied in various domains. For instance, in social networks, link prediction can be used to recommend potential friends, thereby enhancing the social experience for users [5], [6], [7]. In e-commerce platforms, it plays a crucial role in filtering products of interest for users, which increases their dependency on the platform [8]. In biological networks, link prediction is often used in the preprocessing stage, effectively reducing the cost of biological experiments [9], [10], [11], [12].



(a) Traditional dyadic link prediction (b) Higher-order triangle clique prediction

Fig. 1. (a) Represents the traditional dyadic link prediction, i.e., recommending nodes that are likely to be linked to a given node W. (b) denotes the triangle clique prediction task studied in this paper, i.e., recommending potential node (1-clique) for a given seed edge (U, V) (2-clique) that are simultaneously linked to it.

Traditional link prediction methods focus on pairwise interactions between nodes and can be classified into three main categories: similarity-based methods (e.g., CN [13], AA [14]), embedding-based methods (e.g., Deepwalk [15], Node2vec [16]), and machine learning-based methods (e.g., SEAL [17], LGNN [18]). Nevertheless, real-world networks often involve interactions among multiple entities. For example, co-authorship networks frequently exhibit groups of collaborators [19], social networks exhibit multi-person interactions [20], and biological processes rely on multi-molecular interactions [21]. Research [22] shows that interactions among three nodes (i.e., triangles or 3-cliques) form the most fundamental and essential higher-order structures in networks. These structures are widely present across various networks and play a key role in revealing network evolution patterns and understanding their underlying mechanisms. However, traditional link prediction methods are limited to predicting pairwise relationships and cannot capture higher-order interactions within networks.

Given the importance of 3-clique structures in networks, several higher-order interaction researches [23], [24], [25] focus on predicting their formation. Specifically, the core task of triangle clique prediction is to determine which 1-cliques (nodes) are most likely to interact with given 2-cliques (edges) to form 3-cliques (triangles). To clarify the distinction between dyadic link prediction (identifying pairwise connections) and higher-order triangle clique prediction (forecasting 3-node fully-connected structures), refer to Fig. 1. In Fig. 1(a), the traditional link prediction task is illustrated, aiming to infer missing edges (2-cliques)

Received 4 April 2024; revised 26 February 2025; accepted 23 April 2025. Date of publication 5 May 2025; date of current version 25 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62366030, in part by Gansu Provincial Natural Science Foundation under Grant 23JRRA8222, in part by the Higher Education Innovation Fund project of Gansu under Grant 2022A-022, and in part by the Open Project of Key Laboratory of Linguistic and Cultural Computing Ministry of Education under Grant KFKT202304. Recommended for acceptance by Dr. Maksim Kitsak. *(Corresponding author: Yabing Yao.)*

Yabing Yao, Zhiheng Mao, Yangyang He, Zhipeng Xu, Ziyu Ti, Pingxia Guo, and Fuzhong Nian are with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China (e-mail: yaoyabing@lut.edu.cn; mzh123452021@163.com; yangyanghe2023@163.com; xuzhipeng911@gmail.com; tzy97666@163.com; 17339804896@163.com; gdnfz@lut.edu.cn).

Ning Ma is with the Key Laboratory of Linguistic and Cultural Computing Ministry of Education, Northwest Minzu University, Lanzhou 730030, China (e-mail: maning@xbmu.edu.cn).

All source code is publicly available at: https://github.com/yabingyao/SSFL4HigherOrderLinkPrediction.

Digital Object Identifier 10.1109/TNSE.2025.3566227

2327-4697 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

with respect to a given node W. In contrast, Fig. 1(b) shows triangle clique prediction, which focuses on the emergence of triangles (3-cliques). This task aims to identify the 1-cliques (nodes) in the network that are likely to connect with a given 2-clique (edge (U, V)), thereby forming a 3-clique (triangle) structure.

In response to the issue of 3-cliques prediction, Benson et al. [23] propose a pairwise link prediction method that directly predicts 3-clique by identifying 1-cliques likely to interact with a given 2-clique. Subsequently, some researches [25], [26] further explore the prediction of higher-order interactions in networks. Despite these advancements, several critical challenges remain unresolved. Most 3-cliques prediction methods are extensions of heuristic algorithms, which rely on strong prior assumptions. These methods often lack robustness and fail to deliver reliable performance across datasets from diverse domains. Moreover, it remains unclear which specific topological structures contribute to the prediction of 3-clique formations, posing a key limitation to improving the accuracy and effectiveness of 3-cliques prediction.

To address the above challenge of triangle clique prediction, we propose **S**ubgraph **S**tructure **F**eature **L**earning (SSFL), a novel method that integrates a fully connected neural network with enclosing subgraph feature learning to enhance predictive performance. Motivated by the success of traditional link prediction approaches, which demonstrate that enclosing subgraphs are strong indicators of missing links, SSFL extracts the one-hop enclosing subgraph surrounding a target 3-clique and encodes its structural properties into fixed-length feature vectors. These vectors, designed to capture critical topological patterns, are then processed by the neural network to learn effective representations for prediction. Extensive experiments on 12 real-world networks demonstrate that SSFL consistently outperforms similarity-based baselines and achieves performance comparable to state-of-the-art embedding-based and machine learning-based methods. Additionally, we analyze the influence of different topological features on prediction accuracy, providing deeper insights into their predictive value. Finally, inspired by dataset partitioning strategies in traditional link prediction, we propose a novel dataset partitioning approach specifically designed for 3-clique prediction. The contributions of this research are summarized as follows:

1) We propose a novel triangle clique prediction method, SSFL, which predicts the 1-clique in a network likely to connect with a given 2-clique. Additionally, we introduce a new dataset partitioning approach tailored for this task.
2) We investigate which topological features within the one-hop neighborhood around a target 3-clique are effective for prediction. The results indicate that comparable prediction performance can be achieved using only a few fundamental structural features.
3) We evaluate the proposed SSFL method against 14 baselines across 12 empirical networks. The results demonstrate that SSFL consistently outperforms all similarity-based baselines and achieves superior accuracy compared to embedding-based and machine learning-based approaches in most networks.

The rest of this paper is organized as follows: Section II reviews research related to link prediction and higher-order networks. Section III defines the triangle clique prediction problem studied in this paper. In Section IV, we provide a detail description of our proposed method SSFL for triangle clique prediction. Section V introduces the baselines and evaluation metrics used in experiment and Section VI discusses the experimental results. The conclusion of this paper is given in Section VII.

II. Related Work

Link prediction is a fundamental task in network analysis, widely used to uncover potential connections in complex systems. While traditional methods primarily focus on pairwise interactions, many real-world networks involve higher-order interactions among groups of nodes. To provide a comprehensive background, this section reviews traditional link prediction methods and recent advancements in higher-order networks.

A. Traditional Link Prediction

Traditional link prediction techniques can be broadly classified into three categories: similarity-based methods, embedding-based methods and machine learning-based methods [27].

Similarity-based methods estimate the likelihood of a link by computing node similarity based on topological structures. These methods are generally classified into first-order [13], [14], second-order [28], [29], and high-order [1], [30], [31] methods, depending on the level of structural information they leverage. Designed upon empirical assumptions, heuristic similarity indices are crafted to enhance interpretability. However, their reliance on predefined assumptions makes them highly sensitive to dataset-specific characteristics, often resulting in limited generalizability across different network structures.

Embedding-based methods utilize random walks to generate node embeddings, which are subsequently used for link prediction. Deepwalk [15], inspired by natural language processing, employs the Skip-gram model to capture node co-occurrence in random walk sequences. Node2vec [16] extends the Deepwalk approach by introducing tunable parameters that balance breadth-first and depth-first search strategies, enabling more flexible exploration of network topology. LINE [32] preserves both local and global structural information by optimizing probability distributions during embedding process. LP-ROBIN [33] utilizes incremental embedding based on random walks to capture network dynamics and predict new links. It can handle the addition of new nodes over time without prior knowledge. However, these methods are susceptible to sampling biases, resulting in inconsistent performance across datasets.

Machine learning-based methods leverage algorithms to extract potential features from networks and predict the likelihood of connections forming between nodes. These methods can be broadly categorized into two types [27]. The first type employs Multi-Layer Perceptrons (MLPs) as aggregators, with representative methods such as NNESF [34], which effectively utilizes elementary subgraph features to provide deeper insights into the influence of topological structures on link prediction accuracy. Building on this work, Fang et al. [35] incorporate

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

fundamental heuristic elements to further enhance predictive performance. The second type utilizes Graph Neural Networks (GNNs) as aggregators, enabling more efficient and accurate link prediction. SEAL [17] formulates the link prediction task as a graph classification problem by extracting one-hop subgraphs around target node pairs, significantly improving predictive performance. Cai et al. [18] address link prediction using line graphs, transforming it into a node classification task within the corresponding line graph to mitigate information loss caused by graph pooling. LCILP [36] introduces an inductive link prediction strategy for knowledge graphs by leveraging a Personalized PageRank-based local clustering technique, which samples subgraphs around target links to better capture local structural information. LLP [37] incorporates knowledge distillation for link prediction, integrating both rank-based and distribution-based matching strategies to enhance predictive accuracy.

