

Subgraph Encoding with Bicentric Sphere Node Labeling and Pooling for Link Prediction

Zhihong Fang¹, Shaolin Tan^{2*}, Qiu Fang¹, Zhe Li¹, Qing Gao³

¹School of Artificial Intelligence and Robotics, Hunan University, Changsha, China

²Zhongguancun Laboratory, Beijing, China

³School of Automation Science and Electrical Engineering, Beihang University, Beijing, China

fangwhut@hnu.edu.cn, shaolintan@hnu.edu.cn, qfang@hnu.edu.cn, zheli@hnu.edu.cn, gaoqing@buaa.edu.cn

Abstract

Learning representation of the enclosing subgraph of node pairs is recognized as an efficient approach for link-oriented prediction tasks in network applications. The core challenge within this subgraph encoding approach is how to effectively distinguish and then properly aggregate the contribution of nodes in the subgraph into a single vector to indicate the relation between the target node pair. In this work, we propose a novel sphere-based subgraph encoding architecture, namely BS-SubGNN, to address the challenge. In detail, we design two key building blocks, including Bicentric Sphere Node Labeling (BSNL) and Bicentric Sphere Subgraph Pooling (BSSP) to assist message passing in BS-SubGNN. BSNL endows each node a label according to the sphere it belongs to in the subgraph to distinguish the contribution of nodes, while BSSP adopts an attention mechanism to aggregate the contribution of nodes in each sphere. Theoretically, we prove that BS-SubGNN can unify existing node distance labeling methods, and yield discriminative node features with less time complexity. We evaluate the performance of BS-SubGNN in link prediction tasks over a variety of network types, including undirected networks, attribute networks, directed networks, and signed directed networks. Our experimental results demonstrate that BS-SubGNN consistently achieves significant performance improvements over the above diverse types of networks. In particular, compared to those methods with a requisite of multi-hop neighborhood information, BS-SubGNN can obtain better performance even when only one-hop neighborhood information of the node pair is utilized.

Introduction

Graph is a powerful modeling tool for miscellaneous real-world systems, wherein entities are represented as vertices and their interactions are represented as edges (Ou et al. 2016; Zhou et al. 2017; Zhang et al. 2018; Khosla et al. 2020; Zhang et al. 2021b; Kollias et al. 2022; Fiorini et al. 2023). Link prediction, aiming to infer missing links or potential links based on observed links, is a fundamental task in performing analysis of such network data (Leskovec, Huttenlocher, and Kleinberg 2010; Yoo et al. 2023; Ke et al. 2024). Considering the ubiquity of graph-structured data, in

recent years link prediction has attracted large amounts of attention in real-world applications, such as discovering protein interactions (Wu et al. 2024), recommending products in e-commerce system (Xu et al. 2025), social recommendation (Javari et al. 2020; Shu et al. 2021; Xu et al. 2022), drug responses (Liu and Li 2025), and knowledge graph completion (Wang et al. 2025).

Most of the prior efforts on graph neural network (GNN) are devoted to learning node embeddings to preserve label and connectivity patterns around each node, which thus facilitates the downstream link prediction task (Hamilton, Ying, and Leskovec 2017; Zhu et al. 2021). However, as pointed out in (Xu et al. 2018), the expressive power of GNN is upper-bounded by the 1-WL (1-dimensional Weisfeiler-Lehman) test. In addition, GNN is node-centered, and thus leads to representation limitations in capturing edge-oriented patterns.

Recently, subgraph encoding, which learns representation of the enclosing subgraph of node pairs, has been recognized as an efficient approach for link prediction tasks in diverse networks (Zhang and Chen 2017, 2018, 2020; Pan, Shi, and Dokmanić 2021; Cai et al. 2022; Fang et al. 2023; Fang, Tan, and Wang 2023). Subgraph encoding methods surpass the GNN paradigm owing to utilizing structural features which the GNN paradigm is unable to learn. On one hand, a natural idea is to design handcrafted features (Zhang and Chen 2017; Pan, Shi, and Dokmanić 2021; Fang et al. 2023; Fang, Tan, and Wang 2023). However, such features are static and often fail to capture complex topological relationships in graph-structured data. On the other hand, incorporating GNN and distance information (Zhang and Chen 2018, 2020; Cai et al. 2022; Li et al. 2020; Zhang et al. 2021a) is able to capture complex topological relationships in graph-structured data and showcases more expressive power than traditional GNN. For example, subgraph encoding with distance information can distinguish r -regular graph while traditional GNN fails (Li et al. 2020).

Nevertheless, the aforementioned methods still exhibit notable limitations and hence preclude many applications. First, the existing subgraph pattern encoding paradigm suffers from high computational complexity stemming from intricate node labeling or ordering algorithms designed to transcend the 1-WL expressivity barrier. Second, existing link prediction algorithms are inefficient for sizable networks, as

*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

they require traversing higher hop neighborhoods which incur significant overhead. Finally, as the underlying commonalities of miscellaneous real-world networks may be different, a link prediction model is designed case by case and tends to work solely on a particular type of network. Developing a link prediction model that can effectively handle diverse network types remains a significant challenge.

To alleviate the above limitations, theoretically, we propose the definition of bounding distance node labeling as a design guideline of subgraph embedding model. It is proved that typical existing node labeling methods, spanning 0-1 labeling (Zhang et al. 2021a), DRNL (Zhang and Chen 2018) and DE (Li et al. 2020), are special cases of bounding distance node labeling and enjoy the same expressive power. Furthermore, we propose BS-SubGNN, a novel bicentric sphere-based subgraph encoding architecture, to leverage only one-hop neighborhood information for tackling link prediction problem across different network types. In particular, we design two core building blocks, including Bicentric Sphere-based Node Labeling (BSNL) and Bicentric Sphere-based Subgraph Pooling (BSSP) to assist message passing in BS-SubGNN. BSNL is designed to associate each node with a label according to the sphere it belongs to in the subgraph, while BSSP is designed to aggregate the contribution of nodes in each sphere.

In brief, our contributions are summarized as follows. First, we unify the existing distance node labeling methods, and provide theoretical justification of how to balance the efficiency and expressive power of distance node labeling. Second, we propose a link-oriented subgraph encoding architecture, BS-SubGNN, to represent the connectivity pattern in the subgraph of node pairs. BS-SubGNN can effectively distinguish and aggregate the contribution of each node in the subgraph according to the sphere it belongs to. Finally, we conduct extensive experiments on link prediction with diversified real-world datasets, including undirected networks, attribute networks, unsigned directed networks, and signed directed networks. The results demonstrate that the proposed BS-SubGNN outperforms existing link prediction methods in terms of prediction performances, information requirement, flexibility, and time complexity.

Preliminaries

In this section, some notations about the network definition and problem definition are presented.

