Neurocomputing 665 (2026) 132147

Contents lists available at ScienceDirect

Neurocomputing

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

PHLP: Sole persistent homology for link prediction - interpretable feature extraction

Junwon You[a,1], Eunwoo Heo[a,1], Jae-Hun Jung[a,b,*]

[a] Department of Mathematics, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea
[b] Graduate School of Artificial Intelligence, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

HIGHLIGHTS

- Developed PHLP, an explainable link prediction (LP) method using topology.
- Demonstrated close-to-SOTA LP performance with PHLP.
- PHLP enhanced existing LP models, including SOTA.
- First to achieve near-SOTA LP performance without GNN.

ARTICLE INFO

Communicated by X. Yan

Keywords:
Graph analysis
Link prediction
Persistent homology
Topological data analysis

ABSTRACT

Link prediction (LP), inferring the connectivity between nodes, is a significant research area in graph data, where a link represents essential information about relationships between nodes. Although graph neural network (GNN)-based models have achieved high performance in LP, understanding why they perform well is challenging because most consist of complex neural networks. We employ persistent homology (PH), a topological data analysis method that helps analyze the topological information of graphs, to interpret the features used for prediction. We propose a novel method that employs PH for LP (PHLP) focusing on how the presence or absence of target links influences the overall topology. The PHLP utilizes the *angle hop subgraph* and new node labeling method called *degree double radius node labeling (Degree DRNL)*, which distinguishes the information of graphs better than DRNL. Using only a classifier, PHLP performs similarly to state-of-the-art (SOTA) models on most benchmark datasets. Incorporating the outputs calculated using PHLP into the existing GNN-based SOTA models improves performance across all benchmark datasets. To the best of our knowledge, PHLP is the first method to apply PH to LP without GNNs. The proposed approach, employing PH while not relying on neural networks, enables the identification of crucial factors for improving performance.

1. Introduction

Graph data pervade numerous domains such as social networks, biological systems, recommendation engines, and e-commerce networks [59,69]. The graph is well-suited for modeling complex real-world relationships.

Predicting missing or potential connections within a graph is essential for many applications, unlocking valuable insights and facilitating intelligent decision-making. The ability to predict future network interactions can be applied to diverse domains, including friend recommendations on social networks [2,20,61], knowledge graph completion [28,38], identification of potential drug–protein interactions in bioinformatics [37,47], prediction of protein interactions [31,33,37], and optimization of supply chain logistics [10,11].

The link prediction (LP) problem has been categorized into three major paradigms: heuristic methods, embedding methods, and graph neural network (GNN)-based methods, which are explored in detail in

* Corresponding author at: Department of Mathematics, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea.
Email addresses: jwyou627@postech.ac.kr (J. You), hew0920@postech.ac.kr (E. Heo), jung153@postech.ac.kr (J.-H. Jung).
URL: https://sites.google.com/view/jaehunjung/home?authuser=0 (J.-H. Jung).

[1] The authors have equally contributed to this paper.

https://doi.org/10.1016/j.neucom.2025.132147
Received 21 January 2025; Received in revised form 6 November 2025; Accepted 19 November 2025
Available online 20 November 2025

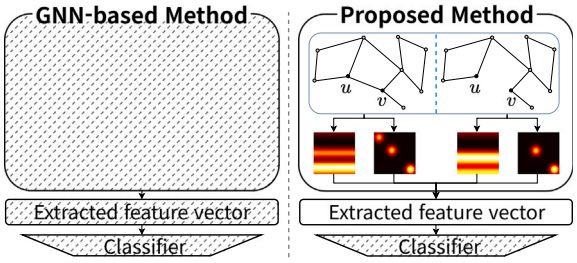

Fig. 1. Difference between the GNN-based and proposed methods. (Left) The GNN-based method extracts feature vectors through optimization (dashed area), making it difficult to interpret what these vectors represent. (Right) The proposed method extracts feature vectors through the designed analysis process, resulting in interpretable vectors.

Section 2. Recently, compared to heuristic [2,4,9,26,35,72] and embedding methods [22,30,43,50], GNN-based models have achieved significant score improvements in capturing intricate relationships within graphs [29,36,42,60,64,67].

However, GNN-based methods are composed of neural networks, making it challenging to understand the reasons for their performance. To explore these reasons, we employ persistent homology (PH), a mathematical tool in topological data analysis (TDA) that enables the inference of topological information regarding the manifold approximating the data [17,24] by quantifying the persistence of topological features across multiple scales. Various research has had successful outcomes in applying PH to graph classification and node classification tasks [13,15,23,25,49,58,62,63,70,71]. In contrast, relatively few studies have explored using PH for LP. The topological loop-counting (TLC) GNN [60] is a notable example that uses PH. The TLC-GNN injects topological information into a GNN, and experiments were conducted on benchmark data where node attributes are available.

In this context, as illustrated in Fig. 1, we present a novel approach to LP, called PHLP, which calculates the topological information of a graph. The main difference between GNN-based methods and our method is described in Section 3.6. To use the topological information of subgraphs for LP, we measure how the topological information changes depending on the existence of the target link, as illustrated in Fig. 2. To extract topological information from various perspectives, we utilize *angle hop subgraphs* for each target node. Additionally, we propose new node labeling method called *degree double radius node labeling (Degree DRNL)*, which incorporates degree information for each node, using DRNL [67].

The contributions are summarized as follows:

- We develop an explainable LP method, PHLP, that employs the topological information for LP through PH without relying on neural networks, as illustrated in Fig. 1.
- We demonstrate that the proposed method, even with a simple classifier such as a multilayer perceptron (MLP), can achieve LP performance close to that of state-of-the-art (SOTA) models. This method surpasses the SOTA performance for the Power dataset.
- We reveal that merely incorporating vectors computed by PHLP into existing LP models, including SOTA models, can improve their performance.
- To the best of our knowledge, the proposed method using PH without a GNN is the first to achieve performance close to that of SOTA models.

2. Related work

2.1. Link prediction

Heuristic methods. Heuristic-based approaches to LP compute the predefined structural features within the observed nodes and edges of the graph. Classic methods, such as common neighbors [2], Adamic-Adar [2], Jaccard coefficient [35], and preferential attachment [4], rely on simple heuristics that capture certain aspects of node relationships. Zhou et al. [72] proposed a local random walk method, whereas Jeh and Widom [26] developed SimRank to quantify similarity based on the structural context. Although heuristic methods provide a preliminary understanding of LP, they are limited by their inability to capture complex relationships within graphs. Furthermore, heuristic methods are effective only when the defined heuristics align with the graph structure; therefore, applying heuristic methods across all graph datasets can be challenging.

Embedding methods. Embedding methods map nodes from the graph into a low-dimensional vector space where geometric relationships mirror the graph structure. Koren et al. [30] demonstrated the power of matrix factorization for collaborative filtering. Perozzi et al. [43] introduced DeepWalk, using random walks to generate node sequences and employing the skip-gram model to produce embeddings. Tang et al. [50] developed large-scale information network embedding (LINE), which preserves local and global structures. Grover and Leskovec [22] further advanced this approach with Node2Vec (N2V), proposing a flexible notion of the neighborhood to capture diverse node relationships.

Embedding methods are advantageous due to their applicability regardless of the data characteristics using optimization. Node representations capture global properties and long-range effects through the learning process. However, these methods often require significantly large dimensions to express basic heuristics, resulting in lower performance compared to heuristic methods [40]. Moreover, in embedding methods, Ribeiro et al. [44] explained that two nodes with similar neighborhood structures may have vastly different embedded vectors, especially when they are far apart in the graph, leading to incorrect predictions.

GNN-based methods. The GNN has become a pivotal approach to LP due to its ability to grasp graph-structured data. By effectively incorporating local and global information through message passing and graph aggregation layers, GNNs enhance LP performance. The model by Zhang *et al.* [67] uses subgraphs as the primary structural units to

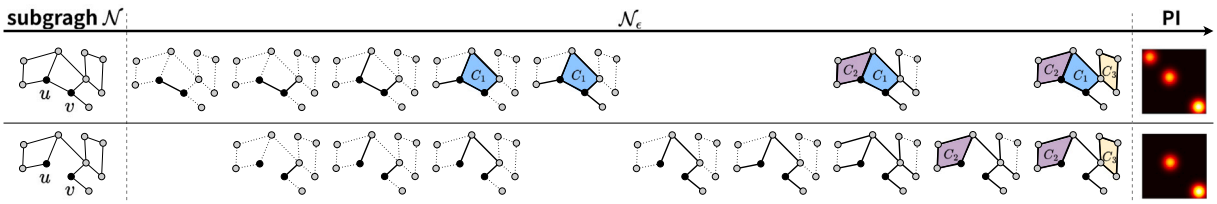

Fig. 2. Topological features in subgraphs with and without a target link (u, v). The diagram illustrates the topological information extraction process for the subgraph $\mathcal{N}$, as described in Section 3.3. The presence (top) or absence (bottom) of the target link changes the topological structure of the graph. Top row: When the target link is connected, three features (C_1, C_2, and C_3) are detected as shown in the persistence image (PI) in the right column. The PI represents the topological features of the subgraph $\mathcal{N}$ (Section 3.3). Bottom row: When the target link is absent, only two features (C_2 and C_3) are detected as depicted in the corresponding PI.