However, these approaches also face inherent challenges, such as rigid structural assumptions, sampling biases, and a limited capacity to capture higher-order interactions. These limitations hinder their ability to be fully applied to complex real-world networks, where higher-order structures are crucial for understanding network dynamics.

B. Higher-Order Networks

With the growing complexity of research in network science, traditional pairwise interaction models are increasingly inadequate for capturing multi-node interactions. To address this issue, researchers have explored higher-order networks, which extend the focus from pairwise relationships to associations among groups of nodes (e.g., triples or larger sets). Battiston et al. [38], [39] provide a foundational overview of higher-order networks, highlighting their advantages over traditional binary models in characterizing the structures and dynamics of complex systems. By modeling higher-order interactions, these networks can more effectively capture intricate dependencies among nodes, enabling precise network modeling and prediction in diverse applications.

Higher-order networks have proven valuable in practical applications, particularly in settings involving multi-party collaboration or group behaviors. For example, in co-authorship networks, collaborations often involve joint efforts among multiple scholars rather than simple pairwise connections [40], [41]. Similarly, in social networks, group interactions (e.g., social circles with three or more members) are common, and modeling these higher-order structures provides deeper insights into complex social patterns and dynamics. Patania et al. [42] and Cencetti et al. [43] examine how higher-order structures shape the topology and evolution of co-authorship and social interaction networks, respectively.

Beyond specific applications, significant methodological advancements have been made in analyzing higher-order networks. Hypergraphs have become a key tool for modeling multi-node interaction structures. Benson et al. [44] introduce a centrality analysis method leveraging hypergraph features to evaluate node influence and structural importance. Lotito et al. [45] develop higher-order motif analysis for identifying complex network patterns, while Contisciani et al. [46] advance community detection and hyperedge dynamics analysis within hypergraphs. Furthermore, Eriksson et al. [47] examine the sensitivity of flow-based community detection to random-walk models and network representations. Landry et al. [48] reveal the topological intricacies of multi-node interactions through the analysis of network simpliciality. Ceria and Wang [49] uncover dynamic patterns in evolving higher-order networks by investigating their temporal-topological properties.

In the field of link prediction, several researches have explored the integration of higher-order interactions to overcome the limitations of traditional methods, which typically focus on pairwise relationships. Benson et al. [23] introduce a higher-order link prediction method based on simplicial closure, enabling better capture of higher-order interactions. Nassar et al. [24] address the triangle clique prediction problem by extending traditional link prediction algorithms and introducing the TRPR method to identify more complex network structures. In a subsequent research [25], the TRPRW method is proposed, which further refines the neighborhood information of a given 2-clique, showcasing the potential of higher-order interactions to enhance prediction accuracy. CICN [50] uses clustering mutual Information of common neighbors to predict 3-cliques. While these higher-order link prediction methods mainly build upon traditional similarity-based algorithms, their performance often varies across different types of networks, revealing the challenges of applying these methods consistently in diverse network settings.

III. Problem Definition

A. Predicting 2-Cliques of Traditional Link Prediction

The traditional link prediction algorithm can be interpreted as the prediction of 2-cliques, aiming to identify missing or future edges between two 1-cliques (nodes). Formally, given an undirected network $G = (V, E)$, where V represents the set of nodes and E denotes the set of existing edges present in the network. In addition, the set of all possible edges in the network can be defined as U. Correspondingly, the cardinality of set U (that is, the number of edges it contains) is $|U| = \frac{|V|(|V|-1)}{2}$. The objective of traditional link prediction is to infer potential links from the set of non-existent edges $U - E$.

In order to evaluate the performance of different link prediction algorithms, the edge set E in the network is usually divided into a training set E^T and a test set E^P according to a predefined ratio r, where $E^T \cap E^P = \varnothing$ and $E^T \cup E^P = E$. The goal of each link prediction algorithm is to predict the edges in E^P as accurately as possible based on the observed information available in E^T.

B. Predicting 3-Cliques of Higher-Order Triangle Clique Prediction

The goal of triangle clique prediction is to identify potential 3-cliques (triangles) in a network by determining which 1-cliques (nodes) are most likely to form a triangle given an existing 2-clique (edge). Consider an undirected network $G = (V, E, K)$,

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

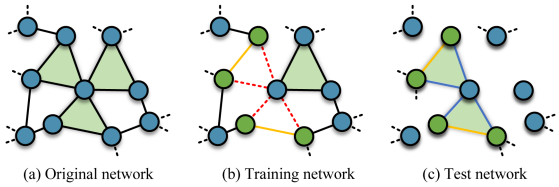


Fig. 2. The construction process of training set network and test set network. (a) In the original network, a specific proportion of 3-cliques (shaded regions) is initially selected. (b) The wedge structure are systematically removed (red dashed lines) to construct the training network. (c) For each chosen 3-clique, one of the 2-cliques is randomly selected as the seed edge (yellow lines) and construct the test network.

where V represents the set of 1-cliques (nodes), E represents the set of 2-cliques (edges), and K denotes the set of existing 3-cliques (triangles). To construct the training and test networks, we begin by selecting 3-cliques from the set K. These 3-cliques are chosen in accordance with predefined selection criteria for constructing the test network. For each selected 3-clique, one 2-clique is designated as the seed edge, and the corresponding wedge structure in the original network is removed. After constructing the test network, the remaining portion of the original network is used as the training network.

Upon completion of the construction of the training and test networks, the remaining 3-cliques within these networks are treated as positive samples. To construct the training set K^T and test set K^P, it is ensured that $K^T \cup K^P = K$ and $K^T \cap K^P = \emptyset$. Then, an equal number of negative samples are selected from the original network. Each sample—whether positive or negative—is represented as a pair $[x, (u, v)]$, where x denotes the node to be predicted, and (u, v) represents the seed edge. To ensure proper labeling, positive samples (where x forms a 3-clique with (u, v)) are assigned a label of 1, while negative samples (where x does not form a 3-clique with (u, v)) are assigned a label of 0.

The construction process of training set network and testing set network is illustrated in Fig. 2, which shows a simplified version of how we divide the dataset into the training and test networks. In the original network (Fig. 2(a)), there are three 3-clique structures. We select two of these 3-cliques to form the test networks (Fig. 2(c)). For each selected 3-clique, one of the 2-cliques is randomly chosen as the seed edge (yellow lines), while the corresponding wedge structure (blue lines) is deleted from the original network to construct the training network (Fig. 2(b)).

To ensure that each removing wedge structure does not include the seed edge, predefined selection criteria must be followed. Without such constraints, the removal process might unintentionally eliminate key seed edges, leading to incorrect test data construction and affecting the validity of the prediction model. To prevent this, the following rules are applied:

- Rule 1: If a selected 3-clique does not contain any previously chosen seed edges, one of its 2-cliques is randomly designated as the seed edge, and the corresponding wedge structure is removed.
- Rule 2: If the selected 3-clique contains only one previously selected seed edge, that edge remains the seed edge, and the corresponding wedge structure is removed.
- Rule 3: If the selected 3-clique contains two or more seed edges, the wedge removal procedure is omitted for that clique.

These rules ensure consistency in constructing K^T and K^P while maintaining a balanced dataset of positive (1) and negative (0) samples for training and evaluation.

IV. Methods

In this section, we introduce the proposed method, SSFL, for 3-clique prediction, as illustrated in Fig. 3. The framework consists of the following three components.

Enclosing subgraph is extracted around a target 3-clique: In order to infer which 1-cliques in the network are most likely to interact with a given 2-clique to form 3-clique, we extract the one-hop neighborhoods around the target 3-clique as enclosing subgraph, and calculate the likelihood of their interaction according to the topology within this subgraph.

Encodes different structural features: To investigate the influence of different topological structures on 3-clique formation, we focus on the enclosing subgraph extracted from the network. Specifically, we manually encode various structural features of the subgraph and use these as input variables for the neural network. This approach allows us to analyze how different structural characteristics impact the formation of 3-cliques within the network.

Neural network for 3-cliques prediction: Multi-Layer Perceptron takes the encoding structural features as input to predict 3-cliques.

A. Enclosing Subgraph is Extracted Around a Target 3-Clique

Traditional link prediction researches have proved that the existence of potential links is usually closely related to the topological structure around them. For example, CN [13], AA [14], RA [29], etc., use the topological features around a pair of nodes to be predicted to design similarity indexes. After that, it has been demonstrated that the h-hop enclosing subgraph around a target link is effective in learning-based approaches [17], [34]. In these approaches, link prediction is conducted by leveraging the topological features within the enclosing subgraph.

Inspired by traditional link prediction methods, this paper extracts the h-hop neighborhoods around a target 3-clique as an enclosing subgraph and investigates which structural features within this subgraph contribute to 3-clique prediction. Specifically, the target 3-clique consists of a given seed edge (u, v) and a 1-clique to be predicted x. The h-hop neighborhood around them can be represented by $\Gamma^h((u, v), x)$, which is defined as:

$$\Gamma^h((u,v),x) = \{n \in V - \{u,v,x\} | \min(d(n,u), d(n,v), d(n,x)) \leq h\}, \quad (1)$$

where $d(i, j)$ denotes the shortest path length between nodes i and j in the network.