We consider a network definition $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{v_i\}_{i=1}^{|\mathcal{V}|}$ denotes the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of links, and $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}| \times d}$ as a 3-dimensional tensor containing node and edge features. The diagonal components $\mathbf{A}_{i,i,:}$ denote features of node v_i , and the off-diagonal components $\mathbf{A}_{i,j,:}$ denote features of edge (v_i, v_j) . Note that the adjacency matrix A can be also viewed as one slice of the feature tensor \mathbf{A} . For convenience, we let $A_{i,j} = 1$ iff there exists a link between v_i and v_j regardless its direction and sign. Otherwise, let $A_{i,j} = 0$. Let $Dist(v_i, v_j)$ refers to the shortest distance between v_i and v_j assessed via the adjacency matrix A . For each node $v_i \in \mathcal{V}$, its h -hop neighborhood is defined as $\Gamma^h(v_i) = \{v_j | Dist(v_i, v_j) = h\}$,

where $h = 0, 1, \dots, \infty$. Specifically, 0-hop neighborhood is defined as $\Gamma^0(v_i) = \{v_j | Dist(v_i, v_j) = 0\} = \{v_i\}$. We denote the degree of node v_i as $k_{v_i} = |\Gamma^1(v_i)|$.

We provide the following definitions for bridging gap between graph isomorphism and link prediction. Subgraph encoding for link prediction aims to learn distinguishable link representations based on the extracted subgraphs.

Definition 1 (Enclosing subgraph). *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ and any node pair $u, v \in \mathcal{V}$, its h -hop enclosing subgraph is $\mathcal{G}_{uv}^h = (\mathcal{V}_{uv}^h, \mathcal{E}_{uv}^h, \mathbf{A}_{(uv,h)})$, where $\mathcal{V}_{uv}^h = \{v_i | Dist(u, v_i) \leq h, \text{ or } Dist(v, v_i) \leq h\}$, $\mathcal{E}_{uv}^h = \{(v_i, v_j) \in \mathcal{E} | v_i \in \mathcal{V}_{uv}^h \text{ and } v_j \in \mathcal{V}_{uv}^h\}$, and the corresponding feature matrix $\mathbf{A}_{(uv,h)}$ is induced from \mathbf{A} .*

Definition 2 (Permutation). *A permutation π is a bijective mapping from \mathcal{V} to \mathcal{V} . The permutation group $\Pi_{|\mathcal{V}|}$ consists of all $n!$ possible permutation π . We denote π acting on a node pair (v_i, v_j) as $\pi[(v_i, v_j)] = (\pi[v_i], \pi[v_j])$. We further define the permutation of \mathbf{A} as $\pi[\mathbf{A}]$, where $\pi[A]_{i',j',:} = \mathbf{A}_{i,j,:}$, $\pi[v_i] = v_{i'}$, $\pi[v_j] = v_{j'}$.*

Definition 3 (Node pair h -isomorphism). *Given two node pairs (u, v) , (u', v') , and their h -hop enclosing subgraphs $\mathcal{G}_{uv}^h, \mathcal{G}_{u'v'}^h$, respectively. We say two node pairs are h -isomorphic if $\exists \pi$, such that $\pi[(u, v)] = (u', v')$ and $\pi[\mathbf{A}_{(uv,h)}] = \mathbf{A}_{(u'v',h)}$.*

For convenience, we denote h -isomorphic node pairs by $(u, v, \mathbf{A}_{(uv,h)}) \simeq (u', v', \mathbf{A}_{(u'v',h)})$. Notably, node pair h -isomorphism requires not only graph isomorphism (i.e. $\pi[\mathbf{A}_{(uv,h)}] = \mathbf{A}_{(u'v',h)}$) but also consistent permutation mappings for two ordered node pairs (i.e. $\pi[(u, v)] = (u', v')$). Hence, it is stricter than the set isomorphism defined in (Zhang et al. 2021a) over enclosing subgraphs. Consequently, node pair h -isomorphism is suitable for both directed and undirected link prediction problems, whereas set isomorphism primarily only targets undirected link prediction.

Bounding Distance Node Labeling for Distance Representation

In this section, we provide the definition of bounding distance node labeling and show that it could unify typical existing node labeling methods, including 0-1 labeling, DE and DRNL. Subsequently, we theoretically analyze the computation efficiency and expressive power of distance representation by bounding distance node labeling.

Distance Representation

The superiority of subgraph embedding methods over node embedding methods lies in the encoded distance information of node pair. We first give the definition of distance representation as follows:

Definition 4 (Distance representation). *Given an enclosing subgraph \mathcal{G}_{uv}^h , a function $f(\cdot)$ that maps each node to an embedding space is said to provide a distance representation if, for any two nodes v_i and v_j in \mathcal{V}_{uv}^h , it holds that $f(v_i) \neq f(v_j)$ whenever $Dist(u, v_i) \neq Dist(u, v_j)$ or $Dist(v, v_i) \neq Dist(v, v_j)$.*

Subgraph embedding methods, which capture distances to target nodes through elaborated node labeling schemes, can enhance the inherent inability of GNNs to distinguish node pair h -isomorphism. We provide the following theorem to demonstrate that distance representations can distinguish two enclosing subgraphs in cases where the GNN architecture fails. For convenience, we use 1-WL-GNN to denote a GNN with 1-WL discriminative power (such as GIN (Xu et al. 2018)). The input and output of the l -th GNN layer is denoted as $z^l(\cdot)$ and $z^{l+1}(\cdot)$ respectively for $l = 1, 2, \dots, L$, where the hyperparameter L is the depth of the GNN.

Theorem 1. *Consider two tuples $\mathcal{T}^{(1)} = (u, v, A^{(1)})$ and $\mathcal{T}^{(2)} = (u', v', A^{(2)})$ in the most difficult setting where features $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ are only different in graph structures specified by $A^{(1)}$ and $A^{(2)}$ respectively. Suppose $A^{(1)}$ and $A^{(2)}$ are uniformly independently sampled from all r -regular graphs over \mathcal{V} where $3 \leq r \leq (2 \log |\mathcal{V}|)^{\frac{1}{2}}$. Then, for any small constant $\epsilon > 0$, if the distance representation is utilized as initial input, then there exists 1-WL-GNN with a depth $L \leq \lceil (\frac{1}{2} + \epsilon) \frac{\log |\mathcal{V}|}{\log(r-1)} \rceil$ such that with probability $1 - o(|\mathcal{V}|^{-1})$, the outputs $[z^L(u) || z^L(v)] \neq [z^L(u') || z^L(v')]$, where $||$ is concatenation operation.*

The proof of Theorem 1 follows that of Theorem 3.3 in (Li et al. 2020). It indicates that distance representation could improve the representation ability of GNN architectures. Next, a natural question arises: how can we obtain efficient distance representation for each node within the enclosing subgraphs? To address this question, we first show that the maximum shortest path distance within an enclosing subgraph is bounded and thus mitigates the difficulty of achieving distance representations, as stated in Theorem 2.