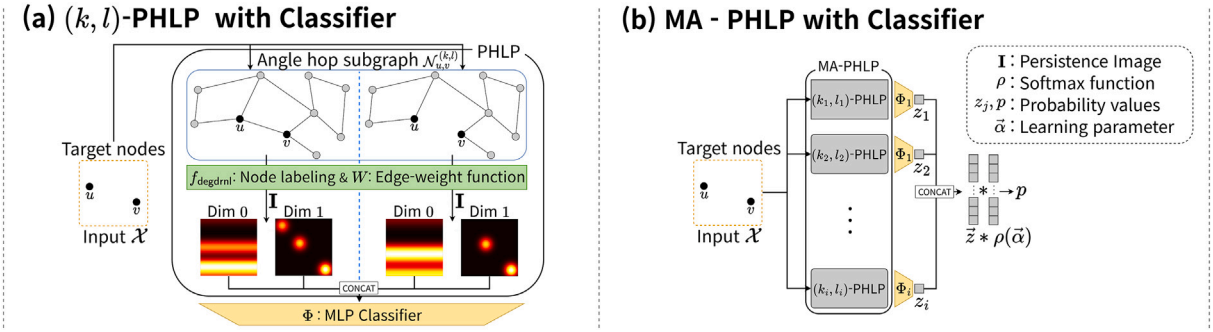

Fig. 3. Overall structure of persistent homology for link prediction (PHLP) and multiangle PHLP (MA-PHLP). (a) PHLP calculates the topological information based on the existence of target links in angle hop subgraphs for each target node. (b) With a classifier, MA-PHLP integrates topological information across various angles to perform LP.

learn and predict connections, resulting in significant improvement. This paradigm shift led to research focusing on refining and advancing subgraph methods in the context of GNNs [12,14,34,36,41,42,56,64,65]. However, despite their superior performance, GNN-based methods pose a challenge in comprehending the underlying mechanisms driving their predictions. Within this context, we develop the PHLP, based on PH, with performance comparable to GNN-based models.

2.2. Persistent homology on graph data

In recent years, PH, a method of analyzing the topological features of data, has been widely used to analyze graph data. It has demonstrated its effectiveness in graph classification tasks by analyzing the topology of graphs [13,23,25,49,58,62,63,70] and has been applied to node classification tasks [15,23,71]. Bhatia et al. [6] successfully proposed applying PH to LP problem. However, its suitability for LP tasks has been limited, and research on applying PH for LP has progressed slowly. Yan et al. [60] proposed an intriguing approach by integrating PH with GNNs. While their model demonstrates the potential of PH for capturing topological features of graph data, it relies on GNN structures. Additionally, the TLC-GNN requires further research on datasets without node attributes.

Although PH has demonstrated success in graph and node classification tasks, its filtration technique, tailored to analyzing the entire graph structure, might not be optimal for LP as the role of each node in LP differs from that in graph or node classification tasks. To address this challenge and advance research in LP, we develop a filtration method tailored explicitly to LP tasks.

3. Method

3.1. Outline of the proposed methods

We propose (a) PHLP and (b) multiangle PHLP (MA-PHLP) as described in Fig. 3. The PHLP method analyzes the topological structure of the graph, focusing on target links. First, PHLP samples a (k,l)-angle hop subgraph for the given target nodes (Section 3.2). Then, PHLP computes persistence images (PIs; Section 3.3) for cases with and without the target link. To calculate PIs, we introduce the node labeling and define the edge-weight function (Section 3.3). Through PHLP, each target node is transformed into a vector comprising PIs. In addition, LP is performed using the calculated vectors with a classifier (Section 3.5). To reflect diverse topological information, we also propose MA-PHLP, which analyzes data from various angles (Section 3.7).

3.2. Extracting angle hop subgraph

Given a graph $G = (V, E)$ and two nodes $u, v \in V$, a k-hop enclosing subgraph for (u, v) is defined as $\mathcal{N}^{k}_{u,v} = (V', E')$ such that

$$V' = \{z \in V \mid d(u,z) \le k \text{ or } d(z,v) \le k\}, \tag{1}$$

$$E' = \{(z,w) \in E \mid z \in V' \text{and } w \in V'\}, \tag{2}$$

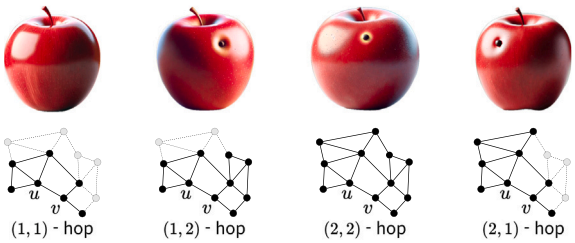

Fig. 4. The motivation of (k,l)-angle hop subgraph. Just as viewing photographs of an apple from multiple angles provides a comprehensive understanding. This figure illustrates the capability to extract subgraphs from various perspectives.

where $d(z,w)$ is the minimum number of edges in any path from z to w in G. We define a (k,l)-angle hop enclosing subgraph, where the term "angle" signifies viewing the subgraph from multiple perspectives. When defining the (k,l)-angle hop subgraph, we were motivated by the observation in Fig. 4 that the information captured in a photograph varies depending on the angle from which it is taken. To confirm features like scratches on an apple, photographs from multiple angles are needed for a comprehensive understanding. Similarly, we hypothesized that a multi-perspective approach is necessary to predict the connections between nodes u and v in a graph. Therefore, we devised a method to extract subgraphs by varying the degrees of separation between u and v, enabling us to capture different views of the graph.

Given a graph $G = (V, E)$ and two nodes $u, v \in V$, a (k,l)-angle hop enclosing subgraph for (u, v) is defined as $\mathcal{N}^{(k,l)}_{u,v} = (V', E')$ such that

$$V' = \{z \in V \mid d(u,z) \le k \text{ or } d(z,v) \le l\}, \tag{3}$$

$$E' = \{(z,w) \in E \mid z \in V' \text{and } w \in V'\}. \tag{4}$$

Thus, the angle hop can generate subgraphs in various forms, providing flexibility to adapt to various graph characteristics. The variation in prediction due to angle is discussed in Section 4.3.

3.3. Topology-aware representation of subgraphs with persistence images

To extract topological information around the target nodes, we convert each local subgraph into PIs, which serve as input to the classifier. This section presents the three-step process involved in generating PIs from a subgraph: (1) assigning node labels and edge weights to subgraphs, (2) computing a persistence diagram from an edge-weighted subgraph, and (3) transforming the persistence diagram into a fixed-size vector representation through a PI method.

Assigning node labels and edge weights to subgraphs. For a given subgraph extracted from a pair of target nodes, we construct the filtration to calculate the topology using PH. Existing filtration methods for graphs do not differentiate the target link, which is crucial for predicting the presence of the target link. To extract topological features from

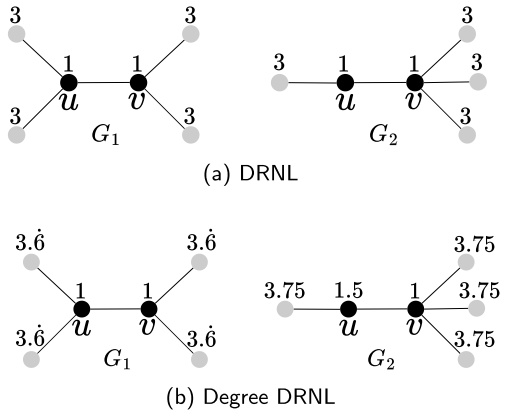

Fig. 5. Node labeling on graphs. (a) Node label values without considering the graph structure cannot distinguish between G_1 and G_2 using DRNL. (b) Applying Degree DRNL allows G_1 and G_2 to be distinguished solely by node label values.

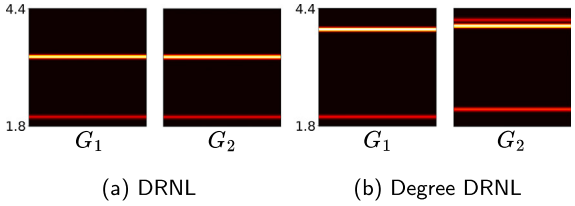

Fig. 6. Persistence images (PIs) for two node labeling methods for the graphs in Fig. 5. (a) DRNL exhibits identical zero-dimensional PIs for G_1 and G_2, (b) Degree DRNL produces distinct outcomes, effectively distinguishing between the two.

graphs for this purpose, the filtration process must distinguish the target link. To address this challenge, we utilize a node labeling that marks nodes based on their relative positions to the target nodes, emphasizing the importance of the target link within the graph's topology. Based on this labeling, we define an edge-weight function that is subsequently used to construct the filtration.

Zhang et al. [67] introduced DRNL, which computes the distance from any node to two fixed nodes. For any subgraph $\mathcal{N} = (V', E')$ of G and two nodes $a, b \in V'$, the DRNL $f^{(a,b)}_{\text{drnl}} : V' \to \mathbb{N}$ based on (a, b) of G for any vertex w in V', is defined as

$$f^{(a,b)}_{\text{drnl}}(w) = 1 + \min(d(w,a), d(w,b)) + q_w(q_w + r_w - 1), \tag{5}$$

where $q_w \in \mathbb{Z}$ and $r_w \in \{0, 1\}$ are integers representing the quotient and remainder, respectively, such that $d(w,a) + d(w,b) = 2q_w + r_w$. We call these two nodes, a and b, *center nodes*. These center nodes do not need to be the target nodes used when extracting the subgraph.