Generally, increasing the value of h allows for capturing more structural information within the extracted subgraph. However,

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

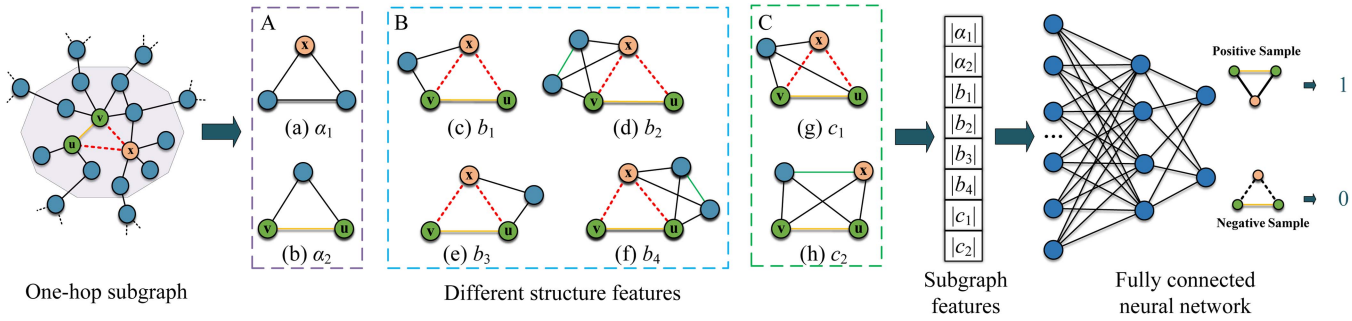


Fig. 3. Different structural features from the one-hop subgraph are used to predict 3-cliques. For a given target 3-clique, consisting of a 2-clique (u, v) and a 1-clique x, the surrounding one-hop subgraphs are extracted. Eight distinct structural features are encoded into fixed-length feature vectors. A Multi-Layer Perceptron is then used to learn the nonlinear relationships between these features and predict the probability of potential 3-cliques.

this also introduces additional noise, potentially affecting predictive performance [34]. To strike a balance between information richness and computational efficiency while mitigating noise, we focus on extracting the one-hop neighborhood around the target 3-clique as the enclosing subgraph.

B. Encodes Different Structural Features

For the extracted enclosing subgraph, our primary motivation is to determine which structural features within the subgraph are effective for predicting potential 3-cliques. In traditional link prediction tasks, Fang et al. [34] encode fundamental structural features of the subgraph to facilitate link prediction. This approach not only reduces computational complexity but also eliminates the need to fix the size of the subgraph. Inspired by this method, in this paper, we extract various structural features centered around a given seed edge 2-clique (u, v) and 1-clique x to predict 3-clique.

Specifically, as illustrated in Fig. 3, we extract eight distinct structural features within the subgraph surrounding the given 2-clique (u, v) and the 1-clique x. These features are categorized into three groups: A, B, and C, which will be introduced in detail in the following sections.

In class A, it contains two sets of structures a_1 and a_2 (corresponding (a) and (b) in Fig. 3, respectively). In detail, a_1 represents the 3-clique structure containing the 1-clique x. We assume that the more such structures exist, the higher the likelihood that the 1-clique x will connect with the given 2-clique (u, v) to form a 3-clique. We count the number of this structure within subgraph and denote it as $|a_1|$. Similarly, for structure set a_2, which represents the 3-clique structure containing the given seed edge 2-clique (u, v), we assume that the more such structures exist, the more likely the 2-clique (u, v) is to connect with the 1-clique x to form a 3-clique. The number of this structure within the subgraph is denoted as $|a_2|$.

Class B focuses on the interaction between the 1-clique x and the two endpoints of the given 2-clique (u, v). It consists of four structural feature sets: b_1, b_2, b_3, and b_4 (corresponding to (c), (d), (e), and (f) in Fig. 3, respectively). Specifically, b_1 measures the set of common neighbors between node x and endpoint v. We assume that the more common neighbors they share, the higher the likelihood of an edge forming between them. The number of common neighbors is denoted as $|b_1|$. Structure set b_2 further examines the common neighbor interactions between node x and endpoint v (shown by the solid green line in Fig. 3(d)), which also indicates the potential interactions between node x and v. The number of such edges is represented by $|b_2|$. Similarly, structure sets b_3 and b_4 focus on the interaction between node x and endpoint u. $|b_3|$ represents the number of common neighbors between x and u, while $|b_4|$ counts the edges connecting those common neighbors.

For class C, it includes two sets of structures, c_1 and c_2 (corresponding to (g) and (h) in Fig. 3, respectively). Structure c_1 counts the nodes within the subgraph that are neighbors of both the 1-clique x and the two endpoints of the 2-clique (u, v). We assume that the more such nodes there are, the higher the likelihood of the 1-clique x interacting with the two endpoints of the 2-clique to form a 3-clique. This count is denoted as $|c_1|$. c_2 represents the set of edges between 1-cliques that can form 3-cliques with the given seed edge. Additionally, it considers the edges connecting these cliques (shown by the solid green line in Fig. 3(h)). The number of such connecting edges is denoted as $|c_2|$.

These 8 different structural features are manually encode as an 8-dimensional feature vector, denoted as $\mathbf{f} = (|a_1|, |a_2|, |b_1|, |b_2|, |b_3|, |b_4|, |c_1|, |c_2|)$ which can be regarded as extracted subgraph features. These structures we selected are all conducive to the prediction of potential triangles, and each of these structural features can be used as an indicator to predict the formation of triangles.

C. Neural Network for 3-Cliques Prediction

In this paper, we focus on predicting 3-cliques that have not yet appeared in the network. Specifically, we aim to predict which 1-clique is most likely to interact with a given 2-clique to form a 3-clique. During training, the 3-cliques present in the training network are considered positive samples, while those absent from the original network are treated as negative samples. The predicting 3-cliques problem is framed as a binary classification task, where a positive sample indicates the presence of a 3-clique between the predicted 1-clique and the given 2-clique, and a negative sample indicates its absence. Positive samples are labeled as 1, and negative samples as 0. By constructing

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Method of SSFL for 3-cliques predication.

Input: Original network G; the given 2-clique (u, v) and 1-clique x to be predicted.

Output: Predicted probability $\hat{y}$ that 1-clique x interacts with the 2-clique (u, v) to form a 3-clique.

1: Extract one-hop neighborhoods around the 1-clique x and the 2-clique (u, v), constructing enclosing subgraphs.
2: Encode eight distinct structural features within the enclosing subgraph as f.
3: Use a Multi-Layer Perceptron (MLP) with f as input to predict the probability $\hat{y}$ of forming a 3-clique.

subgraph features and corresponding labels, we solve the 3-clique prediction task using a binary classification model.

For this task, we employ a Multi-Layer Perceptron (MLP). The choice of MLP is motivated by its ability to learn complex nonlinear relationships within the network, thereby improving both performance and generalization. Furthermore, compared to other deep neural network models, MLPs offer lower computational complexity and faster inference speeds. The time complexity of the fully connected neural network is $O(|\mathbf{f}| \cdot m)$, where $|\mathbf{f}|$ represents the number of input features and m denotes the number of neurons per layer. The input feature vector $\mathbf{f}$ is fed into the MLP, which can be characterized by the function $\mathcal{M}(\cdot)$. The MLP takes $\mathbf{f}$ as an input and maps it to a predicted output $\hat{y}$, which represents the predicted probability of the target 3-clique. The operation can be formulated as follows:

$$\hat{y} = \mathcal{M}(\mathbf{f}). \tag{2}$$

To evaluate the performance of the binary classification model, we use the binary cross-entropy loss function. The supervised loss $\mathcal{L}_{sup}$ between the predicted labels $\hat{y}$ and the true labels y is calculated as follows:

$$\mathcal{L}_{sup}(\hat{y}, y) = y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}). \tag{3}$$

Overall, the pseudocode of method SSFL is showed in Algorithm 1.

V. Experimental Setup

A. Baselines

In Ref. [24] and Ref. [25], Nassar et al. have also investigated the same 3-cliques prediction problem, extending traditional link prediction methods and proposing their own algorithms, TRPR and TRPRW. In this paper, we select 14 baselines to compare with SSFL, including TRPR,TRPRW, and extensions of the 10 traditional link prediction algorithms. In addition, we also extend CNDP [51], the embedding-based method Node2vec [16] and the machine learning-based methods NNESF [34] and HELF [35] to the task of predicting 3-cliques.

In order to extend traditional link prediction methods, the neighbor set of an edge (u, v) are defined as the union of its two endpoint neighbors but excluding nodes u and v, which is defined as follows:

$$\Gamma_{((u,v))} = \Gamma_{(u)} \cup \Gamma_{(v)} \backslash \{u, v\}, \tag{4}$$

where $\Gamma_{(u)}$ and $\Gamma_{(v)}$ represents the neighbor set of nodes u and v, respectively.

1) Common Neighbors (CN) [13]: For a given 2-clique (u, v) and 1-clique x to be predicted, if they have a greater number of common neighbors, they are more likely to interact to form a 3-clique, which can be defined by:

$$S^{CN}_{((u,v),x)} = \left|\Gamma_{((u,v))} \cap \Gamma_x\right|, \tag{5}$$

where $\Gamma_{((u,v))}$ and Γ_x represents the neighbor sets of edge (u, v) and node x, respectively.