Theorem 2. *The enclosing subgraph \mathcal{G}_{uv}^h has the following properties:*

1. *If \mathcal{G}_{uv}^h is connected, there is $1 \leq \text{Dist}(u, v) \leq 2h + 1$. Moreover, for any $v_i \in \mathcal{V}_{uv}^h$, there are $0 \leq \text{Dist}(u, v_i) \leq 3h + 1$ and $0 \leq \text{Dist}(v, v_i) \leq 3h + 1$.*
2. *If \mathcal{G}_{uv}^h is disconnected, then \mathcal{G}_{uv}^h contains two connected components. Nodes in one component satisfy $0 \leq \text{Dist}(v_i, u) \leq h$ and $\text{Dist}(v_i, v) = \infty$, while nodes in the other component satisfy $0 \leq \text{Dist}(v_i, v) \leq h$ and $\text{Dist}(v_i, u) = \infty$.*

Based on the above theorem, we propose a bounding distance node labeling technique for distance representation as follows. It can unify typical existing node labeling methods, including 0-1 labeling, DE, and DRNL.

Definition 5 (Bounding distance node labeling). *Given a node pair (u, v) and enclosing subgraph \mathcal{G}_{uv}^h , the bounding distance node labeling assigns two integer labels for arbitrary node $v_i \in \mathcal{V}_{uv}^h$:*

$$l_u(v_i) = \min\{\text{Dist}(v_i, u), D_T\}, \quad (1)$$

$$l_v(v_i) = \min\{\text{Dist}(v_i, v), D_T\}, \quad (2)$$

where $D_T = 1, 2, \dots, 3h+2$ is a predefined distance bound with $D_T = 1$ in 0-1 labeling and $D_T = 3h + 2$ in DRNL and DE.

Based on bounding distance node labeling, 0-1 labeling and DE generate node feature of v_i by conducting a set aggregation (e.g., sum-pooling and min-pooling) on $l_u(v_i)$ and $l_v(v_i)$ (Li et al. 2020; Zhang et al. 2021a), while DRNL utilizes a hashing function to map $l_u(v_i)$ and $l_v(v_i)$ into a node label, which is subsequently utilized to generate one-hot node feature of v_i (Zhang and Chen 2018). Furthermore, regarding the expressive power of bounding distance node labeling, we have the following theorem.

Theorem 3. *Given enclosing subgraph \mathcal{G}_{uv}^h of node pair (u, v) and bounding distance node labeling with distance bound D_T , distance representations for each node $v_i \in \mathcal{V}_{uv}^h$ can be obtained by at most $(3h + 2 - D_T)$ -layer 1-WL-GNN.*

This theorem inspires us that those node labeling methods are equivalent from 1-WL-GNN with sufficient depth L . Nevertheless, the hyper-parameters, including the subgraph hop h , the distance bound D_T , and the GNN depth L , should be carefully designed in practice to balance expressive power and computation efficiency. In the following, we analyze that the distance bound should not be too small ($D_T = 1$) or large ($D_T = 3h + 2$). A proper choice is to set $D_T = h + 1$.

Failure of Small Distance Bound ($D_T = 1$)

From Theorem 3, sufficient subgraph hop h and GNN depth L can yield powerful distance representation even with tiny D_T . For example, distance representation can be obtained by conducting 0-1 labeling ($D_T = 1$) and $(3h + 1)$ -layer 1-WL-GNN on enclosing subgraphs with large h . However, 0-1 labeling method often achieve subpar performance owing to technical limitations of large h and L .

In detail, regarding L , GNN with large L commonly incurs over-smoothing (Li, Han, and Wu 2018) problems and non-negligible time complexity. Regarding h , we argue that link prediction score can be safely computed within tiny h while higher h introduces additional noise. In fact, in previous works the hyper-parameter h is set to 1 in IGMG (Zhang and Chen 2020), SHFF (Liu et al. 2020), BSAL (Li et al. 2022), NNESSF (Fang et al. 2023) and many others. The hyper-parameter h is set to 2 in SEAL (Zhang and Chen 2018), LGLP (Cai et al. 2022). Theoretically, existing directed link prediction heuristics can be predominantly calculated in the h -hop enclosing subgraph. We prove that two widely used directed heuristics, including rooted PageRank (Jeh and Widom 2003; Brin and Page 2012) and SimRank (Jeh and Widom 2002), can be approximated via enclosing subgraph by following theorem.

Theorem 4. *For a given node pair (u, v) , if the PageRank and SimRank heuristics are approximated from \mathcal{G}_{uv}^h , then the approximation error is bounded by $(1 - c_{PR})^{2h+2}$ and $\frac{(1 - c_{SR})^{2h+2}}{c_{SR}}$ respectively, where $c_{PR}, c_{SR} \in (0, 1)$.*

Theorem 4 proves that PageRank and SimRank can be approximated by enclosing subgraphs. In addition, other simpler heuristics, for example, common neighbors, can be accurately computed by enclosing subgraphs.

Inefficiency of Large Distance Bound

$$(D_T = 3h + 2)$$

From Theorem 3, it can also yield expressive distance representation with tiny 1-WL-GNN layer L if the distance bound is large. For example, DRNL and DE ($D_T = 3h + 2$) can directly output ideal distance representations without 1-WL-GNN. However, we argue that DRNL and DE require expensive time complexity while the performance gain is marginal.

In detail, in case of $D_T = 3h + 2$, the required time complexity to associate node labels increases significantly due to the distance computation operation compared to that of small D_T . Additionally, the performance gain to set $D_T = 3h + 2$ is limited. On one hand, the distance label of most nodes are within h in the enclosing subgraphs, while only a small portion of nodes have a label greater than h . On the other hand, the discussion of hyper-parameter h in Section inspires labels greater than h to be unnecessary. In particular, as many real-world is networks exhibit power-law degree distributions $P_k = -ck^{-\gamma}$ with exponents $2 < \gamma < 3$ (Barabási and Albert 1999; Fronczak, Fronczak, and Hołyst 2004), we explain the rare appearance of large distance bound by the following Theorems.

Theorem 5. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a scale-free network satisfying:

1. **Power-law degree distribution:** The graph obeys a power-law degree distribution $P_k \propto k^{-\gamma}$ with exponent $\gamma \in (2, 3)$.
2. **Degree constraint:** The maximum degree k_{\max} satisfies $k_{\max} = O(|\mathcal{V}|^{\frac{1+\epsilon}{6h+2-2h\gamma}})$ for any constant $\epsilon > 0$.

For a uniformly sampled node pair (u, v) with h -hop enclosing subgraph \mathcal{G}_{uv}^h , DRNL and DE are equivalent to bounding distance node labeling of $D_T \geq h + 1$ with probability exceeding $O(\exp(-|\mathcal{V}|^{-\epsilon}))$.