However, DRNL encounters limitations when the graph is transformed into node label information. As depicted in Fig. 5(a) DRNL assigns the same node labels to different graphs, resulting in identical zero-dimensional PIs (Fig. 6a). To incorporate the local topology of each node with the effects of DRNL, we introduced *Degree DRNL*. For a given subgraph $\mathcal{N} = (V', E')$ of G and center nodes $a, b \in V'$, the Degree DRNL $f^{(a,b)}_{\text{degdrnl}} : V' \to \mathbb{R}$ based on (a, b), for all vertices w in V', is defined as

$$f^{(a,b)}_{\text{degdrnl}}(w) = f^{(a,b)}_{\text{drnl}}(w) + \frac{M - \deg(w)}{M}, \tag{6}$$

where M denotes the maximum degree of nodes in $\mathcal{N}$. The $(M - \deg(w))/M$ term above assigns larger values for lower degrees of w. When $M = \deg(w)$ for some vertex w, the value of Degree DRNL matches the original DRNL, ensuring that the edges connected to nodes with higher degrees are assigned smaller values, promoting their earlier emergence in the filtration. Fig. 5(b) demonstrates various node labels obtained using Degree DRNL, resulting in PIs that can be distinguished from each other (Fig. 6b).

For a given subgraph $\mathcal{N} = (V', E')$, let $f : V' \to \mathbb{R}$ be a node labeling function. The edge-weight function $W(f) : E' \to \mathbb{R}$, for any edge (w, z) in E', is defined as $W(f)(w, z) = \max(f(w), f(z))$.

Computing persistence diagram from an edge-weighted subgraph. Given an edge-weighted subgraph $\mathcal{N} = (V', E', W)$, we construct a Rips filtration [19,21,54] and compute its PH. First, we create a sequence of subgraphs $\{\mathcal{N}_\epsilon\}_{\epsilon \in \mathbb{R}}$, where each $\mathcal{N}_\epsilon = (V', E'_\epsilon)$ and $E'_\epsilon = \{e \in E' \mid W(e) \le \epsilon\}$. Second, we convert each subgraph $\mathcal{N}_\epsilon$ into the Rips complex $K_\epsilon = \{\tau \in \mathbb{X} \mid (w, z) \in E'_\epsilon \text{for any two vertices } w, z \in \tau\}$, where $\mathbb{X}$ is the power set of V'. In K_ϵ, a simplex τ is formed when the vertices in τ are pairwise connected by edges in $\mathcal{N}_\epsilon$. Then, the Rips filtration is obtained as $K_{\epsilon_1} \hookrightarrow K_{\epsilon_2} \hookrightarrow \cdots \hookrightarrow K_{\epsilon_m} = \mathbb{X}$ for $\epsilon_1 \le \epsilon_2 \le \cdots \le \epsilon_m$. Third, we compute the p-dimensional homology group $H_p(K_\epsilon)$ for each complex K_ϵ and track how these groups change as ϵ increases. The persistence diagram D [19] comprises persistence pairs (b, d) representing the ϵ values at which a homological feature appears b and disappears d, respectively, in the filtration.

Transforming the persistence diagram into a fixed-size vector representation. We convert the persistence diagram into a PI [3]. For a given persistence diagram D, consider a linear transform $L : \mathbb{R}^2 \to \mathbb{R}^2$ defined by $L(x, y) = (x, y - x)$. The image set of D under this transformation is denoted as $L(D)$. For each point (b, d') in $L(D)$, a weight function $\phi_{(b,d')} : \mathbb{R}^2 \to \mathbb{R}$ is defined that assigns a weight to each point in the persistence diagram. A common choice for $\phi_{(b,d')}$ is the Gaussian function centered at (b, d'). The nonnegative function is defined as $h : \mathbb{R}^2 \to \mathbb{R}$, as $h(x, y) = 1/\log(1 + |y|)$. The function h is zero along the horizontal x-axis, and is continuous and piecewise differentiable, satisfying the conditions presented in [3]. The persistence surface $\rho_D : \mathbb{R}^2 \to \mathbb{R}$ is defined as

$$\rho_D(z) = \sum_{(b,d') \in L(D)} h(b, d')\phi_{(b,d')}(z). \tag{7}$$

The continuous surface ρ_D is discretized into a finite-dimensional representation over a predefined grid. This grid consists of n cells, each corresponding to a specific region in the plane. The PI is defined as an array of values $I(\rho_D)_p$ for each cell p. Each $I(\rho_D)_p$ in this array is computed by integrating the persistence surface ρ_D over the area of cell p:

$$I(\rho_D)_p = \iint_p \rho_D \, dy \, dx. \tag{8}$$

The resulting vector $I(\rho_D)$, generated from the diagram D, is used as the input feature for link prediction.

3.4. Theoretical analysis of the proposed filtration method

We construct filtrations of graphs to compute persistence diagrams and extract topological features. Existing filtrations do not explicitly consider the target link in a graph. We propose a novel filtration method as explained in Section 3.3 based on node labeling. The proposed construction demonstrates stability for node labeling functions, providing a strong theoretical foundation for validity. Empirical validation of this theoretical stability under noisy node labeling is provided in Appendix D.

Definition 1 (Graph isomorphism). A *graph isomorphism* between two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is a bijection between the vertex sets $f : V_1 \to V_2$ such that any two vertices u and v of V_1 are adjacent in G_1 if and only if $f(u)$ and $f(v)$ are adjacent in G_2.

When representing graphs as vectors, the goal is to ensure that the same vector represents isomorphic graphs. However, in the context of the LP problem, two isomorphic graphs G_1 and G_2 can be represented by different vectors, depending on the placement of the target node. To

address this, we redefine graph isomorphism to account for target nodes explicitly. Originally, the target node is needed for calculating the node label, but we denote the target nodes (u, v) in graph G as $G^{(u,v)}$.

Definition 2 (Graph isomorphism with target nodes). A *graph isomorphism with target nodes* between two graphs $G_1^{(u_1,v_1)}$ and $G_2^{(u_2,v_2)}$ is a graph isomorphism f between G_1 and G_2 such that $f(\{u_1, u_2\}) = \{v_1, v_2\}$.

To guarantee the robustness of our approach under minor perturbations, we present a stability theorem that quantifies the impact of changes in node labeling on the persistence diagrams.

Theorem 1 (Stability theorem). *Let $G^{(u,v)} = (V, E)$ be a graph with target nodes u, v in V and let $f^{(u,v)}$ and $g^{(u,v)}$ be two node labeling functions defined on $G^{(u,v)}$. Denote the p-dimensional persistence diagrams of $G^{(u,v)}$ obtained from the filtrations constructed by $f^{(u,v)}$ and $g^{(u,v)}$ as $dgm_p(f^{(u,v)})$ and $dgm_p(g^{(u,v)})$, respectively. Then, the following stability inequality holds:*

$$D_B(dgm_p(f^{(u,v)}), dgm_p(g^{(u,v)})) \leq \|f^{(u,v)} - g^{(u,v)}\|_\infty,$$

where D_B is bottleneck distance, $\|\cdot\|_\infty$ is infinity norm.

The proof of the theorem is in Appendix A. This theorem aligns with the general framework of stability results for persistence diagrams [16], as established in the context of sub-level set filtrations. Our result extends this framework to node labeling-based filtrations we designed in the context of graphs, addressing the challenges of incorporating target nodes in LP tasks. It guarantees that small changes in the node labeling function lead to bounded changes in the resulting persistence diagrams, measured by the bottleneck distance. This result validates the robustness of persistence diagrams derived from node labeling functions, ensuring their reliability for LP tasks. In particular, this theorem applies when the node labeling functions are defined as DRNL $f_{drnl}^{(u,v)}$ and Degree DRNL $f_{degdrnl}^{(u,v)}$, as shown in the following corollary.

Corollary 1. *Let $G^{(u,v)} = (V, E)$ be a graph with target nodes u, v in V. Then, the bottleneck distance between two persistence diagrams $dgm_p(f_{drnl}^{(u,v)})$ and $dgm_p(f_{degdrnl}^{(u,v)})$ is bounded by 1.*

Proof of Corollary 1. By Theorem 1, we have $D_B(dgm_p(f_{drnl}^{(u,v)}), dgm_p(f_{degdrnl}^{(u,v)}) \leq \|f_{drnl}^{(u,v)} - f_{degdrnl}^{(u,v)}\|_\infty = \max_w(M - \deg(w))/M \leq 1$. This completes the proof.

The boundedness established in Corollary 1 demonstrates that DRNL and Degree DRNL yield similar persistence diagrams. This finding confirms that degree information is included while maintaining the information extracted using the DRNL.

Corollary 2. *Suppose there exists a graph isomorphism with target nodes between two graphs $G_1^{(u_1,v_1)}$ and $G_2^{(u_2,v_2)}$. Then, the persistence diagrams of $G_1^{(u_1,v_1)}$ and $G_2^{(u_2,v_2)}$, obtained from filtrations constructed using the same fixed node labeling function, are identical.*

The result guarantees consistency in the representation of isomorphic graphs that preserve target nodes, ensuring they produce identical persistence diagrams under the same node labeling function.