2) Adamic–Adar (AA) [14]: On the basis of CN, AA takes into account the contribution of each common neighbor. Specifically, the neighbor with a smaller degree contributes more to the composition of triangles. It is defined by:

$$S^{AA}_{((u,v),x)} = \sum_{z \in \Gamma_{((u,v))} \cap \Gamma_x} \frac{1}{\log(|\Gamma_z|)}, \tag{6}$$

where z is the common neighbor of edge (u, v) and node x, and $|\Gamma_z|$ is the degree of node z.

3) AA-MAX and AA-MUL [25]: On the basis of AA, AA-max and AA-MUL consider the relationship between (u, x) and (v, x) respectively, which can be defined as:

$$S^{AA-MAX}_{((u,v),x)} = \max\left(\sum_{\delta \in \Gamma_u \cap \Gamma_x} \frac{1}{\log(|\Gamma_\delta|)}, \sum_{\eta \in \Gamma_v \cap \Gamma_x} \frac{1}{\log(|\Gamma_\eta|)}\right), \tag{7}$$

$$S^{AA-MUL}_{((u,v),x)} = \sum_{\delta \in \Gamma_u \cap \Gamma_x} \frac{1}{\log(|\Gamma_\delta|)} \bullet \sum_{\eta \in \Gamma_v \cap \Gamma_x} \frac{1}{\log(|\Gamma_\eta|)}, \tag{8}$$

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

4) Jaccard Similarity (JS) [52]: JS normalizes the common neighbors of seed edges (u, v) and node x, which can be defined by:

$$S^{JS}_{((u,v),x)} = \frac{|\Gamma_{((u,v))} \cap \Gamma_x|}{|\Gamma_{((u,v))} \cup \Gamma_x|}. \tag{9}$$

5) JS-MAX and JS-MUL [25]: Similarly, for seed edge (u, v) and node x, JS-MAX and JS-MUL can be defined as:

$$S^{JS-MAX}_{((u,v),x)} = \max\left(\frac{|\Gamma_u \cap \Gamma_x|}{|\Gamma_u \cup \Gamma_x|}, \frac{|\Gamma_v \cap \Gamma_x|}{|\Gamma_v \cup \Gamma_x|}\right), \tag{10}$$

$$S^{JS-MUL}_{((u,v),x)} = \frac{|\Gamma_u \cap \Gamma_x|}{|\Gamma_u \cup \Gamma_x|} \bullet \frac{|\Gamma_v \cap \Gamma_x|}{|\Gamma_v \cup \Gamma_x|}. \tag{11}$$

6) Resource Allocation (RA) [29]: Compared with AA, RA has increased the punishment on the common neighbors with small degree. It is defined by:

$$S^{RA}_{((u,v),x)} = \sum_{z \in \Gamma_{((u,v))} \cap \Gamma_x} \frac{1}{|\Gamma_z|}. \tag{12}$$

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

7) Common Neighbors Degree Penalization (CNDP) [51]: We extend the CNDP method by incorporating additional structural information. Beyond considering the common neighbors shared between the given 2-clique (u, v) and the target 1-clique x, CNDP also considers the direct links connecting them, which is defined by:

$$S^{CNDP}_{((u,v),x)} = \sum_{z \in \Gamma_{((u,v))} \cap \Gamma_x} \mid T_z \mid (\mid \Gamma_z \mid)^{-\beta C}, \tag{13}$$

where z is the common neighbor between the given 2-clique (u, v) and 1-clique x to be predicted, and $\mid T_z \mid$ is the number of links between them. C is the average clustering coefficient of the network, and β is the hyperparameter.

8) Node2vec [16]: Node2vec is also extended to the task of predicting 3-cliques in this paper. Specifically, the vector of the given 2-clique can be represented as the average of the two endpoint vectors, the distance between the 1-clique to be predicted and the given 2-clique in the latent vector space is used to predict the 3-clique. The distance between them can be expressed as:

$$d = \|vec_{(u,v)} - vec_x\|_2, \tag{14}$$

where $vec_{(u,v)}$ represents the vector of the given 2-clique (u, v) and vec_x represents the vector of the node to be predicted.

9) TRPR and TRPRW: Inspired by PageRank, Nassar et al. propose their own methods TRPR and TRPRW based on the random walk strategy, and the specific algorithm process can be found Ref. [25].

10) NNESF [34]: This approach directly extracts fundamental subgraph features from the 1-hop neighborhood of the target 3-clique. The subgraph is partitioned into distinct communities, and the feature vector is constructed based on the number of 1-cliques and 2-cliques within each community. Unlike heuristic methods, this method does not depend on predefined assumptions, making it more flexible and data-driven.

11) HELF [35]: Similar to NNESF, this method extracts fundamental heuristic features, such as node degrees, the number of common neighbors, and their related derivative features. After that, these extracted features are fed into a classifier to make predictions about the target variables.

B. Metrics

In this paper, the area under the receiver operating characteristic curve (AUC) and the area under the PR curve (AUPR) are used to evaluate the performance of different baseline methods.

1) AUC [53]: refers to the area under the ROC (Receiver Operating Characteristic) curve. The ROC curve is plotted based on TPR (True Positive Rate) and FPR (False Positive Rate). AUC can be used for the evaluation of 3-cliques prediction task. Specifically, based on the information observed in K^T, we calculate the prediction score for all potential 3-cliques in $T - K^T$, which consists of K^P and $T - K$. T and K represent the set of all possible 3-cliques and the set of all existing 3-cliques in the original network, respectively. The value of ROC can be calculated as:

$$TPR = \frac{TP}{TP + FN}, \tag{15}$$

$$FPR = \frac{FP}{FP + FN}, \tag{16}$$

where TP (True Positive) refers to the number of existing 3-cliques in the network that are correctly predicted as true by the model, FN (False Negative) represents the number of existing 3-cliques that are incorrectly predicted as false, FP (False Positive) indicates the number of non-existent 3-cliques that are incorrectly predicted as true, and TN (True Negative) signifies the number of non-existent 3-cliques that are correctly predicted as false. Once the ROC curve is constructed, the area under the curve (AUC) is used to quantify the model's performance. The AUC value is computed as follows:

$$AUC = \int_0^1 TPR(\theta) d(FPR(\theta)), \tag{17}$$

where the variable θ refers to the decision threshold that is used to classify predictions as positive or negative. The AUC value typically falls within the range [0.5, 1], where a higher AUC indicates better model performance and greater predictive accuracy.

2) AUPR [54]: the 3-cliques prediction can be treated as a binary classification task. Therefore, the Area Under the Precision-Recall curve (AUPR), commonly employed as an evaluation metric for classification tasks, can be utilized to assess the performance of each algorithm. AUPR is constructed with Recall on the X-axis and Precision on the Y-axis. Specifically, Recall and Precision can be defined as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{18}$$

$$Recall = \frac{TP}{TP + FN}. \tag{19}$$

The meanings of TP, FP, and FN are consistent with those used in the calculation of AUC. The PR curve (Precision-Recall curve) is plotted with Recall on the horizontal axis and Precision on the vertical axis. Once the PR curve is generated, the AUPR is calculated as the area under this curve. The AUPR serves as a performance metric similar to AUC but specifically evaluates the trade-off between precision and recall, particularly useful when dealing with imbalanced datasets. The AUPR value can be calculated as:

$$AUPR = \int_0^1 Precision(\theta) d(Recall(\theta)). \tag{20}$$

C. Datasets

We evaluate the SSFL method using 12 datasets of varying sizes from diverse fields. Table I presents the topological structure information associated with these datasets, along with the respective fields they belong to.

Chebyshev [55] is a network based on Chebyshev polynomials in graph algorithms. The nodes represent computation steps, and the edges represent computation paths. Trial [55]

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

TABLE I
THE BASIC TOPOLOGICAL FEATURES OF 12 NETWORKS

Network	$\|V\|$	$\|E\|$	$\langle k\rangle$	CC	Tri	$Field$
Chebyshev	261	1542	11.82	0.88	3831	Mathematics
Trial	928	4626	9.97	0.18	9239	Healthcare
Olm	1000	3996	3.99	0.54	998	Physics
145bit	1002	11251	22.46	0.15	13655	Information
SmaGri	1024	4916	9.60	0.31	5694	Information
Rail	1357	5171	5.62	0.48	2452	Transportation
Delaunay	2048	6127	5.98	0.44	4104	Energy
Erdos	5094	7515	2.95	0.08	1610	Social
DD21	5736	14240	4.97	0.46	9264	Label
Ca-HepTh	9877	25998	5.26	0.47	28339	Coauthor
Grid-Human	9527	62364	6.61	0.11	17192	Biological
Ca-CondMat	23133	93497	8.08	0.63	173361	Coauthor

$|V|$ and $|E|$ represent the number of nodes and edges in a network, respectively. $\langle k\rangle$ is the average degree, *CC* is the clustering coefficient, *Tri* is the number of triangles present in the network and *Field* corresponds to the domain to which the dataset belongs.

is used in health research, where the nodes represent doctors or patients, and the edges represent interactions between them. Olm [55] is a fluid dynamics network, where the nodes represent fluid elements, and the edges represent interactions between these fluid elements. 145Bit [55] network consists of 145-bit binary numbers, where each node is a 145-bit binary number, and the edges are determined based on a specific metric. SmaGri [55] is a network generated by the Pajek software. Rail [55] is a transportation network, where the nodes represent different railways, and the edges represent connections between them. Delaunay [55] is an electrical network, where the nodes represent power plants, and the edges represent the flow of electricity between power plants. Erdos [55] is a social network, where the nodes represent users, and the edges represent interactions between users. DD21 [55] is a labeled network. Ca-HepTh [55] is an academic collaboration network in the HepTh field, where the nodes represent authors, and the edges represent collaboration relationships. Grid-Human [55] is a biological network, where the nodes represent proteins, and the edges represent interactions between proteins. Ca-CondMat [55] is an academic collaboration network in the CondMat field, where the nodes represent authors, and the edges represent collaboration relationships. All of the above networks can be downloaded from the websites https://networkrepository.com/index.php and https://snap.stanford.edu/data/. Note that all datasets used in this paper are undirected and unweighted networks.