Theorem 6. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a scale-free network satisfying:

1. **Power-law degree distribution:** The graph obeys a power-law degree distribution $P_k \propto k^{-\gamma}$ with exponent $\gamma \in (2, 3)$.
2. **Degree constraint:** The maximum degree k_{\max} satisfies $k_{\max} = O(|\mathcal{V}|^{\frac{1+\epsilon}{(3-\gamma)(2h+1)}})$ for any constant $\epsilon > 0$.

Consider a uniformly sampled node pair (u, v) and the corresponding h -hop enclosing subgraph \mathcal{G}_{uv}^h . If \mathcal{G}_{uv}^h is connected, then the bounding distance node labeling with $D_T = h + 1$, together with an L -layer 1-WL-GNN ($L > 1$), almost surely yields a distance representation for all nodes in \mathcal{G}_{uv}^h .

Theorem 5 and Theorem 6 describe that superiority of distance information of $D_T > h + 1$ is limited.

Proper Distance Bound ($D_T = h + 1$)

Based on the above discussion, in this paper we set $D_T = h + 1$ as a proper distance bound to balance the expressive power and computation efficiency.

In detail, regarding the computation efficiency, since subgraph extraction is a Breadth-First Search (BFS) process in

which distance within $h + 1$ is implicitly generated, the distance information within $h + 1$ can be directly obtained in the subgraph extraction process. In this case, neighborhoods extraction, subgraph extraction and node labeling process can be incorporated into a single process. Hence, the computational cost of the bounding distance node labeling process is the same for all $D_T \leq h + 1$. In comparison, the distance information ranging from $h + 1$ to $3h + 1$ requires additional shortest path distance computation, which could greatly incur time consuming. Further experimental comparisons and theoretical analysis of the time complexity of different node labeling methods are shown in Section .

Regarding the expressive power, as discussed in Section , bounding distance labeling with $D_T = h + 1$ theoretically enjoys identical discriminative expressive power compared to the ideal case of $D_T = 3h + 2$. Experimentally, distance labeling with $D_T = h + 1$ is easier to converge than distance labeling with $D_T = 3h + 1$.

Next section, we propose a novel subgraph pattern encoding method with $D_T = h + 1$ to encounter the link prediction task.

Overall Framework of BS-SubGNN

In this section, we propose the neural architecture operating on link-oriented subgraphs with bicentric sphere node labeling and pooling. The framework of the proposed BS-SubGNN is shown in Fig. 1.

Bicentric Sphere Definition

Given the enclosing subgraph \mathcal{G}_{uv}^h of the node pair (u, v) , we make the following three types of sphere definition.

Definition 6. CN_q -sphere: Given $q = 1, 2, \dots, h$, the CN_q -sphere of nodes u, v is given by the node set $\mathcal{V}_{uv, CN}^q = \{v_i | \text{Dist}(u, v_i) = q, \text{Dist}(v, v_i) = q\}$.

Definition 7. u_q -sphere: Given $q = 1, 2, \dots, h$, the u_q -sphere of nodes u, v is given by the node set $\mathcal{V}_{uv, u}^q = \{v_i | \text{Dist}(u, v_i) = q, \text{Dist}(v, v_i) > q\}$.

Definition 8. v_q -sphere: Given $q = 1, 2, \dots, h$, the v_q -sphere nodes u, v is given by the node set $\mathcal{V}_{uv, v}^q = \{v_i | \text{Dist}(u, v_i) > q, \text{Dist}(v, v_i) = q\}$.

In the following, the above sphere structure will be utilized to generate node label and design pooling technique for subgraph representation. The sphere can be described by the distance Bound $D_T = h + 1$ as shown in Eqs. (4) and (5).

Bicentric Sphere Node Labeling

The Bicentric Sphere Node Labeling (BSNL) is the first step of BS-SubGNN. It aims to assign a label to each node in the subgraph. In detail, we utilize the breadth-first-search (BFS) method to generate the above subgraph and node label in the meantime. The process contains two main parts. The first part obtains the q -hop neighborhood $\Gamma^q(u)$, $\Gamma^q(v)$ ($q = 0, 1, \dots, h$) of the target node u and v , respectively. The second part of the node labeling process is to generate the enclosing subgraph and the corresponding node label.

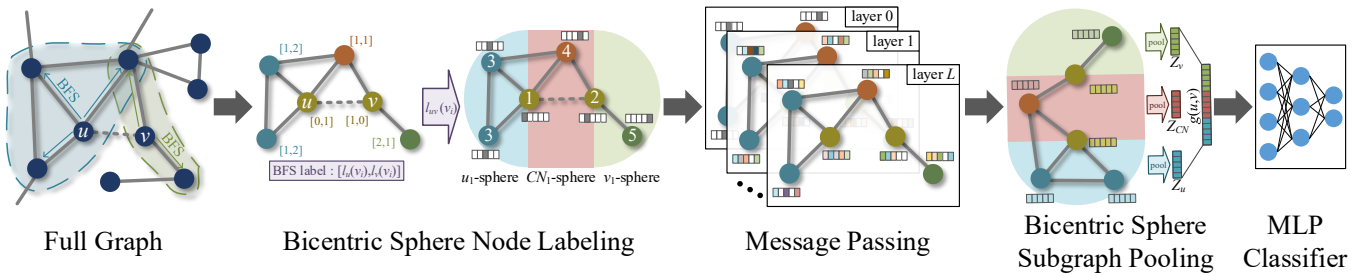


Figure 1: The overall architecture of BS-SubGNN.

The node set \mathcal{V}_{uv}^h of the subgraph \mathcal{G}_{uv}^h can be obtained by merging all the above q -neighborhood sets as follows:

$$\mathcal{V}_{uv}^h = \bigcup_{q=0}^h \Gamma^q(u) \cup \bigcup_{q=0}^h \Gamma^q(v) \quad (3)$$

Then, for node $v_i \in \mathcal{V}_{uv}^h$, we obtain its u -BFS label $l_u(v_i)$ with $D_T = h + 1$ by

$$l_u(v_i) = \begin{cases} q, & \text{if } v_i \in \Gamma^q(u) \\ h + 1, & \text{otherwise.} \end{cases} \quad (4)$$

Similarly, we obtain its v -BFS label $l_v(v_i)$ with $D_T = h + 1$ by

$$l_v(v_i) = \begin{cases} q, & \text{if } v_i \in \Gamma^q(v) \\ h + 1, & \text{otherwise.} \end{cases} \quad (5)$$

Finally, for each node $v_i \in \mathcal{V}_{uv}^h$, its bicentric sphere label $l_{uv}(v_i)$ is determined as follows:

$$\begin{aligned} l_{uv}(v_i) = & \min\{l_u(v_i), 1\} \min\{l_v(v_i), 1\} \\ & [3 \min\{l_u(v_i), l_v(v_i)\} \\ & + \min\{\max\{l_u(v_i) - l_v(v_i), -1\}, 1\} - 1] + \\ & \min\{l_u(v_i), 1\} + 1. \end{aligned} \quad (6)$$

Bicentric Sphere Subgraph Pooling

Bicentric Sphere subgraph pooling is the third step of BS-SubGNN, which takes the node embeddings $\{z(v_i), \forall v_i \in \mathcal{V}_{uv}^h\}$ in the subgraph \mathcal{G}_{uv}^h as input and outputs a fix-sized subgraph embedding vector $g(u, v)$.