3.5. Predicting the existence of the target link

For the given target nodes (u, v), we sample the (k, l)-angle hop subgraph $\mathcal{N}_{u,v}^{(k,l)}$, denoted as $\mathcal{N}^-$ (Section 3.2), assuming that the target link does not exist during this process. On this subgraph, we extract topological features by calculating PH and its vectorization (i.e., the PI, as described in Section 3.3). The vectorization is calculated for each dimension and concatenated. If $k \neq l$, for symmetry, we repeat the same process with the (l, k)-angle hop subgraph once and consider the average of the two vectors, denoting this vector as x^-. To observe the difference in topological features, we consider a subgraph $\mathcal{N}^+$ obtained by connecting the target link to $\mathcal{N}^-$. For this graph, x^+ denotes the vector obtained using this method.

To predict the existence of the target link with the vectors x^- and x^+, we employ an MLP classifier $\Phi : \mathbb{R}^{2(d+1)n^2} \to \mathbb{R}$ where n represents the resolution of the PI, and d denotes the maximal dimension of PH. The model predicts the existence of a link between two target nodes with the following probability:

$$z_{uv} = \sigma(\Phi(x)), \tag{9}$$

where x is the concatenation of x^- and x^+, and σ is the activation function. For the training dataset $\mathcal{X} \subseteq V \times V$, comprising positive and negative links corresponding to the elements of E and $(V \times V) \setminus E$, respectively, we define the loss function as follows:

$$\sum_{(u,v)\in\mathcal{X}} BCE(z_{uv}, y_{uv}), \tag{10}$$

where $BCE(\cdot, \cdot)$ represents the binary cross-entropy loss and y_{uv} denotes the label of the target link (u, v), which is 0 for negative links or 1 for positive links.

3.6. The interpretability of the feature extraction

Our method enhances the interpretability of feature extraction compared to GNN-based methods. While GNNs can extract feature vectors and visualize them using dimensional reduction techniques, understanding the specific reasons behind the values of these features remains challenging. This difficulty stems from the complex and uncertain nature of the training process, which involves various factors such as optimization methods, learning rates, batch sizes, loss functions, and data distributions. In contrast, our method employs a predefined feature extraction process that operates independently of any training phase, as illustrated in Fig. 1. This approach not only eliminates the ambiguities associated with training but also allows us to precisely understand what each component of the extracted feature vector represents and identify the critical aspects of our method that influence these values. This level of clarity is particularly valuable for applications requiring transparent and accountable decision-making processes.

For example, the development of the Degree DRNL node labeling was enabled by the inherent interpretability of our method. Initially, we utilized the DRNL to construct filtrations but noticed that the Power dataset exhibited the lowest accuracy among our benchmark datasets. This observation led us to conduct an extensive analysis of the subgraphs and their corresponding feature vectors within the Power dataset. As depicted in Fig. 5(a) we identified instances where structurally distinct subgraphs produced identical feature vectors, highlighting a limitation when utilizing DRNL. Despite one subgraph being labeled as 1 and the other being labeled as 0 according to the presence or absence of a target link, our method resulted in identical feature vectors for both subgraphs. We easily identified the reasons behind these identical vectors, which were straightforward since feature extraction does not involve a training process. To address this, we incorporated degree information into the node labeling, as detailed in Eq. (6). We refined the labeling by adding a fractional value between 0 and 1 based on the degree. This enhancement complements the DRNL by assigning an order according to degree, thereby prioritizing simplexes during their simultaneous addition in the filtration process. Furthermore, we established a stability theorem that guarantees minor perturbations in node labelings result in bounded variations in persistence diagrams, thus enhancing the robustness and reliability of our method.

Our methodology not only facilitates a deep understanding of the feature extraction process but also allows for targeted modifications based on specific needs. This adaptability ensures that our approach can be finely tuned to enhance performance or address specific characteristics of the data set.

3.7. Multiangle PHLP

The MA-PHLP maximizes the advantages of PHLP by examining data from various angles through the extraction of subgraphs based on a

Table 1
Link prediction performance measured by the AUC on benchmark datasets (90 % observed links). The top three scores for each dataset are highlighted as follows: First (red), Second (blue), and Third (violet).

Dataset	USAir	NS	PB	Yeast	C. ele	Power	Router	E. coli
AA	95.06 ± 1.03	94.45 ± 0.93	92.36 ± 0.34	89.43 ± 0.62	86.95 ± 1.40	58.79 ± 0.88	56.43 ± 0.51	95.36 ± 0.34
Katz	92.88 ± 1.42	94.85 ± 1.10	92.92 ± 0.35	92.24 ± 0.61	86.34 ± 1.89	65.39 ± 1.59	38.62 ± 1.35	93.50 ± 0.44
PR	94.67 ± 1.08	94.89 ± 1.08	93.54 ± 0.41	92.76 ± 0.55	90.32 ± 1.49	66.00 ± 1.59	38.76 ± 1.39	95.57 ± 0.44
WLK	96.63 ± 0.73	98.57 ± 0.51	93.83 ± 0.59	95.86 ± 0.54	89.72 ± 1.67	82.41 ± 3.43	87.42 ± 2.08	96.94 ± 0.29
WLNM	95.95 ± 1.10	98.61 ± 0.49	93.49 ± 0.47	95.62 ± 0.52	86.18 ± 1.72	84.76 ± 0.98	94.41 ± 0.88	97.21 ± 0.27
N2V	91.44 ± 1.78	91.52 ± 1.28	85.79 ± 0.78	93.67 ± 0.46	84.11 ± 1.27	76.22 ± 0.92	65.46 ± 0.86	90.82 ± 1.49
SPC	74.22 ± 3.11	89.94 ± 2.39	83.96 ± 0.86	93.25 ± 0.40	51.90 ± 2.57	91.78 ± 0.61	68.79 ± 2.42	94.92 ± 0.32
MF	94.08 ± 0.80	74.55 ± 4.34	94.30 ± 0.53	90.28 ± 0.69	85.90 ± 1.74	50.63 ± 1.10	78.03 ± 1.63	93.76 ± 0.56
LINE	81.47 ± 10.71	80.63 ± 1.90	76.95 ± 2.76	87.45 ± 3.33	69.21 ± 3.14	55.63 ± 1.47	67.15 ± 2.10	82.38 ± 2.19
SEAL	97.10 ± 0.87	98.25 ± 0.61	95.07 ± 0.39	97.60 ± 0.33	89.54 ± 1.23	86.21 ± 2.89	95.07 ± 1.63	97.57 ± 0.30
WP	98.20 ± 0.57	99.12 ± 0.45	95.42 ± 0.25	98.21 ± 0.17	93.30 ± 0.91	92.11 ± 0.76	97.15 ± 0.29	98.54 ± 0.19
LGLP	97.09 ± 0.13	99.12 ± 0.00	94.70 ± 0.04	97.53 ± 0.13	88.64 ± 0.29	85.63 ± 0.07	95.51 ± 0.07	98.39 ± 0.08
MPLP	97.01 ± 0.54	96.17 ± 0.84	94.06 ± 0.58	94.25 ± 0.43	90.48 ± 0.87	73.71 ± 1.08	91.90 ± 0.50	96.67 ± 0.14
MA-PHLP	97.10 ± 0.69	98.88 ± 0.45	95.10 ± 0.26	97.98 ± 0.22	90.33 ± 1.16	93.05 ± 0.45	96.30 ± 0.43	97.64 ± 0.20
MA-PHLP (dim 0)	97.10 ± 0.73	98.78 ± 0.65	95.06 ± 0.28	97.98 ± 0.23	89.88 ± 1.22	93.37 ± 0.41	96.37 ± 0.43	97.72 ± 0.17

hyperparameter, the maximum hop (max hop, denoted as $k_{\max}$). The types of angles are elements of all combinations of k and l within the set $\{(k,l) \in \mathbb{Z}^2 | 0 \le l \le k \le k_{\max}, k > 0\}$. If we define the prediction probability of a PHLP for each type of angle hop as z_i for $i = 1, 2, \dots, N$, then MA-PHLP predicts the likelihood of the link existence with the following probability:

$$p = \sum_{i=1}^{N} \alpha_i z_i, \tag{11}$$

where $\alpha = (\alpha_1, \dots, \alpha_N) \in \mathbb{R}^N$ is a trainable parameter. We apply the softmax function to the parameter α to ensure that the sum of all elements equals 1. Moreover, MA-PHLP is trained using the binary cross-entropy loss.

3.8. Hybrid method

The proposed approach easily integrates with existing subgraph methods. Subgraph methods treat the LP task as a binary classification problem comprising two components: a feature extractor F and classifier P. Vectors with PH information calculated using the proposed methods are incorporated through concatenation before the classifier. The detailed process of the hybrid method is outlined as follows:

1. **Subgraph extraction:** For the given graph G and target nodes (u, v), k-hop subgraph $\mathcal{N}^k_{u,v}$ is extracted.
2. **Feature extraction:** Existing methods extract features $Z = F(\mathcal{N}^k_{u,v})$ from the subgraph.
3. **Persistent image calculation:** The methods described in Section 3.3 are applied to $\mathcal{N}^k_{u,v}$, where I denotes the PI vector. An MLP $\Phi : \mathbb{R}^m \to \mathbb{R}^n$ transforms the PI into a format similar to Z. For the hybrid method of MA-PHLP, $\mathcal{N}^k_{u,v}$ is replaced with multiangle subgraphs, concatenating their PI vectors.
4. **Classification:** Next, α_1 Z and $\alpha_2 \Phi(I)$ are concatenated, where α_1 and α_2 are trainable parameters. The softmax function is applied to the parameter $\alpha = (\alpha_1, \alpha_2)$, ensuring that the sum of elements equals 1, denoted by J. This concatenated vector is classified using the existing method's classifier, $P(J)$.