VI. EXPERIMENT RESULT

In this section, we comprehensively evaluate the performance of our proposed method SSFL against other baselines by answering the following questions:

- *Q1*: How does SSFL compare with other baselines in terms of AUC and AUPR?
- *Q2*: How robust are SSFL and the baseline models across different proportions of training data?
- *Q3*: How do the different structural features in the enclosing subgraph contribute to the 3-cliques prediction?
- *Q4*: How does the reduction of structural features impact prediction accuracy?

A. The Performance of Different Methods to Predict 3-Cliques (Q1)

We compare the prediction accuracy of different methods in 12 networks, the AUC and AUPR results are presented in Tables II and III, respectively, where the best results are highlighted in bold and the second-best are shown by underline. It is worth noting that 90% of all existing 3-cliques in the network are used as training data and the remaining 10% as test data. In addition, we set the hyperparameter β in CNDP method to 1.84. For Node2vec approach, the hyperparameters p and q are set to 0.25 and 4, respectively, the walk length is 40 and the dimension of nodes is 128.

For AUC, SSFL achieves the best results on datasets such as Trial, Olm, Rail, Delaunay, Ca-HepTh, Grid-Human, and Ca-CondMat. For AUPR, SSFL also excels, particularly on datasets like Chebyshev, Trial, Olm, 145 b, Rail, Ca-HepTh, Grid-Human and Ca-CondMat, where it achieves the highest scores. On other datasets, SSFL's performance is very close to the best-performing methods. These results indicate that SSFL consistently delivers strong performance across various datasets, highlighting its ability to capture complex relationships and structural information between nodes, as well as its broad adaptability to diverse network data.

We also compare SSFL with other baseline methods, using the AUC metric as an example. The results show that SSFL significantly outperforms traditional heuristic approaches. For instance, in the Chebyshev dataset, the heuristic method CN achieves an AUC of 0.6892, while SSFL reaches an AUC of 0.8132. Heuristic methods rely on simple assumptions (such as the number of common neighbors or similarity-based features), which limit their ability to capture complex nonlinear relationships and higher-order dependencies. SSFL also outperforms embedding-based methods like Node2vec in most cases. In the Olm dataset, Node2vec achieves an AUC of 0.9640, while SSFL achieves 0.9925, demonstrating a significant improvement. Node2vec captures only shallow structural information and struggles with deeper relationships. Furthermore, SSFL surpasses machine learning-based methods such as NNESF and HELF. In the Trial dataset, NNESF achieves an AUC of 0.8952, and HELF achieves 0.8875, while SSFL reaches 0.9534. NNESF and HELF, which are adapted from traditional link prediction methods, do not specifically address higher-order structural information. In contrast, SSFL effectively captures higher-order interactions between the target 1-clique and the given 2-clique, highlighting its superior ability to handle complex structural relationships.

Finally, we analyze the comparison between baseline methods. Node2vec demonstrates comparable performance on most datasets. It achieves the highest AUC score on one dataset and the second-best performance on two others. In terms of AUPR, it ranks first on two datasets and second on one. This superior performance is likely attributed to the random walk strategy employed by Node2vec, which enables it to capture a broader range of structural information, making it particularly effective for 3-clique prediction. For the TRPR and TRPRW methods, their ability to aggregate structural information beyond the single-hop

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

TABLE II
COMPARISON OF PREDICTION ACCURACY UNDER THE AUC INDICATOR OF 15 METHODS ON 12 NETWORKS

Datasets	CN	AA	AA-MAX	AA-MUL	JS	JS-MAX	JS-MUL	RA	TRPR	TRPRW	CNDP	Node2vec	NNESF	HELF	SSFL
Chebyshev	0.6892	0.8095	0.8050	0.8046	0.4299	0.5508	0.3764	**0.8263**	0.6572	0.6838	0.7577	0.5715	0.7577	0.7892	0.8132
Trial	0.9286	0.9298	0.9222	0.8692	0.8664	0.8153	0.8422	0.9221	0.9356	0.9408	0.8038	0.4090	0.8952	0.8875	**0.9534**
Olm	0.9538	0.9543	0.9539	0.6211	0.9553	0.9542	0.6211	0.9543	0.4338	0.9219	0.4993	0.9640	0.9656	0.9721	**0.9925**
145bit	0.7959	0.8223	0.8200	0.7127	0.5118	0.4634	0.5703	**0.8357**	0.7123	0.7138	0.6621	0.4465	0.7389	0.7874	0.8297
SmaGri	0.9175	**0.9263**	0.9218	0.8281	0.8643	0.8353	0.8074	0.9240	0.8491	0.8684	0.7797	0.7570	0.8809	0.8733	0.9142
Rail	0.9672	0.9677	0.9666	0.8105	0.9683	0.9669	0.8104	0.9677	0.8010	0.9525	0.5856	0.9851	0.9911	0.9746	**0.9949**
Delaunay	0.9763	0.9764	0.9754	0.8246	0.9773	0.9763	0.8252	0.9764	0.8430	0.9797	0.5259	0.9941	0.9936	0.9921	**0.9964**
Erdos	0.8965	0.8972	0.8971	0.7169	0.8941	0.8931	0.7161	0.8971	0.8738	**0.9087**	0.6765	0.8436	0.9081	0.8992	0.9015
DD21	0.9038	0.9038	0.9038	0.7118	0.9040	0.9040	0.7118	0.9038	0.6872	0.9667	0.5984	**0.9674**	0.9264	0.9283	0.9521
Ca-HepTh	0.9128	0.9217	0.9157	0.7268	0.9179	0.9101	0.7231	0.9137	0.7321	0.9517	0.5983	0.9634	0.9631	0.9542	**0.9832**
Grid-Human	0.8531	0.8672	0.8674	0.8281	0.8643	0.9060	0.7181	0.9087	0.7123	0.7138	0.6621	0.8636	0.8993	0.8724	**0.9132**
Ca-CondMat	0.8981	0.9017	0.9037	0.7048	0.8919	0.9120	0.7382	0.9123	0.6372	0.9417	0.6983	0.8934	0.9231	0.9335	**0.9541**

The training set is split independently at a ratio of 9:1, each result represents the average value over 50 independent implementations. The best results is marked in bold, and the second-best are underlined.

TABLE III
COMPARISON OF PREDICTION ACCURACY UNDER THE AUPR INDICATOR OF 15 METHODS ON 12 NETWORKS

Datasets	CN	AA	AA-MAX	AA-MUL	JS	JS-MAX	JS-MUL	RA	TRPR	TRPRW	CNDP	Node2vec	NNESF	HELF	SSFL
Chebyshev	0.7011	0.8133	0.8044	0.8555	0.5756	0.6679	0.4184	0.8272	0.7457	0.7599	0.7755	0.6458	0.8120	0.8531	**0.8634**
Trial	0.9196	0.9303	0.9180	0.8718	0.8326	0.7196	0.8046	0.9195	0.9248	0.9294	0.8003	0.4151	0.9226	0.9145	**0.9518**
Olm	0.9513	0.9532	0.9524	0.6215	0.9551	0.9525	0.6211	0.9532	0.5998	0.9457	0.5038	0.9778	0.9652	0.9736	**0.9889**
145bit	0.8005	0.8392	0.8335	0.7575	0.5575	0.5025	0.4901	0.8478	0.7809	0.7851	0.6830	0.4473	0.7663	0.7724	**0.8577**
SmaGri	0.9081	**0.9286**	0.9217	0.8348	0.8469	0.7871	0.7804	0.9248	0.8597	0.8737	0.7809	0.7445	0.9048	0.9091	0.9282
Rail	0.9651	0.9651	0.9632	0.8094	0.9671	0.9633	0.8090	0.9655	0.8587	0.9559	0.5831	0.9894	0.9910	0.9889	**0.9943**
Delaunay	0.9746	0.9749	0.9722	0.8227	0.9771	0.9748	0.8253	0.9749	0.8889	0.9751	0.5256	**0.9956**	0.9901	0.9912	0.9951
Erdos	0.8952	0.8984	0.8981	0.7173	0.8932	0.8868	0.7133	0.8980	0.9026	**0.9343**	0.6770	0.8636	0.8926	0.9132	0.9337
DD21	0.9034	0.9035	0.9035	0.7117	0.9040	0.9040	0.7119	0.9035	0.8036	0.9798	0.5984	**0.9809**	0.9153	0.9344	0.9600
Ca-HepTh	0.9040	0.9295	0.9132	0.7252	0.9164	0.9201	0.7145	0.9051	0.6795	0.9598	0.5898	0.9545	0.9720	0.9632	**0.9753**
Grid-Human	0.8896	0.9096	0.9124	0.8201	0.8556	0.9147	0.7265	0.8999	0.7037	0.7224	0.6710	0.8843	0.8737	0.8635	**0.9220**
Ca-CondMat	0.8965	0.8931	0.8943	0.7128	0.8998	0.9140	0.7097	0.9008	0.6450	0.9331	0.6901	0.8848	0.9131	0.9248	**0.9627**

The training set is split independently at a ratio of 9:1, each result represents the average value over 50 independent implementations. The best results is marked in bold, and the second-best are underlined.

neighborhood leads to better prediction accuracy compared to other Similarity-based methods. Among all similarity-based baseline methods, AA and RA performed better in terms of AUC and AUPR by considering the number of common neighbors. Meanwhile, NNESF and HELF demonstrate balanced performance across all datasets. Their competitive results highlight the strong generalization capability of neural network-based models in capturing complex structural patterns.