In our proposed BS-SubGNN, the subgraph embedding is obtained by

$$g(u, v) = [S_u || S_{CN} || S_v], \quad (7)$$

where S_u is u_1 -sphere embedding, which is computed by

$$S_u = \sum_{v_i \in \mathcal{V}_{uv}^1, u} \alpha_{u,i} z(v_i), \quad (8)$$

where attention coefficient $\alpha_{u,i}$ is computed as follows:

$$\alpha_{u,i} = \exp(\tanh(z(v_i) \vec{a}_u)). \quad (9)$$

Similarly, CN_1 -sphere embedding S_{CN} and v -sphere embedding S_v are computed respectively by

$$S_{CN} = \sum_{v_i \in \mathcal{V}_{uv}^1, CN} \alpha_{CN,i} z(v_i), \quad (10)$$

$$S_v = \sum_{v_i \in \mathcal{V}_{uv}^1, v} \alpha_{v,i} z(v_i), \quad (11)$$

where attention coefficient $\alpha_{CN,i}$ and $\alpha_{v,i}$ is computed as follows:

$$\alpha_{CN,i} = \exp(\tanh(z(v_i) \vec{a}_{CN})), \quad (12)$$

$$\alpha_{v,i} = \exp(\tanh(z(v_i) \vec{a}_v)). \quad (13)$$

Here, $\vec{a}_u \in \mathbb{R}^{Ld_{hid}}$, $\vec{a}_{CN} \in \mathbb{R}^{Ld_{hid}}$ and $\vec{a}_v \in \mathbb{R}^{Ld_{hid}}$ are parameters to be trained.

The Bicentric Sphere Subgraph Pooling only utilizes one-hop neighborhood information of the enclosing subgraph. The reason lies in two aspects. On one hand, in the most difficult setting where multi-hop node labeling is unavailable, BSSP can still yield discriminative link representations. On the other hand, observing that only one-hop is utilized in most of situations (Zhang and Chen 2020; Liu et al. 2020; Li et al. 2022; Fang et al. 2023), BSSP further enhance the labeling within one-hop to attain better performance and quickly converge. In addition, Eqs. (9), (12) and (13) enjoy both advantages of the sum operator in (Xu et al. 2018) and that of the attention mechanism in (Velickovic et al. 2017; Huang et al. 2019, 2021a).

Related Work

In this section, we review some closely related literature on graph representation methods regarding link prediction tasks, including network embedding methods and subgraph encoding methods.

Network Embedding Methods

Network embedding methods project graph nodes as lower-dimensional vectors for downstream tasks such as node classification and link prediction (Yang, Cohen, and Salakhudinov 2016; Wang et al. 2017; Derr, Ma, and Tang 2018; Islam, Prakash, and Ramakrishnan 2018; Chen et al. 2018; Shchur et al. 2018; Huang et al. 2021b; Rozemberczki, Allen, and Sarkar 2021; Yun et al. 2021). Random-walk-based methods such as deepwalk (Perozzi, Al-Rfou, and Skiena 2014), LINE (Tang et al. 2015) and node2vec (Grover and Leskovec 2016), treat nodes as words and random walks as sentences, and hence obtain node embeddings via language models. In addition, graph neural networks (GNNs) learn node embeddings by conducting message passing over each node and its corresponding neighborhoods. For instance, graph convolutional networks (Kipf and Welling 2016a)

| Statistic | C.ele | SMG | EML | YST | KHN | ADV | GRQ | LDG | HPD | ZWL |
|-------------|-------|------|------|------|-------|-------|-------|-------|-------|-------|
| Nodes | 297 | 1024 | 1133 | 2284 | 3772 | 5155 | 5241 | 8324 | 8756 | 6651 |
| Edges | 1024 | 4916 | 5451 | 6646 | 12718 | 39285 | 14484 | 41532 | 32331 | 54182 |
| Avg. degree | 14.46 | 9.60 | 9.62 | 5.82 | 6.74 | 15.24 | 5.53 | 9.98 | 7.38 | 16.29 |

Table 1: Statistical information of each real-world undirected network. Nodes, Edges, Avg. degree refer to the number of nodes, number of edges and average degree, respectively.

| Model | Hop | C.ele | SMG | EML | YST | KHN | ADV | LDG | HPD | GRQ | ZWL |
|-----------|---------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Katz | $h > 1$ | 84.84±2.05 | 86.09±1.06 | 88.45±0.68 | 80.56±0.78 | 84.60±0.79 | 92.13±0.21 | 92.96±0.19 | 85.47±0.35 | 89.81±0.59 | 96.42±0.12 |
| PR | $h > 1$ | 89.14±1.35 | 89.13±0.90 | 89.46±0.63 | 81.40±0.75 | 88.43±0.80 | 92.78±0.18 | 94.46±0.19 | 87.19±0.34 | 89.98±0.57 | 97.20±0.12 |
| SR | $h > 1$ | 75.65±2.24 | 78.39±1.14 | 86.90±0.71 | 73.93±0.95 | 79.55±0.90 | 86.18±0.22 | 90.95±0.14 | 81.73±0.37 | 89.81±0.58 | 95.97±0.16 |
| N2V | $h > 1$ | 80.08±1.52 | 78.30±1.22 | 83.06±1.42 | 77.07±0.36 | 82.21±1.19 | 77.70±0.83 | 91.88±0.56 | 79.61±1.14 | 91.33±0.53 | 94.38±0.51 |
| GAE | $h > 1$ | 83.73±0.75 | 85.88±0.90 | 86.78±1.07 | 77.07±0.36 | 84.37±0.39 | 90.55±0.23 | 93.84±0.21 | 85.21±0.45 | 91.15±0.45 | 95.46±0.30 |
| SEAL | $h = 2$ | 87.44±1.21 | 91.53±0.46 | 92.01±0.38 | 82.07±0.96 | 92.69±0.14 | 95.07±0.13 | 96.44±0.13 | 92.26±0.09 | 97.10±0.12 | 97.46±0.02 |
| LGLP | $h = 1$ | 87.91±0.79 | 91.14±0.59 | 91.22±0.58 | 90.83±0.40 | 92.73±0.43 | 94.95±0.23 | 96.75±0.26 | 92.20±0.26 | 97.51±0.24 | 97.78±0.08 |
| | $h = 2$ | 90.16±0.76 | 92.53±0.29 | 92.03±0.28 | 91.97±0.12 | 93.30±0.09 | 95.40±0.10 | 96.70±0.07 | 92.58±0.08 | 97.68±0.10 | 97.76±0.01 |
| NNESF | $h = 1$ | 85.19±1.24 | 89.96±0.79 | 88.91±0.66 | 89.30±0.50 | 91.02±0.48 | 95.13±0.10 | 94.64±0.15 | 90.09±0.26 | 96.77±0.27 | 95.71±0.10 |
| | $h = 2$ | 86.54±1.14 | 91.11±0.58 | 91.20±0.67 | 89.97±0.44 | 92.79±0.40 | 95.08±0.14 | 97.44±0.06 | 91.67±0.21 | 97.14±0.25 | 97.98±0.10 |
| BS-SubGNN | $h = 1$ | 91.31±0.80 | 92.80±0.55 | 91.36±0.66 | 91.76±0.43 | 94.32±0.49 | 96.27±0.08 | 97.62±0.24 | 92.87±0.76 | 97.63±0.20 | 98.33±0.07 |
| | $h = 2$ | 91.43±0.86 | 93.03±0.54 | 92.79±0.76 | 92.18±0.40 | 94.82±0.41 | <u>95.67±0.11</u> | 97.99±0.12 | 93.70±0.21 | 97.68±0.19 | 98.49±0.11 |