4. Experiments

This section evaluates the performance of MA-PHLP. The experiments were also conducted using only zero-dimensional homology (MA-PHLP (dim 0)). We used the area under the curve (AUC) [7] as an evaluation metric. We repeated all experiments 10 times and reported the mean and standard deviation of the AUC values. Computation time and memory usage results are provided in Appendix E. The code for our implementation is available at https://github.com/AI-hew-math/MA-PHLP.

Table 2
Statistics of the datasets.

Dataset	#Nodes	#Edges	Avg. node deg.	Density
USAir	332	2126	12.81	3.86e-2
NS	1589	2742	3.45	2.17e-3
PB	1222	16,714	27.36	2.24e-2
Yeast	2375	11,693	9.85	4.15e-3
C.ele	297	2148	14.46	4.87e-2
Power	4941	6594	2.67	5.40e-4
Router	5022	6258	2.49	4.96e-4
E.coli	1805	15,660	16.24	9.61e-3

4.1. Experimental settings

Baselines. To evaluate the effectiveness of PHLP, we compared the proposed model with five heuristic methods, four embedding-based methods, and two GNN-based models. The heuristic methods include the Adamic-Adar (AA) [2], Katz index (Katz) [27], PageRank (PR) [8], Weisfeiler-Lehman graph kernel (WLK) [45], and Weisfeiler-Lehman neural machine (WLNM) [66]. For the embedding-based methods, we applied N2V [22], spectral clustering (SPC) [51], matrix factorization (MF) [30], and LINE [50]. Moreover, SEAL [67], WP [42], LGLP [12] and MPLP [18] represent the GNN-based methods.

Datasets. In line with previous studies [67] and [42], we evaluate the performance of our MA-PHLP on the eight datasets in Table 2 without node attributes: USAir [5], NS [39], PB [1], Yeast [55], C. elegans (C. ele) [57], Power [57], Router [46], and E. coli [68]. The detailed statistics for each dataset are summarized in Table 2.

Implementation details. All edges in the datasets were split into training, validation, and testing datasets with proportions of 0.85, 0.05, and 0.1, respectively, ensuring a fair comparison with previous studies. The max hop $k_{\max}$ was set to 3 for most datasets (Table 1). We empirically observed that increasing the number of hops beyond 3 did not lead to significant improvements in performance. Therefore, we adopted a max hop setting of 3 as a general rule. However, for the E. coli dataset, it was reduced to 2 when employing one-dimensional homology due to memory constraints. Conversely, for the Power dataset, the max hop was set to 7. This decision was motivated by the extremely sparse connectivity of the Power graph, which results in very small local subgraphs even at 3 hops. We empirically observed that increasing

Table 3
AUC scores for SEAL with and without TDA features.

Dataset	SEAL	MA-PHLP + SEAL
USAir	97.10 ± 0.87	**97.41 ± 0.62**
NS	98.25 ± 0.61	**98.97 ± 0.30**
PB	95.07 ± 0.39	**95.14 ± 0.39**
Yeast	97.60 ± 0.33	**97.93 ± 0.18**
C.ele	89.54 ± 1.23	**89.61 ± 1.12**
Power	86.21 ± 2.89	**95.53 ± 0.33**
Router	95.07 ± 1.63	**96.15 ± 1.26**
E.coli	97.57 ± 0.30	**97.93 ± 0.34**

the hop size enabled the model to capture more informative topological patterns, leading to meaningful performance gains, as shown in Table 9. Moreover, despite the larger hop size, the resulting subgraph sizes remained comparable to those of other datasets at 3 hops. Therefore, this setting did not incur significant computational or memory overhead. Details regarding computational time and memory usage are provided in Appendix E.

The MLP configurations are fixed across all datasets, since the performance was insensitive to the choice of hyperparameters, such as the number of hidden layers or units. The sigmoid function was employed for the activation function of the PHLP classifier. Tables 3 and 4 present the results of the hybrid methods using SEAL [67] and WP [42], respectively. We choose them because SEAL is simple and powerful GNN method and WP shows the highest scores on 7 datasets. For these experiments, a two-layer MLP was used for the MLP Φ in Step 3 of Section 3.8. We set the k-hops following the original methods, SEAL and WP, and the max hops $k_{\max}$ of MA-PHLP were set as the k, except for the Power dataset. For the Power dataset, we set the k-hop to 1-hop and max hop $k_{\max}$ to 7, respectively, which is discussed in detail in Section 4.4.

4.2. Results

Results of MA-PHLP. Table 1 presents the AUC scores for each model on the benchmark datasets. The top three scores for each dataset are highlighted as follows: First (red), Second (blue), and Third (violet). The results of AA, Katz, WLK, WLNM, N2V, SPC, MF, and LINE are copied from SEAL [67] for comparison. Basically, we followed the experimental setup used in SEAL, including the data split ratio and AUC evaluation. The process of splitting positive and negative samples involves randomness. To ensure statistical reliability and fair comparison, we fixed the random seed from 1 to 10, which consistently determined both the dataset splits and the negative sampling process. The MA-PHLP demonstrates high performance across most datasets, achieving competitive scores. The proposed model outperforms several baselines, falling between the GNN-based models in terms of the AUC score. Notably, for the Power dataset, MA-PHLP achieves the highest AUC score, indicating its effectiveness in capturing link patterns. Experiments with 50 % observed links can be found in Appendix D.

We also investigated whether incorporating higher dimensional topological features would further enhance the performance. Specifically, we compared the MA-PHLP (using both zero- and one-dimensional PH) with the MA-PHLP (dim 0) that uses only zero-dimensional features. As reported in Table 1, the inclusion of one-dimensional PH led to only marginal improvement (around 0.015 % on average), while substantially increasing computational cost. This suggests that zero-dimensional features sufficiently capture the topological information for link prediction in our framework. Given this trade-off, we employed the MA-PHLP (dim 0) for all subsequent ablation and hybrid experiments.

Results of hybrid methods. Simply concatenating the PI vector calculated using PHLP with the final output of the SEAL model increases AUC scores for all datasets, as listed in Table 3. This outcome suggests that when the SEAL model lacks topological information for inference, the vectors calculated using PHLP can serve as additional inputs.

Table 4
AUC scores for WALKPOOL (WP) with and without TDA features.

Dataset	WP	MA-PHLP + WP
USAir	98.20 ± 0.57	**98.27 ± 0.53**
NS	99.12 ± 0.45	**99.24 ± 0.32**
PB	95.42 ± 0.25	**95.58 ± 0.32**
Yeast	98.21 ± 0.17	**98.25 ± 0.18**
C.ele	93.30 ± 0.91	**93.32 ± 0.71**
Power	92.11 ± 0.76	**96.09 ± 0.38**
Router	97.15 ± 0.29	**97.18 ± 0.24**
E.coli	98.54 ± 0.19	**98.57 ± 0.20**

Table 5
AUC scores for MA-PHLP (dim 0) by node labeling.

Dataset	DRNL	Degree DRNL
USAir	96.73 ± 0.64	**97.10 ± 0.73**
NS	98.35 ± 0.58	**98.78 ± 0.65**
PB	94.49 ± 0.27	**95.06 ± 0.28**
Yeast	97.42 ± 0.27	**97.98 ± 0.23**
C.ele	88.97 ± 1.37	**89.88 ± 1.22**
Power	88.51 ± 0.81	**92.77 ± 0.47**
Router	96.21 ± 0.53	**96.37 ± 0.43**
E.coli	97.15 ± 0.18	**97.72 ± 0.17**

Table 6
AUC scores for PHLP (dim 0) with various (k, l)-angle hops.

Dataset	(1,0)	(1,1)	
USAir	**96.15 ± 0.83**	95.87 ± 0.83	
NS	98.28 ± 0.55	**98.66 ± 0.66**	
PB	93.95 ± 0.34	**94.46 ± 0.36**	
Yeast	95.52 ± 0.32	**97.31 ± 0.20**	
C.ele	86.18 ± 2.12	**87.57 ± 1.20**	
Power	73.39 ± 0.99	**77.83 ± 1.44**	
Router	92.09 ± 0.57	**93.25 ± 0.47**	
E.coli	96.94 ± 0.24	**96.95 ± 0.28**	
Dataset	**(2,0)**	**(2,1)**	**(2,2)**
USAir	96.69 ± 0.92	96.74 ± 0.84	**96.85 ± 0.83**
NS	**98.72 ± 0.51**	98.59 ± 0.65	98.56 ± 0.47
PB	94.78 ± 0.30	94.73 ± 0.30	**94.82 ± 0.24**
Yeast	**97.71 ± 0.18**	97.66 ± 0.27	97.58 ± 0.28
C.ele	88.86 ± 1.48	**89.16 ± 1.31**	89.08 ± 1.07
Power	80.27 ± 1.07	83.90 ± 1.29	**86.12 ± 0.86**
Router	95.65 ± 0.44	**95.71 ± 0.39**	94.51 ± 0.69
E.coli	97.26 ± 0.16	97.29 ± 0.24	**97.41 ± 0.21**

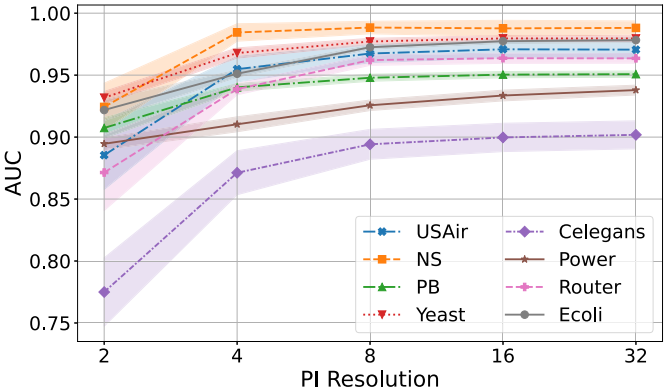

Fig. 7. Sensitivity analysis of MA-PHLP (dim 0) to persistence image resolution size $m \times m$ for $m \in \{ 2^i \mid i = 1, 2, \dots, 5 \}$. The AUC scores are averaged over 10 runs. The shaded regions denote the range of one standard deviation around the average.