B. Robustness Analysis (Q2)

In order to analyze the robustness of different methods, we vary the training set proportion p across 0.6, 0.7, 0.8, and 0.9 and evaluate the prediction accuracy of each method. The corresponding AUC and AUPR results are presented in Figs. 4 and 5, respectively. It is worth noting that the hyperparameter settings for CNDP and Node2vec remain consistent with those in Section VI-A.

Overall, as the training set proportion p increases, the available structural information in the network expands. Consequently, most methods demonstrate improved AUC and AUPR scores. For our proposed method, SSFL's ability to integrate diverse structural features within one-hop neighborhoods contributes to its exceptional performance across different training set proportions, particularly on Olm, Ca-CondMat, and Ca-HepTh. For the AUC results, SSFL exhibits a clear performance advantage, consistently outperforming all similarity-based link prediction methods across most datasets. Meanwhile, compared to embedding-based methods such as Node2vec, SSFL achieves superior results on the majority of datasets. Furthermore, even when compared to NNESF and HELF, SSFL maintains its advantage, achieving optimal performance in both AUC and AUPR. When the training set ratio is 0.6, meaning that available network information is highly limited, SSFL surpasses all baselines in nearly all networks except DD21 and Delaunay, demonstrating strong robustness. This can be attributed to its adaptive learning capabilities, which allow it to effectively adjust weights and biases during training, maintaining high prediction accuracy even in data-scarce scenarios. Thereby achieving effective generalization from the training data and learning and grasping the stable representation of the network structure. Therefore, even when making predictions in sparse networks, this method can maintain a relatively high prediction accuracy.

For the baselines, TRPRW and Node2vec achieve outstanding performance, particularly on the Delaunay and DD21 networks, where they even outperform SSFL. This is likely due to their ability to capture long-distance structural information, with Node2vec leveraging neural networks for node representation learning. Compared to other similarity-based methods, TRPRW

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

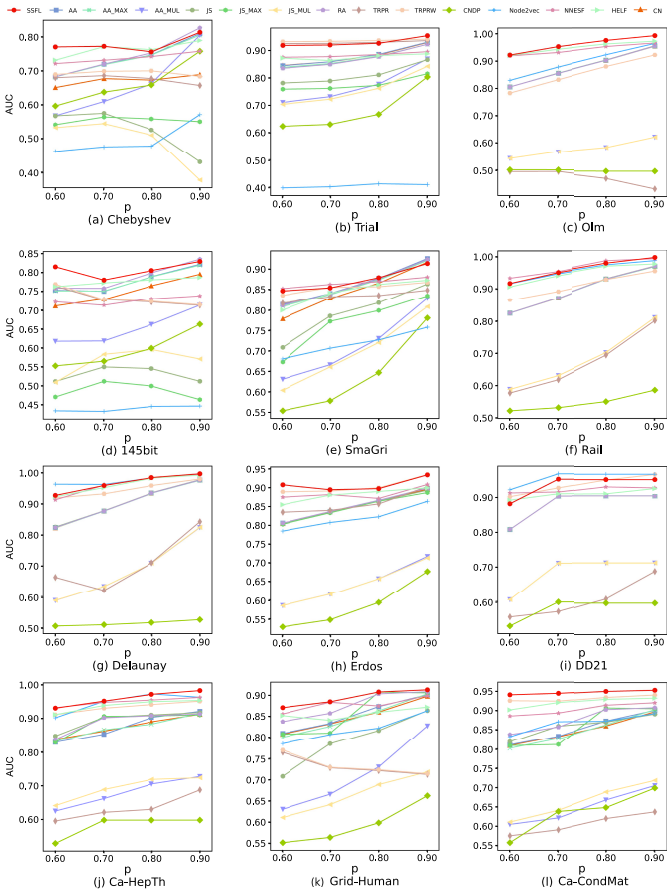


Fig. 4. The AUC results of 15 methods under different proportional training sets on 12 networks. X-axis is the training set ratio from 0.6 to 0.9, Y-axis is the AUC result.

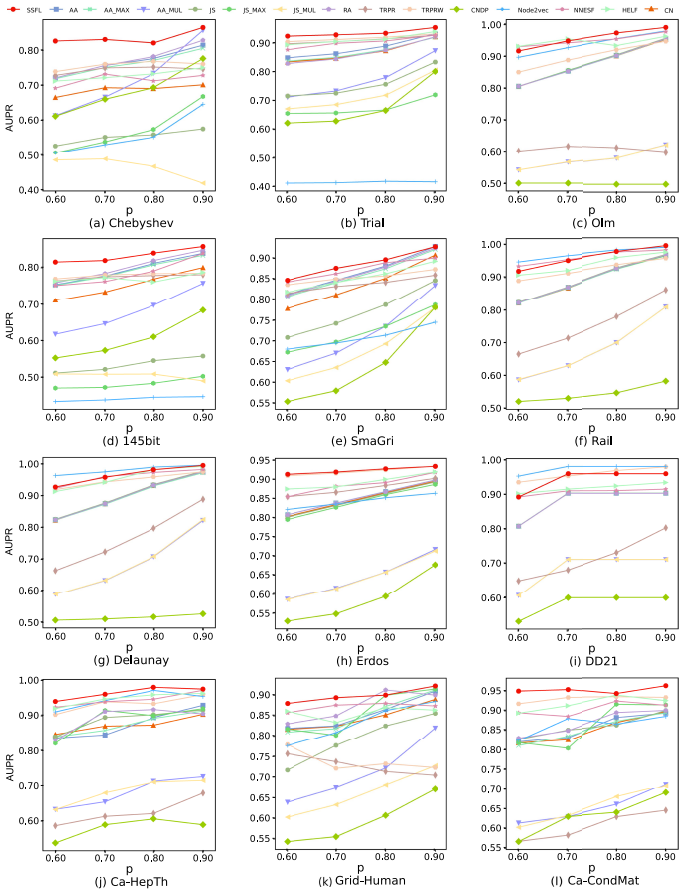


Fig. 5. The AUPR results of 15 methods under different proportional training sets on 12 networks. X-axis is the training set ratio from 0.6 to 0.9, Y-axis is the AUPR result.

and Node2vec extract richer structural information surrounding 3-cliques, resulting in performance comparable to SSFL. Meanwhile, NNESF and HELF exhibit consistent and robust performance across all datasets, with minimal performance variation compared to other methods. Their effectiveness can be attributed to the strong learning capabilities of neural networks, which allow them to capture reliable structural features. However, these approaches do not focus on 3-clique prediction when constructing structural information features, resulting in their relatively lower performance compared to SSFL. All similarity-based algorithms rely on network structural information for 3-clique prediction, and their performance improves significantly with an increasing training set proportion. Among these, AA and RA achieve stronger performance, particularly on networks such as 145-Bit and SmaGri. Additionally, we observe that AA-MUL and JS-MUL consistently underperform compared to AA-MAX and JS-MAX across nearly all networks, suggesting that multiplying two scores fails to effectively capture triangle formation patterns, which are essential for 3-clique prediction.

C. The Influence of Different Structural Features on Accuracy (Q3)

For the SSFL method, we extract various structural features from the one-hop neighborhoods around the target 3-clique for learning. In this section, we explore how different structural features affect the accuracy of 3-clique prediction. Specifically, we use the Delaunay network as an example and apply Shapley Additive Explanations (SHAP) [56] to analyze each feature's contribution to AUC performance, as shown in Fig. 6. The vertical axis represents the 8 extracted features within the subgraph,

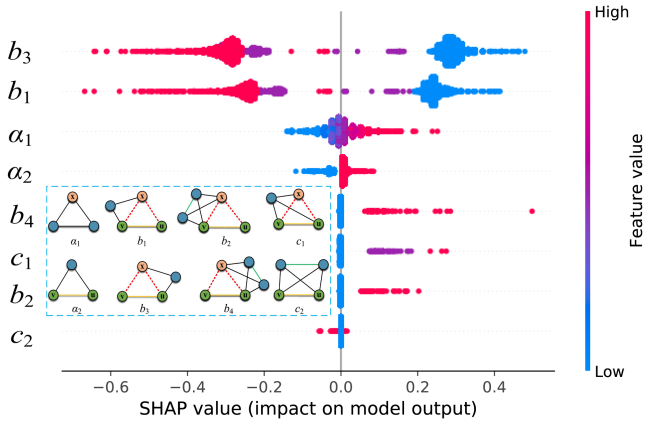


Fig. 6. SHAP summary of 8 different structure features for Delaunay network. The X-axis is the SHAP values, where positive values represent positive impact on the prediction accuracy, negative values represents negative effect, X-axis denote different structural features.