| Model | Hop | C.ele | SMG | EML | YST | KHN | ADV | LDG | HPD | GRQ | ZWL |
|-----------|---------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Katz | $h > 1$ | 85.94±3.46 | 87.68±0.90 | 90.54±0.53 | 85.76±0.64 | 88.27±0.32 | 93.72±0.16 | 94.91±0.27 | 89.52±0.32 | 93.08±0.29 | 97.08±0.09 |
| PR | $h > 1$ | 87.96±1.69 | 91.07±0.59 | 91.01±0.67 | 86.34±0.72 | 92.17±0.24 | 94.03±0.24 | 96.26±0.22 | 91.01±0.23 | 93.18±0.34 | 97.69±0.08 |
| SR | $h > 1$ | 66.43±2.39 | 70.39±1.67 | 87.24±0.84 | 77.56±1.09 | 77.16±0.81 | 83.31±0.35 | 88.71±0.79 | 84.16±0.42 | 92.97±0.31 | 95.44±0.15 |
| N2V | $h > 1$ | 77.98±1.54 | 77.01±1.79 | 83.08±1.36 | 78.48±1.03 | 83.26±0.79 | 79.02±0.65 | 92.12±0.50 | 80.57±0.81 | 93.92±0.31 | 93.82±0.39 |
| GAE | $h > 1$ | 82.53±1.51 | 85.95±0.67 | 88.73±0.92 | 82.65±0.86 | 87.52±1.17 | 90.87±0.26 | 95.24±0.19 | 86.62±0.39 | 93.78±0.33 | 95.79±0.27 |
| SEAL | $h = 2$ | 86.49±1.08 | 91.90±0.31 | 91.93±0.31 | 91.85±0.20 | 93.40±0.13 | 95.18±0.12 | 96.55±0.11 | 93.41±0.09 | 97.86±0.11 | 97.54±0.02 |
| LGLP | $h = 1$ | 87.12±0.11 | 91.41±0.61 | 92.01±0.58 | 92.00±0.31 | 93.68±0.40 | 95.19±0.25 | 97.02±0.11 | 93.27±0.26 | 98.01±0.18 | 97.93±0.09 |
| | $h = 2$ | 89.70±0.53 | 92.92±0.21 | 92.61±0.23 | 92.98±0.10 | 94.14±0.09 | 95.72±0.08 | 96.86±0.06 | 93.65±0.08 | 98.14±0.10 | 97.91±0.01 |
| NNESF | $h = 1$ | 84.27±1.69 | 90.54±0.76 | 90.46±0.55 | 89.97±0.61 | 92.32±0.41 | 95.40±0.12 | 95.60±0.11 | 91.24±0.25 | 97.44±0.19 | 96.56±0.10 |
| | $h = 2$ | 85.37±1.41 | 91.52±0.62 | 91.76±0.57 | 90.60±0.41 | 93.79±0.35 | 95.43±0.14 | 97.54±0.07 | 92.61±0.21 | 97.72±0.18 | 98.01±0.13 |
| BS-SubGNN | $h = 1$ | <u>90.09±1.25</u> | <u>93.35±0.57</u> | <u>92.60±0.58</u> | 92.72±0.41 | <u>95.31±0.37</u> | 96.59±0.09 | <u>97.99±0.15</u> | <u>94.19±0.54</u> | 98.13±0.15 | <u>98.51±0.07</u> |
| | $h = 2$ | 90.49±1.06 | 93.48±0.61 | 93.46±0.64 | 93.05±0.28 | 95.51±0.40 | <u>96.02±0.13</u> | 98.12±0.10 | 94.71±0.18 | 98.15±0.14 | 98.54±0.12 |

Table 2: AUC (upper) and AP (lower) comparison with other undirected link prediction algorithms with 80% training links. The best results and the second-best results for each dataset are in bold and underlined, respectively. Hop refers to the number of hops used to generate features of node pair.

generalize the convolution operation to graph-structured data, while graph attention networks (Velickovic et al. 2017) use attention mechanisms to weigh the importance of different neighbors. Network embedding provides efficient frameworks to project graph data into Euclidean space. However, since network embedding methods are mainly node-centered, the performances in link prediction tasks are limited compared to those edge-centered methods.

Subgraph Encoding Methods

Subgraph encoding methods project the enclosing subgraph of an edge as a lower-dimensional vector. WLN (Zhang and Chen 2017) first proposes the subgraph pattern encoding framework by representing an enclosing subgraph as an adjacency matrix. SEAL (Zhang and Chen 2018) first utilizes GNN architectures to encode enclosing subgraphs. LGLP (Cai et al. 2022) introduces line graph transformation into subgraph pattern encoding to overcome information loss in graph pooling layers. NNESF (Fang et al. 2023) proposes handcraft-based features to overcome limitations of existing

subgraph pattern encoding methods. In addition, subgraph encoding methods also attain superior link prediction performance in other types of networks such as signed networks. For example, SELO (Fang, Tan, and Wang 2023) characterizes enclosing subgraphs via linear optimization models, wherein sign, direction, high order information is encoded.

However, the performance of the aforementioned methods still has some potential for improvement. On one hand, the existing subgraph pattern encoding methods rely on expensive node labeling strategies, including DRNL (Zhang and Chen 2018) and DE (Li et al. 2020). On the other hand, existing subgraph pattern encoding methods are designed only for a specific type of network. For example, WLN (Zhang and Chen 2017), SEAL (Zhang and Chen 2018), LGLP (Cai et al. 2022) and NNESF (Cai et al. 2022) neglect direction and heterogeneous information. SELO is elaborately designed for signed directed networks. Motivated by the above limitations, in this paper, we propose BS-SubGNN with BSNL and BSSP to facilitate efficient and consistent link-oriented subgraph representation.