Table 7
Ablation study on the use of persistence images computed with (w/) and without (w/o) the target link. Δ indicates the performance gap relative to the full setting (MA-PHLP (dim 0)), which utilizes both features jointly.

Dataset	MA-PHLP (dim 0)	w/ target only	w/o target only	Δ (w/)	Δ (w/o)
USAir	**97.10 ± 0.73**	96.59 ± 0.75	96.56 ± 0.68	−0.51	−0.54
NS	**98.78 ± 0.65**	98.01 ± 0.79	97.53 ± 0.66	−0.77	−1.25
PB	**95.06 ± 0.28**	94.71 ± 0.30	94.63 ± 0.31	−0.35	−0.43
Yeast	**97.98 ± 0.23**	97.38 ± 0.43	97.04 ± 0.28	−0.60	−0.94
C.ele	**89.88 ± 1.22**	88.86 ± 1.43	89.42 ± 1.39	−1.02	−0.46
Power	**93.37 ± 0.41**	91.30 ± 1.94	92.51 ± 0.95	−2.07	−0.86
Router	**96.37 ± 0.43**	95.39 ± 0.76	94.81 ± 0.55	−0.98	−1.56
E.coli	**97.72 ± 0.17**	97.30 ± 0.23	97.25 ± 0.25	−0.42	−0.47

Table 8
Comparison of AUC scores with TLC-GNN.

Dataset	GCN	TLC-GNN	MA-PHLP-GNN
Cora	92.20 ± 0.83	**93.16 ± 0.56**	93.14 ± 0.93
CiteSeer	86.52 ± 1.29	87.38 ± 0.97	**92.08 ± 0.53**
PubMed	96.63 ± 0.15	96.30 ± 0.25	**98.07 ± 0.07**

Similarly, we attempted to hybridize PHLP with the current SOTA model, WP. As presented in Table 4, a slight increase in AUC scores is observed for all datasets. The Power dataset demonstrates significant improvement. Detailed statistical significance analyses of these hybrid results, including paired t-tests [48] and 95 % bootstrap confidence intervals [53], are provided in Appendix C.

4.3. Ablation study

Effects of Degree DRNL. To assess the proposed Degree DRNL regarding the influence of incorporating degree information on model performance, we conducted experiments using DRNL and Degree DRNL and compared the results. We used MA-PHLP (dim 0) for the experiments. Table 5 presents the AUC scores of MA-PHLP (dim 0) with DRNL and Degree DRNL. Across all datasets, MA-PHLP (dim 0) yields higher AUC scores when used with Degree DRNL than with DRNL. The substantial improvement observed in the Power dataset is noteworthy, where Degree DRNL yields an increase of over 4 points in the AUC score. These experiments demonstrate the importance of incorporating degree information into node labeling, revealing its efficacy in enhancing the performance of MA-PHLP.

Angles of PHLP. Table 6 presents the performance of PHLP (dim 0) concerning various (k, l)-angle hop subgraphs. Section 3.2 proposed angle hop subgraphs as an alternative to the k-hop subgraphs to capture information from various perspectives. Moreover, MA-PHLP is proposed to aggregate information from multiple angles. To investigate performance when extracting information from specific angles, we conducted experiments using PHLP at different angles. We used only zero-dimensional PIs for the experiments. Overall, the results demonstrate that the performance is favorable for cases corresponding to the k-hop subgraph (where k and l are the same). Some datasets perform better when k and l differ, highlighting the importance of varying angles to achieve the best performance. Therefore, using MA-PHLP is recommended to maximize performance consistently across datasets.

PI resolution. We conducted experiments to examine how the performance of MA-PHLP (dim 0) varies with respect to the PI resolution grid size $m \times m$ for $m \in \{ 2^i \mid i = 1, 2, \dots, 5 \}$, as shown in Fig. 7. The figure demonstrates that, as long as the resolution is not too small, the performance remains stable, suggesting that PI resolution is not a highly sensitive hyperparameter. Based on these results, we chose $m = 16$ for all experiments in this paper as a balanced choice between computational cost and performance. All experimental settings, except for the PI resolution, are the same as those used for MA-PHLP (dim 0) in Table 1.

Effectiveness of PIs from the subgraphs $\mathcal{N}^+$ and $\mathcal{N}^-$. We conducted experiments to investigate the individual contributions of features derived from the subgraph $\mathcal{N}^+$ containing the target link and the subgraph $\mathcal{N}^-$ without it (notation used in Section 3.5). While MA-PHLP (dim 0) uses both PI features x^+ and x^- computed from $\mathcal{N}^+$ and $\mathcal{N}^-$ respectively, this experiment evaluates MA-PHLP (dim 0) when only one of these representations is used. The results are presented in Table 7.

Across all datasets, the joint use of both PI features consistently outperforms the use of either feature alone. This highlights the effectiveness of combining both features in better capturing the topological information relevant to link prediction. The largest performance gaps were observed in the Power and Router datasets, with decreases of 2.07 and 1.56, respectively. These datasets also exhibit the lowest densities as shown in Table 2, suggesting that incorporating both x^+ and x^- is particularly valuable when structural information is limited.

Comparison with TLC-GNN. To demonstrate that the proposed method extracts superior topological information compared to the conventional TLC-GNN approach, we conducted further experiments. The TLC-GNN was constructed by augmenting the graph convolutional network (GCN) with PI vectors, followed by an MLP classifier that takes the concatenation of GCN embeddings and PI vectors as input.

Table 9
AUC scores on the power dataset varying k-hop and max hop $k_{\max}$ of the hybrid methods.

		MA-PHLP (with max hop $k_{\max}$)						
$k_{\max}$		1	2	3	4	5	6	7
SEAL (with k-hop)	k	not robust to k			robust to k			
	1	86.66 ± 0.56	90.22 ± 0.79	92.63 ± 0.54	94.50 ± 0.41	95.12 ± 0.40	95.46 ± 0.38	**95.53 ± 0.33**
	2	91.40 ± 0.88	90.20 ± 0.80	92.50 ± 0.59	94.39 ± 0.39	95.00 ± 0.46	95.31 ± 0.40	**95.39 ± 0.36**
	3	93.21 ± 0.64	92.79 ± 0.60	92.57 ± 0.58	94.22 ± 0.43	94.86 ± 0.42	**95.21 ± 0.45**	95.19 ± 0.44
	4	94.51 ± 0.58	94.23 ± 0.34	94.21 ± 0.41	94.31 ± 0.40	94.80 ± 0.37	95.10 ± 0.33	**95.27 ± 0.36**
	5	94.73 ± 0.56	94.45 ± 0.44	94.61 ± 0.51	94.80 ± 0.53	94.91 ± 0.54	95.13 ± 0.51	**95.19 ± 0.46**
	6	94.58 ± 0.94	94.81 ± 0.32	94.87 ± 0.42	95.06 ± 0.50	95.11 ± 0.46	**95.25 ± 0.45**	95.25 ± 0.46
	7	93.97 ± 0.73	94.22 ± 0.35	94.43 ± 0.44	94.78 ± 0.45	94.92 ± 0.39	**94.99 ± 0.52**	94.98 ± 0.39
WP (with k-hop)	k	not robust to k		robust to k				
	1	87.53 ± 0.73	91.48 ± 0.64	93.55 ± 0.48	94.84 ± 0.43	95.53 ± 0.46	95.88 ± 0.31	**96.09 ± 0.38**
	2	92.51 ± 0.58	91.59 ± 0.77	93.49 ± 0.58	94.83 ± 0.53	95.56 ± 0.59	95.88 ± 0.38	**96.06 ± 0.45**
	3	94.04 ± 0.46	93.07 ± 0.67	93.61 ± 0.52	94.86 ± 0.54	95.61 ± 0.60	95.86 ± 0.40	**96.00 ± 0.52**
	4	93.55 ± 0.71	92.61 ± 0.76	93.68 ± 0.55	94.85 ± 0.55	95.59 ± 0.58	95.87 ± 0.38	**96.03 ± 0.45**
	5	93.40 ± 0.70	92.64 ± 0.69	93.66 ± 0.53	94.84 ± 0.54	95.55 ± 0.59	95.85 ± 0.39	**96.04 ± 0.52**
	6	93.34 ± 0.75	92.66 ± 0.72	93.64 ± 0.55	94.91 ± 0.57	95.55 ± 0.58	95.85 ± 0.44	**95.98 ± 0.55**
	7	93.30 ± 0.73	92.61 ± 0.69	93.65 ± 0.56	94.87 ± 0.56	95.56 ± 0.58	95.90 ± 0.39	**96.01 ± 0.52**