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

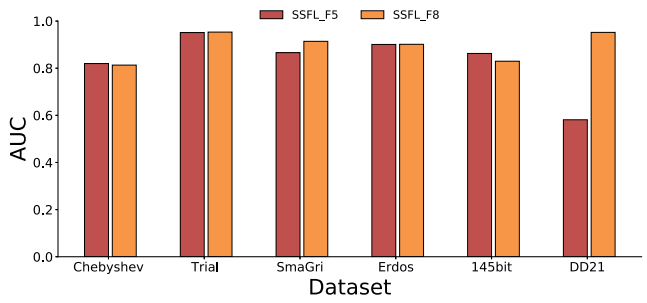


Fig. 7. AUC results of methods SSFL_F5 and SSFL_F8 on 6 networks. SSFL_F5 represents that only 5 features (a_1, a_2, b_2, b_4, c_1) are used for triangle prediction, and SSFL_F8 means that the initial 8 features are used for prediction. X-axis shows the different networks, Y-axis is the AUC result.

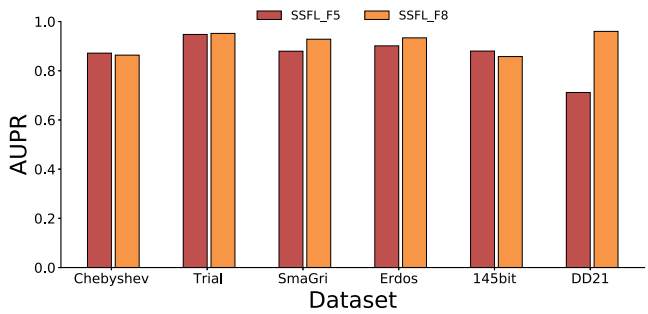


Fig. 8. AUPR results of methods SSFL_F5 and SSFL_F8 on 6 networks. SSFL_F5 represents that only 5 features (a_1, a_2, b_2, b_4, c_1) are used for triangle prediction, and SSFL_F8 means that the initial 8 features are used for prediction. X-axis shows the different networks, Y-axis is the AUPR result.

ranked by their importance, with the most influential feature at the top. The horizontal axis shows the SHAP values, where positive values indicate a favorable impact on triangle clique prediction accuracy, while negative values suggest a detrimental effect. Each data point corresponds to a sample, and the color gradient from blue to red indicates the range of structural feature values, from low to high.

As seen in Fig. 6, features b_3 and b_1 have the greatest influence on the predicted AUC results. Specifically, b_3 focuses on the potential link between the 1-clique and one endpoint of the given 2-clique, while b_1 represents the potential link to the other endpoint. The analysis shows that smaller values of these two features positively impact prediction accuracy, whereas larger values tend to have a negative effect.

Although features a_1, a_2, b_4, c_1, and b_2 are not the most influential on AUC performance, their impact gradually increases as their values rise. Specifically, a_1 and a_2 account for the number of 3-cliques in the one-hop neighborhoods containing the 1-clique and the given 2-clique, respectively. Higher values of a_1 and a_2 indicate a greater number of 3-cliques within the subgraph, increasing the likelihood of 3-clique formation. Similarly, b_2 and b_4, which measure the number of connected edges between common neighbors based on b_1 and b_3, suggest a higher probability of forming a 3-clique when their values are larger. Feature c_1, which represents the number of common neighbors between the predicted node and the two endpoints of the 2-clique, also correlates positively with prediction accuracy.

Additionally, we find that feature c_2 contributed very little to model performance and, in some cases, may even have a negative impact.

D. Impact of Reducing Structural Features on Prediction Accuracy (Q4)

SSFL extracts 8 structural features within the subgraph for 3-clique prediction. Section 3 explores how different structural features affect prediction accuracy. This section further examines how reducing the number of features impacts AUC and AUPR across six networks, with the results visualized in Figs. 7 and 8. Based on Section 3, features a_1, a_2, b_2, b_4, and c_1 positively contribute to 3-clique prediction, therefore we select these features for further analysis. Specifically, we construct a feature vector $(|a_1|, |a_2|, |b_2|, |b_4|, |c_1|)$ and use it as input to a fully connected neural network for 3-clique prediction. To differentiate between configurations, we denote SSFL_F5 as the model using only these five selected features, while SSFL_F8 utilizes all eight features.

Overall, SSFL_F5 shows comparable prediction performance to SSFL_F8 in the Chebyshev, Trial, and Erdos networks, suggesting that the 5 selected features can effectively capture the formation pattern of potential triangles. Interestingly, SSFL_F5 even outperforms SSFL_F8 in the 145-bit network, likely due to the removal of noise introduced by the additional three features. However, in the SmaGri network, SSFL_F5 slightly underperforms SSFL_F8, while in the DD21 network, the performance gap is more pronounced. This discrepancy may stem from the higher complexity of the DD21 network, where the five selected features alone may be insufficient to fully represent the intricate structural patterns within one-hop neighborhoods.

In summary, different structural features in the one-hop neighborhoods around a target 3-clique affect prediction accuracy differently. Using only a subset of positive features can achieve similar prediction performance across most networks, but prediction accuracy may be limited in larger, more complex networks.

VII. Conclusion

In this paper, we propose a 3-clique prediction method, SSFL, aimed at recommending a 1-clique for a given 2-clique to form a 3-clique. First, we extract the one-hop neighborhoods around the given 2-clique and the 1-clique as enclosing subgraphs. Then, eight different structural features within these subgraphs are selected as inputs for the neural network. A fully connected neural network is employed to learn the potential relationships between different structural features for 3-cliques prediction. Experimental results on 12 networks demonstrate that SSFL outperforms other baselines and exhibits good robustness. Additionally, we explore the contribution of different structural features to prediction accuracy and the impact of using a smaller set of features on prediction results. We present a new perspective on the formation mechanisms of 3-cliques in networks, deepening the understanding of network evolution

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

rules. Predicting 3-cliques reflects the principle of triadic closure, which strengthens network cohesion—especially in social and collaboration networks—facilitates larger community formation, enhances stability, and supports efficient information flow. Unlike traditional link prediction, 3-clique prediction captures higher-order interaction patterns, offering deeper insights into network dynamics. Additionally, network evolution often follows a hierarchical pattern, with 3-cliques serving as fundamental building blocks. Understanding their formation enables more effective modeling of long-term structural changes and better prediction of key connections and potential network shifts, providing valuable insights into network evolution and dynamics.

Since this paper only considers the case of predicted 3-cliques, which is the simplest higher-order structure in the network. Therefore, for future work, we will try to predict other higher-order structures in the network and propose a general method of dataset partitioning. In addition, we will strengthen the theoretical analysis of the study and strengthen the theoretical research on the formation mechanism of higher-order structures in the network.

REFERENCES

[1] Y. Yao et al., "Link prediction based on the mutual information with high-order clustering structure of nodes in complex networks," *Physica A: Stat. Mechan. Appl.*, vol. 610, 2023, Art. no. 128428.

[2] S. Kumar, A. Mallik, and B. Panda, "Link prediction in complex networks using node centrality and light gradient boosting machine," *World Wide Web*, vol. 25, no. 6, pp. 2487–2513, 2022.

[3] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Stat. Mechan. Appl.*, vol. 390, no. 6, pp. 1150–1170, 2011.

[4] Y. Yao et al., "Deep non-negative matrix factorization with edge generator for link prediction in complex networks," *Appl. Intell.*, vol. 54, no. 1, pp. 592–613, 2024.

[5] S. Li, X. Song, H. Lu, L. Zeng, M. Shi, and F. Liu, "Friend recommendation for cross marketing in online brand community based on intelligent attention allocation link prediction algorithm," *Expert Syst. Appl.*, vol. 139, 2020, Art. no. 112839.

[6] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey," *Physica A: Stat. Mechan. Appl.*, vol. 553, 2020, Art. no. 124289.

[7] N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, "Applications of link prediction in social networks: A review," *J. Netw. Comput. Appl.*, vol. 166, 2020, Art. no. 102716.

[8] M. V. Farashah, A. Etebarian, R. Azmi, and R. E. Dastjerdi, "A hybrid recommender system based-on link prediction for movie baskets analysis," *J. Big Data*, vol. 8, pp. 1–24, 2021.

[9] K. Abbas et al., "Application of network link prediction in drug discovery," *BMC Bioinf.*, vol. 22, pp. 1–21, 2021.

[10] K. Han et al., "A review of approaches for predicting drug–drug interactions based on machine learning," *Front. Pharmacol.*, vol. 12, 2022, Art. no. 814858.