Experimental Results and Analysis

In this section, we empirically evaluate the performances of our framework BS-SubGNN in undirected networks.

Datasets and Evaluation Metrics Following the experimental settings in (Cai et al. 2022), we consider 10 different datasets, including C.ele, SMG, EML, YST, KHN, ADV, GRQ, LDG, HPD and ZWL (Watts and Strogatz 1998; Newman 2001). Table 1 provides basic statistical information about each undirected network. We randomly select 80% of existing links as positive training samples, and the rest are used as positive test samples. The same number of non-existent links are randomly selected as negative samples. Performance is measured by AUC (area under curve) and AP (average precision), with results averaged over 10 splits.

Baselines The following typical baseline methods for link prediction are adopted as benchmarks to compare the performance of the proposed BS-SubGNN in undirected networks. These methods can be generally classified into: 1) Heuristics methods, such as Katz (Katz 1953), PageRank (PR) (Brin and Page 2012; Wang et al. 2020) and SimRank (SR) (Jeh and Widom 2002), 2) Graph embedding methods, including node2vec (N2V) (Grover and Leskovec 2016) and GAE (Kipf and Welling 2016b), and 3) Subgraph encoding methods, such as (Zhang and Chen 2018), LGLP (Cai et al. 2022), and NNEFS (Fang et al. 2023).

Prediction Performance The hyperparameters for BS-SubGNN in undirected networks are configured as follows. The hidden dimension d_{hid} is 32, the batch size is 256, the learning rate is 0.001, and the message passing layer L is 3 for all networks. The MLP module consists of three hidden layers with 32, 32, and 16 neurons, respectively, followed by a softmax output layer. The number of hops h for subgraph generation is initially set to 1 and then adjusted to 2. For the message passing function $\text{Agg}(\cdot)$, we follow the same graph convolution (Kipf and Welling 2016a) employed in SEAL (Zhang and Chen 2018) and LGLP (Cai et al. 2022) for fair comparison. Epoch numbers are set to 40 for datasets C.ele and SMG, 20 for EML, YST, KHN, and GRQ, and 10 for ADV, LDG, HPD, and ZWL. Table 2 presents the prediction results of AUC and AP.

Several observations of link prediction in undirected networks can be derived from Table 2. First, the subgraph encoding models, including SEAL, LGLP, NNEFS and our BS-SubGNN, outperform graph embedding models and heuristic models, which suggests the effectiveness of utilizing enclosing local topology for learning edge-specific features. Second, when only one-hop information is leveraged, our proposed BS-SubGNN surpasses all the benchmarks on most datasets, except EML and YST. This demonstrates that BS-SubGNN achieves better accuracy even with limited local data. Finally, BS-SubGNN gains the state-of-the-art performance in all datasets when 2-hop information is utilized ($h = 2$), indicating the capability of BS-SubGNN in encoding subgraph patterns with powerful discriminative ability.

| Data | SEAL | LGLP | NNEFS | BS-SubGNN |
|-------|-----------------------------|-----------------------------|-----------------------------|---------------------------------|
| | $O(\mathcal{E} \bar{n}^3)$ | $O(\mathcal{E} \bar{n}^3)$ | $O(\mathcal{E} \bar{k}^h)$ | $O(\mathcal{E} \bar{k}^{h-1})$ |
| C.ele | 100.83±6.35 | 225.47±27.86 | 15.64±0.23 | 8.69±1.30 |
| SMG | 461.48±10.68 | 798.04±95.42 | 36.83±1.15 | 20.27±1.64 |
| EML | 202.44±2.56 | 349.44±23.74 | 41.17±0.95 | 20.62±1.36 |
| YST | 160.92±5.12 | 306.33±3.78 | 50.27±1.15 | 25.07±0.88 |

Table 3: Comparison of time complexity and computation time of subgraph extraction and node attribute assignment procedures for each subgraph encoding model. (Unit is in seconds.)

Time Complexity

In this subsection, we provide insight into comparing the time complexity of our BS-SubGNN with SEAL, LGLP and NNEFS. The number of nodes is $|\mathcal{V}|$, the number of edges is $|\mathcal{E}|$, the average degree is \bar{k} , the average number of nodes in the subgraph is \bar{n} , the enclosing subgraph hop is h . Notice that $\bar{k}^h \leq \bar{n}$. For convenient analysis, we suppose the link prediction task requires to output link scores of all node pairs in \mathcal{E} .

In subgraph encoding models, the total time complexity is mainly determined by node labeling procedure. As discussed in (Liu et al. 2020; Fang et al. 2023), SEAL and LGLP assign node labels via DRNL or DE with time complexity $O(|\mathcal{E}|\bar{n}^3)$. NNEFS needs to extract the intersection of h -hop neighborhood to obtain count features, whose time complexity is $O(|\mathcal{E}|\bar{k}^h)$ (Fang et al. 2023). For our BS-SubGNN, we first extract h -hop neighbors of each node via BFS with time complexity $O(|\mathcal{V}|\bar{k}^h) = O(|\mathcal{E}|\bar{k}^{h-1})$. Then the time complexity of BSNL is neglected as only BFS information is required. Overall, the time complexity of BS-SubGNN is $O(|\mathcal{E}|\bar{k}^{h-1})$.

Table 3 presents time complexity and computation time on subgraph extraction and node attribute assignment procedure of different subgraph encoding models. The data in Table 3 is obtained by running on a computer with eight Intel Xeon CPU E5-2686 v4 CPUs which installs Ubuntu 18.04. It can be observed that the computation time of our BS-SubGNN is significantly less than the computation time of existing subgraph encoding models.

Conclusion

In this paper, we propose BS-SubGNN, an edge-oriented subgraph embedding architecture, for link prediction tasks in a variety type of networks. The BS-SubGNN framework uses a bicentric sphere node labeling block to distinguish the contribution of each node in the subgraph, and uses a bicentric sphere subgraph pooling block to aggregate the contribution of each node. Comprehensive experiments are presented to demonstrate the superiority of BS-SubGNN in terms of prediction performances, information requirement, flexibility, and time complexity.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2022YFF0902800 and National Natural Science Foundation of China under Grant T2322023.