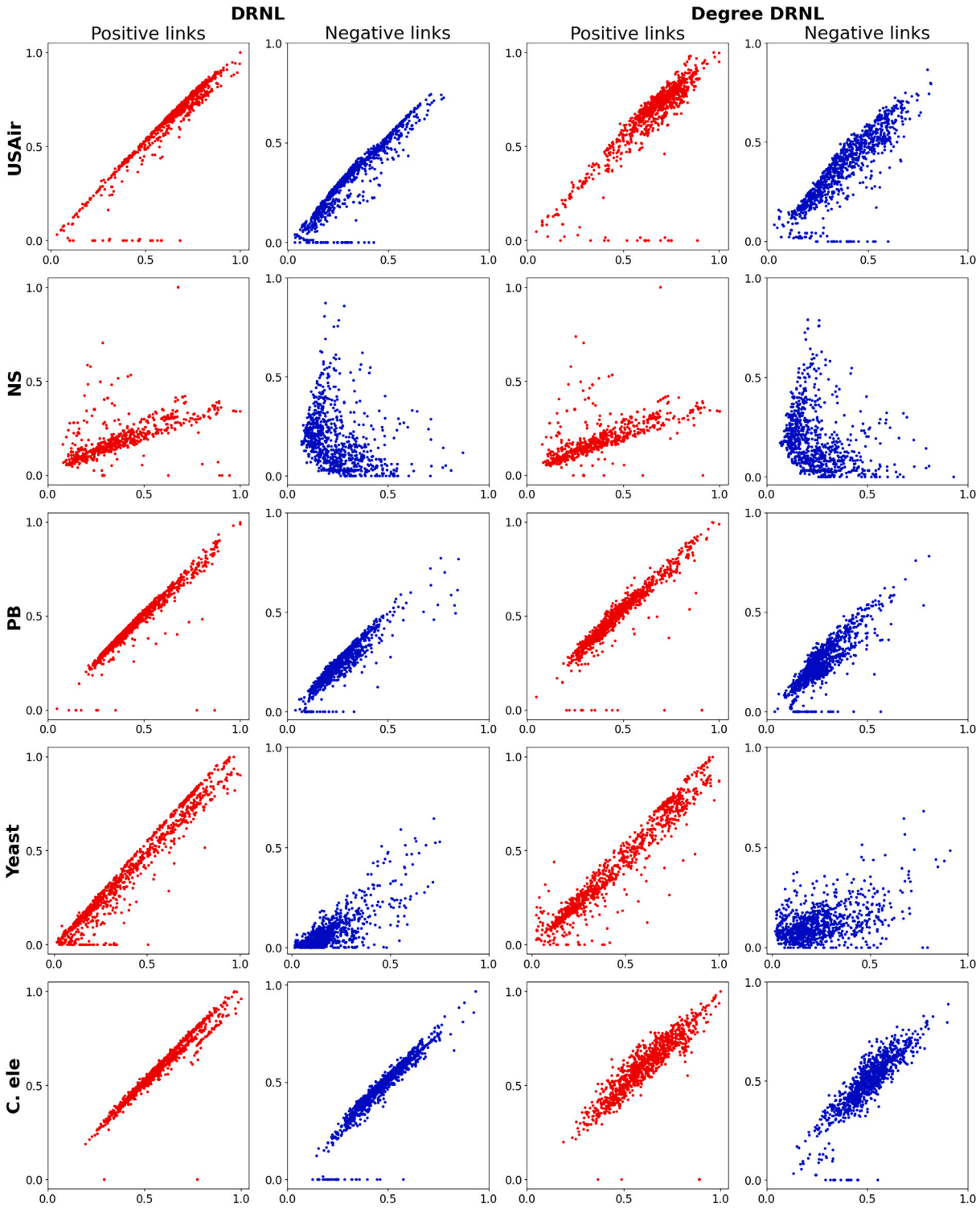

Fig. 8. Visualization of vectors calculated using MA-PHLP (dim 0). For each dataset, the first and second columns depict the projections of persistence images when double radius node labeling (DRNL) is applied for node labeling, and the third and fourth columns represent the values obtained when Degree DRNL is applied. The first and third columns plot the values produced from positive edges (i.e., target nodes labeled 1), and the second and fourth columns plot the values produced from negative edges (i.e., target nodes labeled 0).

To ensure a fair and isolated comparison of topological features, we replaced the PI component of the TLC-GNN model with the PI vector produced by MA-PHLP, resulting in the MA-PHLP-GNN model. Both TLC-GNN and MA-PHLP-GNN use the same GCN backbone and node attributes to compute the base representations, and differ only in how the PI vectors are computed. Therefore, any observed performance difference directly reflects the effect of topological feature construction. The zero-dimensional PH was employed in this study for fair comparison because TLC-GNN used only zero-dimensional PH. Additionally, we conducted experiments where the PI vectors were replaced with zero vectors, denoted as GCN. Table 8 presents the experimental results.

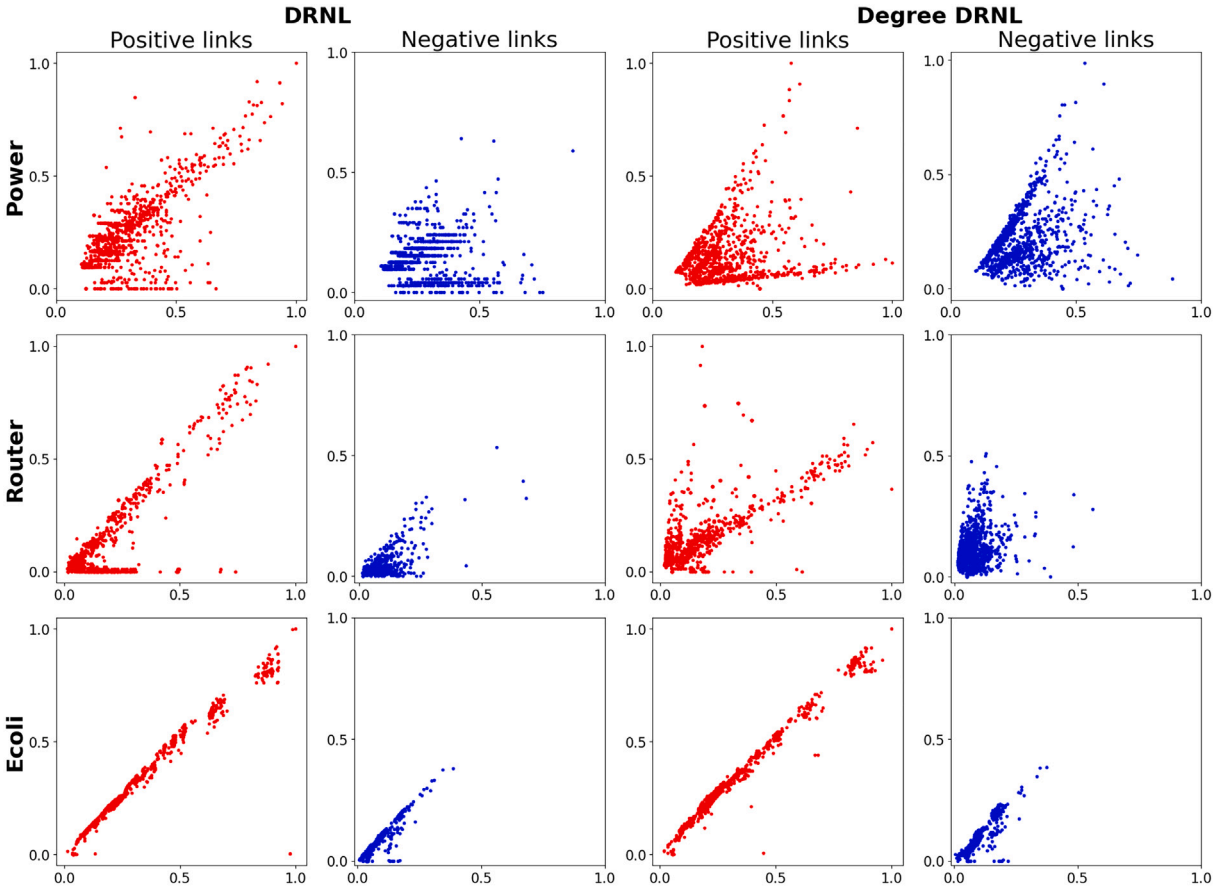

Fig. 9. Visualization of vectors calculated using MA-PHLP (dim 0).

TLC-GNN is designed for attributed datasets. Hence, we conducted experiments using the following widely used benchmark datasets with node attributes: Cora [37], CiteSeer [21], and PubMed [38]. It is important to emphasize that all models, including GCN, TLC-GNN, and MA-PHLP-GNN, utilized node attributes in the same way to ensure a controlled comparison. The only difference between the models lies in whether and how topological features are generated. The MA-PHLP-GNN outperformed the TLC-GNN significantly on the CiteSeer and PubMed datasets while achieving similar performance on the Cora dataset. The TLC-GNN does not exhibit performance improvement for the PubMed dataset despite adding topological information. However, the proposed MA-PHLP-GNN demonstrates substantial performance enhancement. Although the proposed model is developed for datasets without node attributes, it exhibits effective performance on datasets with node attributes through hybridization with the existing methods: SEAL + PHLP, WP + PHLP, and MA-PHLP-GNN. These experiments verify the versatility and effectiveness of this approach across diverse datasets.