[11] E. Nasiri, K. Berahmand, M. Rostami, and M. Dabiri, "A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding," *Comput. Biol. Med.*, vol. 137, 2021, Art. no. 104772.

[12] P. Lu, J. Gao, and W. Liu, "DMNAG: Prediction of disease-metabolite associations based on neighborhood aggregation graph transformer," *Comput. Biol. Chem.*, vol. 115, 2025, Art. no. 108320.

[13] M. E. Newman, "Clustering and preferential attachment in growing networks," *Phys. Rev. E*, vol. 64, no. 2, 2001, Art. no. 025102.

[14] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Netw.*, vol. 25, no. 3, pp. 211–230, 2003.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[16] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[17] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5165–5175.

[18] L. Cai, J. Li, J. Wang, and S. Ji, "Line graph neural networks for link prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5103–5113, Sep. 2021.

[19] R. Lambiotte, M. Rosvall, and I. Scholtes, "From networks to optimal higher-order models of complex systems," *Nature Phys.*, vol. 15, no. 4, pp. 313–320, 2019.

[20] S. Piaggesi, A. Panisson, and G. Petri, "Effective higher-order link prediction and reconstruction from simplicial complex embeddings," in *Proc. Learn. Graphs Conf.*, 2022, pp. 1–55.

[21] D. Shi and G. Chen, "Simplicial networks: A powerful tool for characterizing higher-order interactions," *Nat. Sci. Rev.*, vol. 9, no. 5, 2022, Art. no. nwac038.

[22] D. Easley et al., *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, vol. 1. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[23] A. R. Benson, R. Abebe, M. T. Schaub, A. Jadbabaie, and J. Kleinberg, "Simplicial closure and higher-order link prediction," *Proc. Nat. Acad. Sci.*, vol. 115, no. 48, pp. E11221–E11230, 2018.

[24] H. Nassar, A. R. Benson, and D. F. Gleich, "Pairwise link prediction," in *Proc. 2019 IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, 2019, pp. 386–393.

[25] H. Nassar, A. R. Benson, and D. F. Gleich, "Neighborhood and pagerank methods for pairwise link prediction," *Social Netw. Anal. Mining*, vol. 10, pp. 1–13, 2020.

[26] B. Liu, R. Yang, and L. Lü, "Higher-order link prediction via local information," *Chaos: An Interdiscipl. J. Nonlinear Sci.*, vol. 33, no. 8, 2023.

[27] D. Arrar, N. Kamel, and A. Lakhfif, "A comprehensive survey of link prediction methods," *J. Supercomputing*, vol. 80, no. 3, pp. 3902–3942, 2024.

[28] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," *Internet Math.*, vol. 1, no. 2, pp. 226–251, 2004.

[29] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *Eur. Phys. J. B*, vol. 71, pp. 623–630, 2009.

[30] G. Jeh and J. Widom, "SimRank: A measure of structural-context similarity," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2002, pp. 538–543.

[31] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[32] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.

[33] E. P. Barracchia, G. Pio, A. Bifet, H. M. Gomes, B. Pfahringer, and M. Ceci, "LP-ROBIN: Link prediction in dynamic networks exploiting incremental node embedding," *Inf. Sci.*, vol. 606, pp. 702–721, 2022.

[34] Z. Fang, S. Tan, Y. Wang, and J. Lu, "Elementary subgraph features for link prediction with neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3822–3831, Apr. 2023.

[35] S. Fang, L. Li, S. Bai, Z. Ma, and X. Chen, "Link prediction based on fundamental heuristic elements," *Int. J. Modern Phys. C*, vol. 35, no. 12, pp. 1–21, 2024.

[36] H. A. Mohamed, D. Pilutti, S. James, A. D. Bue, M. Pelillo, and S. Vascon, "Locality-aware subgraphs for inductive link prediction in knowledge graphs," *Pattern Recognit. Lett.*, vol. 167, pp. 90–97, 2023.

[37] Z. Guo et al., "Linkless link prediction via relational distillation," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 12012–12033.

[38] F. Battiston et al., "Networks beyond pairwise interactions: Structure and dynamics," *Phys. Rep.*, vol. 874, pp. 1–92, 2020.

[39] F. Battiston et al., "The physics of higher-order interactions in complex systems," *Nature Phys.*, vol. 17, no. 10, pp. 1093–1098, 2021.

[40] I. Iacopini, M. Karsai, and A. Barrat, "The temporal dynamics of group interactions in higher-order social networks," *Nature Commun.*, vol. 15, no. 1, 2024, Art. no. 7391.

[41] B. A. García, A. Longa, Q. F. Lotito, S. Meloni, and G. Cencetti, "Patterns in temporal networks with higher-order egocentric structures," *Entropy*, vol. 26, no. 3, 2024, Art. no. 256.

[42] A. Patania, G. Petri, and F. Vaccarino, "The shape of collaborations," *EPJ Data Sci.*, vol. 6, pp. 1–16, 2017.

[43] G. Cencetti, F. Battiston, B. Lepri, and M. Karsai, "Temporal properties of higher-order interactions in social networks," *Sci. Rep.*, vol. 11, no. 1, 2021, Art. no. 7028.

[44] A. R. Benson, "Three hypergraph eigenvector centralities," *SIAM J. Math. Data Sci.*, vol. 1, no. 2, pp. 293–312, 2019.

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.

[45] Q. F. Lotito, F. Musciotto, A. Montresor, and F. Battiston, "Higher-order motif analysis in hypergraphs," *Commun. Phys.*, vol. 5, no. 1, 2022, Art. no. 79.
[46] M. Contisciani, F. Battiston, and C. De Bacco, "Inference of hyperedges and overlapping communities in hypergraphs," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 7229.
[47] A. Eriksson, D. Edler, A. Rojas, M. de Domenico, and M. Rosvall, "How choosing random-walk model and network representation matters for flow-based community detection in hypergraphs," *Commun. Phys.*, vol. 4, no. 1, 2021, Art. no. 133.
[48] N. W. Landry, J.-G. Young, and N. Eikmeier, "The simpliciality of higher-order networks," *EPJ Data Sci.*, vol. 13, no. 1, pp. 17–37, 2024.
[49] A. Ceria and H. Wang, "Temporal-topological properties of higher-order evolving networks," *Sci. Rep.*, vol. 13, no. 1, 2023, Art. no. 5885.
[50] Y. Yao et al., "CICN: Higher-order link prediction with clustering mutual information of common neighbors," *J. Comput. Sci.*, vol. 85, 2025, Art. no. 102513.
[51] S. Rafiee, C. Salavati, and A. Abdollahpouri, "CNDP: Link prediction based on common neighbors degree penalization," *Physica A: Stat. Mechan. Appl.*, vol. 539, 2020, Art. no. 122950.
[52] P. J. Etude, "Comparative de la distribution florale dans une portion des alpes et des jura," *Bull. Soc. Vaud. Sci. Nat.*, vol. 37, pp. 547–579, 1901.
[53] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
[54] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.
[55] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, no. 1, 2015, pp. 4292–4293.
[56] Y. Meng, N. Yang, Z. Qian, and G. Zhang, "What makes an online review more helpful: An interpretation framework using XGBoost and SHAP values," *J. Theor. Appl. Electron. Commerce Res.*, vol. 16, no. 3, pp. 466–490, 2020.

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

Zhiheng Mao received the bachelor's degree in 2021, from the Lanzhou University of Technology, Lanzhou, China, where he is currently working toward the M.S. degree with the School of Computer and Communication. His research interests include link prediction and graph neural architecture search.

Yangyang He received the bachelor's degree from the Chaohu University, Hefei, China, in 2020. He is currently working toward the M.S. degree with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China. His research interests include complex network analysis, link prediction, and higher-order link prediction.

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

Ziyu Ti received the B.S. computer science and technology from the University of Xianyang Normal, Xianyang, China, in 2021. She is currently working toward the M.S. degree with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China. Her research interests include link prediction and graph self supervised learning.

Pingxia Guo received the bachelor's degree in software engineering from the Lanzhou Institute of Technology, Lanzhou, China, in 2022. She has also been working toward the master's degree with the School of Computer Science and Communication, Lanzhou University of Technology, Lanzhou, since 2022. Her main research interests include link prediction and multi-scale representation learning.

Fuzhong Nian received the B.S. degree in physics from Northwest Normal University, Lanzhou, China, in 1998, the M.S. degree in control theoretics and engineering from Lanzhou University of Technology, Lanzhou, in 2004, respectively, and the Ph.D degree from the Dalian University of Technology, Dalian, China. He is currently a Professor with the School of Computer and Communication, Lanzhou University of Technology. His research interests include nonlinear dynamics and control, complex networks and systems, and neural networks.

Ning Ma is currently the Doctoral Supervisor of language intelligence and cultural computing in Northwest University, Xi'an, China, for Nationalities, and the Postgraduate Supervisor of computer science and technology and artificial intelligence in computer technology. His research interests include machine learning and natural language processing.

Authorized licensed use limited to: Lanzhou University of Technology. Downloaded on August 24,2025 at 10:34:18 UTC from IEEE Xplore. Restrictions apply.