References

- Barabási, A.-L.; and Albert, R. 1999. Emergence of scaling in random networks. *Science*, 286(5439): 509–512.
- Brin, S.; and Page, L. 2012. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18): 3825–3833.
- Cai, L.; Li, J.; Wang, J.; and Ji, S. 2022. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5103–5113.
- Chen, Y.; Qian, T.; Liu, H.; and Sun, K. 2018. “Bridge” enhanced signed directed network embedding. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 773–782.
- Derr, T.; Ma, Y.; and Tang, J. 2018. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining*, 929–934.
- Fang, Z.; Tan, S.; and Wang, Y. 2023. A signed subgraph encoding approach via linear optimization for link sign prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10): 14659–14670.
- Fang, Z.; Tan, S.; Wang, Y.; and Lü, J. 2023. Elementary subgraph features for link prediction with neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(4): 3822–3831.
- Fiorini, S.; Coniglio, S.; Ciavotta, M.; and Messina, E. 2023. SigMaNet: One laplacian to rule them all. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7568–7576.
- Fronczak, A.; Fronczak, P.; and Hołyst, J. A. 2004. Average path length in random networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 70(5): 056110.
- Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30: 1025–1035.
- Huang, J.; Shen, H.; Hou, L.; and Cheng, X. 2019. Signed graph attention networks. In *International Conference on Artificial Neural Networks*, 566–577.
- Huang, J.; Shen, H.; Hou, L.; and Cheng, X. 2021a. SDGNN: Learning node representation for signed directed networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 196–203.
- Huang, Z.; Zhang, S.; Xi, C.; Liu, T.; and Zhou, M. 2021b. Scaling up graph neural networks via graph coarsening. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 675–684.
- Islam, M. R.; Prakash, B. A.; and Ramakrishnan, N. 2018. SIGNet: Scalable embeddings for signed networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 157–169.
- Javari, A.; Derr, T.; Esmailian, P.; Tang, J.; and Chang, K. C.-C. 2020. ROSE: Role-based signed network embedding. In *Proceedings of The Web Conference 2020*, 2782–2788.
- Jeh, G.; and Widom, J. 2002. SimRank: A measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 538–543.
- Jeh, G.; and Widom, J. 2003. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, 271–279.
- Katz, L. 1953. A new status index derived from sociometric analysis. *Psychometrika*, 18(1): 39–43.
- Ke, Z.; Yu, H.; Li, J.; and Zhang, H. 2024. DUPLEX: Dual GAT for complex embedding of directed graphs. 235: 23430–23448.
- Khosla, M.; Leonhardt, J.; Nejdil, W.; and Anand, A. 2020. Node representation learning for directed graphs. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part 1*, 395–411.
- Kipf, T. N.; and Welling, M. 2016a. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Kipf, T. N.; and Welling, M. 2016b. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.
- Kollias, G.; Kalantzis, V.; Idé, T.; Lozano, A.; and Abe, N. 2022. Directed graph auto-encoders. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 7211–7219.
- Leskovec, J.; Huttenlocher, D.; and Kleinberg, J. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, 641–650.
- Li, B.; Zhou, M.; Zhang, S.; Yang, M.; Lian, D.; and Huang, Z. 2022. BSAL: A framework of bi-component structure and attribute learning for link prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2053–2058.
- Li, P.; Wang, Y.; Wang, H.; and Leskovec, J. 2020. Distance Encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems*, 33: 4465–4478.
- Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Liu, X.; and Li, M. 2025. Knowledge-guided domain adaptation model for transferring drug response prediction from

- cell lines to patients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 523–531.
- Liu, Z.; Lai, D.; Li, C.; and Wang, M. 2020. Feature fusion based subgraph classification for link prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 985–994.
- Newman, M. E. 2001. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2): 404–409.
- Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; and Zhu, W. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1105–1114.
- Pan, L.; Shi, C.; and Dokmanić, I. 2021. Neural link prediction with walk pooling. In *9th International Conference on Learning Representations*, 1–18.
- Perozzi, B.; Al-Rfou, R.; and Skiena, S. S. 2014. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710.
- Rozemberczki, B.; Allen, C.; and Sarkar, R. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2): 1–22.
- Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.
- Shu, L.; Du, E.; Chang, Y.; Chen, C.; Zheng, Z.; Xing, X.; and Shen, S. 2021. SGCL: Contrastive representation learning for signed graphs. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, 1671–1680.
- Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. LINE: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, 1067–1077.
- Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y.; et al. 2017. Graph attention networks. In *5th International Conference on Learning Representations*, 1–12.
- Wang, H.; Wei, Z.; Gan, J.; Wang, S.; and Huang, Z. 2020. Personalized PageRank to a target node, revisited. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 657–667.
- Wang, S.; Tang, J.; Aggarwal, C. C.; Chang, Y.; and Liu, H. 2017. Signed network embedding in social media. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, 327–335.
- Wang, Z.; Ma, S.; Wang, K.; and Zhuang, Z. 2025. Rule-guided graph neural networks for explainable knowledge graph reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12784–12791.
- Watts, D. J.; and Strogatz, S. H. 1998. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684): 440–442.
- Wu, L.; Huang, Y.; Tan, C.; Gao, Z.; Hu, B.; Lin, H.; Liu, Z.; and Li, S. Z. 2024. PSC-CPI: Multi-scale protein sequence-structure contrasting for efficient and generalizable compound-protein interaction prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 310–319.
- Xu, C.; He, Y.; Wang, J.; and Zhang, W. 2025. STAIR: Manipulating collaborative and multimodal information for e-commerce recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12899–12907.
- Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.
- Xu, P.; Zhan, Y.; Liu, L.; Yu, B.; Du, B.; Wu, J.; and Hu, W. 2022. Dual-branch density ratio estimation for signed network embedding. In *Proceedings of the ACM Web Conference 2022*, 1651–1662.
- Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, 40–48. PMLR.
- Yoo, H.; Lee, Y.-C.; Shin, K.; and Kim, S.-W. 2023. Disentangling degree-related biases and interest for out-of-distribution generalized directed network embedding. In *Proceedings of the ACM Web Conference 2023*, 231–239.
- Yun, S.; Kim, S.; Lee, J.; Kang, J.; and Kim, H. J. 2021. Neo-GNNs: Neighborhood overlap-aware graph neural networks for link prediction. *Advances in Neural Information Processing Systems*, 34: 13683–13694.
- Zhang, M.; and Chen, Y. 2017. Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 575–583.
- Zhang, M.; and Chen, Y. 2018. Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems*, 5165–5175.
- Zhang, M.; and Chen, Y. 2020. Inductive matrix completion based on graph neural networks. In *8th International Conference on Learning Representations*, 1–14.
- Zhang, M.; Cui, Z.; Neumann, M.; and Chen, Y. 2018. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4438–4445.
- Zhang, M.; Li, P.; Xia, Y.; Wang, K.; and Jin, L. 2021a. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34: 9061–9073.
- Zhang, X.; He, Y.; Brugnone, N.; Perlmutter, M.; and Hirn, M. 2021b. MagNet: A neural network for directed graphs. *Advances in Neural Information Processing Systems*, 34: 27003–27015.
- Zhou, C.; Liu, Y.; Liu, X.; Liu, Z.; and Gao, J. 2017. Scalable graph embedding for asymmetric proximity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2942–2948.
- Zhu, S.; Li, J.; Peng, H.; Wang, S.; and He, L. 2021. Adversarial directed graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4741–4748.