4.4. The hops and max hops of the hybrid methods

Determining the hyperparameters such as "hop" and "max hop" is crucial for the performance of the hybrid method. We conducted experiments to explore the effects of different combinations of these parameters. Given that the hybrid methods (e.g., MA-PHLP + SEAL and MA-PHLP + WP) exhibited the highest performance improvement on the Power dataset, we conducted experiments on the Power dataset. Table 9 presents the AUC scores for varying hop (SEAL or WP) and max hop (MA-PHLP). For each target node, while the SEAL and WP extract a k-hop subgraph, the MA-PHLP calculates the PIs based on a subgraph with max hop $k_{\max}$. When the parameter $k_{\max}$ is 1 or 2, the AUC scores are not robust to k, showing large variations; however, when $k_{\max}$ is 3, although MA-PHLP + SEAL still exhibits variations up to 2, MA-PHLP + WP shows only minor variations. As $k_{\max}$ exceeds 3, the AUC scores of MA-PHLP + SEAL and MA-PHLP + WP are robust to k, exhibiting little sensitivity (maximum 0.84) to variations. This suggests that setting both the hop and the max hop to identical values may be permissible without further searching for optimal hyperparameters.

5. Analysis

5.1. Analysis of the PHLP

Figs. 8 and 9 visualize concatenated PIs to illustrate how MA-PHLP (dim 0) extracts topological features for LP. We let $\mathcal{Z} \subseteq \mathbb{R}^{2\times k\times r^2}$ be a set of vectors calculated by MA-PHLP, where k is the number of angles, and r denotes the PI resolution. For $(z_1, z_2) \in \mathcal{Z}$, $z_1 \in \mathbb{R}^{k\times r^2}$ is the concatenation of PIs for all angles with a target link, and $z_2 \in \mathbb{R}^{k\times r^2}$ is the concatenation for cases without a target link. We consider a function $h : \mathbb{R}^{k\times r^2} \to \mathbb{R}$ defined as $h(\vec{v}_1, \dots, \vec{v}_k) = \frac{1}{k}\sum_{i=1}^{k} \|\vec{v}_i\|_1$, where $\vec{v}_i \in \mathbb{R}^{r^2}$ are PIs, and $\|\cdot\|_1$ denotes the L_1-norm. For visualization, we transform $\mathcal{Z}$ into points in $\mathbb{R}^2$ using the function G, defined as $G(z_1, z_2) = (h(z_1), h(z_2))$ for each $(z_1, z_2) \in \mathcal{Z}$.

Unlike dimensionality reduction techniques such as t-SNE or PCA, which rely on optimization-based procedures, we visualize the PI vectors using the explicitly defined function G. This approach avoids the opacity associated with black-box embeddings and provides a deterministic and interpretable projection. By directly mapping the high-dimensional PI vectors onto a 2D plane, our method allows a transparent understanding of how filtration and homology information influence the resulting vectors. The visualizations using t-SNE and PCA are in Appendix F.

We plot distributions of points separately for positive and negative links, considering both DRNL and Degree DRNL. The distributions of

Table 10
Average number of nodes in subgraphs for the Power and Router datasets.

	Power		Router	
	positive	negative	positive	negative
1-hop	8.03	9.12	5.11	6.72
2-hop	22.26	24.85	29.21	13.94
3-hop	43.11	49.50	120.35	55.22
4-hop	71.72	82.16	411.87	176.34
5-hop	99.28	116.75	740.80	411.35
6-hop	136.23	158.27	1272.42	852.13
7-hop	182.22	210.35	1835.46	1498.58

the NS and Yeast datasets between positive and negative links display significant differences, supporting the highest performance in Table 5. In contrast, the distributions for the C. ele and Power datasets are the most similar when using Degree DRNL, correlating with the lowest scores in Table 5.

5.2. Analysis of the power dataset

In most LP models, including the SOTA models SEAL and WP, the Power dataset tends to have the lowest AUC scores among the datasets. In Table 1, the Power dataset is at the bottom in terms of scores across models (e.g., WLK, WLNM, MF, LINE, SEAL, and WP). However, the proposed model achieves the highest AUC scores on the Power dataset among baseline models, prompting an analysis of the reasons for this performance.

In Fig. 9, for DRNL, the Power dataset exhibits horizontal lines, indicating that the values $h(z_2)$ have a limited range of outcomes for vectors z_2 in cases without the target link; thus, the set of values $h(z_2)$ with the same value should be spread out. This observation implies that, for numerous subgraphs the calculation of PIs yields similar outcomes despite the differences in their topological structures, posing a challenge in distinguishing between them. To address this problem, we applied Degree DRNL, which incorporates degree information. The points in Fig. 9 are distributed without horizontal lines, leading to the highest score increase, as listed in Table 5.

The performance of heuristic methods, such as AA, Katz, and PR, tends to be similar to random guessing on datasets with low density, particularly in the cases of the Power and Router datasets. Embedding methods also display low performance. In contrast, the GNN-based methods demonstrate improved performance using subgraphs and the network learning ability. However, the performance for the Power dataset is significantly lower than that for the Router dataset.

To bridge this gap, we analyzed subgraphs with node labeling. The number of nodes within the selected subgraphs between positive and negative links was significantly different in the Router dataset but not in the Power dataset (Table 10). This difference is attributed to the presence of the hub nodes in the Router dataset, which are connected to numerous nodes. Thus, the subgraphs corresponding to positive links tend to have more nodes than those corresponding to negative links.

However, the Power dataset does not have hub nodes, and the number of nodes in the subgraph of positive links remains small.

We randomly changed the center nodes (a, b) for node labeling $f^{(a,b)}_{\text{degdrnl}}(w)$ increasing the performance, as listed in Table 11. This outcome highlights that setting target nodes as the center nodes may not effectively analyze the topological structure in the case of small graphs. Furthermore, the performance for the Power dataset continues to increase with increasing hops (Table 11), achieving an AUC score of 95.87, which is significantly better than 92.11 for WP.

6. Limitations and future work

MA-PHLP generates graph features without any training process. However, it achieves the highest performance only on the Power dataset. In addition, while hybrid methods consistently lead to performance improvements, such approaches often sacrifice interpretability. Our future work will focus on designing hybrid frameworks that can improve predictive accuracy while preserving interpretability. Furthermore, we plan to extend our approach to directed and multi-relational graphs, such as knowledge graphs, as well as to dynamic graphs.

Table 11
Comparison of models by Max hop settings on the Power and Router datasets.

	Model	MA-PHLP	MA-PHLP	WP	MA-PHLP + WP
	Center	target	random	–	random
Power	1-hop	78.05 ± 1.20	85.66 ± 0.86	80.24 ± 0.95	87.53 ± 0.73
	2-hop	86.34 ± 1.04	90.52 ± 0.73	89.40 ± 1.00	91.59 ± 0.77
	3-hop	89.65 ± 0.64	91.90 ± 0.58	**92.11 ± 0.77**	93.61 ± 0.52
	4-hop	91.38 ± 0.53	92.67 ± 0.55	91.67 ± 0.80	94.85 ± 0.55
	5-hop	92.27 ± 0.40	93.06 ± 0.44	91.39 ± 0.78	95.55 ± 0.59
	6-hop	92.77 ± 0.47	93.16 ± 0.49	91.55 ± 0.83	95.85 ± 0.44
	7-hop	**93.06 ± 0.43**	**93.37 ± 0.41**	91.50 ± 0.89	**96.01 ± 0.52**
Router	1-hop	93.12 ± 0.45	93.40 ± 0.46	94.48 ± 0.36	94.83 ± 0.41
	2-hop	95.96 ± 0.40	95.70 ± 0.45	97.15 ± 0.27	97.22 ± 0.23
	3-hop	96.38 ± 0.41	96.11 ± 0.43	**97.28 ± 0.24**	**97.42 ± 0.27**
	4-hop	96.45 ± 0.40	96.22 ± 0.43	OOM[1]	OOM
	5-hop	**96.46 ± 0.42**	**96.24 ± 0.48**	OOM	OOM
	6-hop	96.44 ± 0.45	96.23 ± 0.47	OOM	OOM
	7-hop	96.43 ± 0.45	96.19 ± 0.49	OOM	OOM

[1] OOM denotes “out of GPU memory”.

7. Conclusion

This paper proposes PHLP, an explainable method that applies PH to analyze the topological structure of graphs to overcome the limitations of GNN-based methods for LP. By employing the proposed methods, such as angle hop subgraphs and Degree DRNL, PHLP improves the analysis of the topological structure of graphs. The experimental results demonstrate that the proposed PHLP method achieves competitive performance across benchmark datasets, even SOTA performance, especially on the Power dataset. Additionally, when integrated with existing GNN-based methods, PHLP improves performance across all datasets. By analyzing the topological information of the given graphs, PHLP addresses the limitations of GNN-based methods and enhances overall performance. As demonstrated, PHLP provides explainable algorithms without relying on complex deep learning techniques, providing insight into the factors that significantly influence performance for the LP problem of graph data.

CRediT authorship contribution statement

Junwon You: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Eunwoo Heo:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Jae-Hun Jung:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Research Foundation of Korea Grant numbers under the grant number 2021R1A2C3009648, RS-2023–00219980 and RS -2021-NR060139.

Appendix A. The proof of theorem

Proof of Theorem 1. Let $G^{(u,v)} = (V, E)$ be a graph with target nodes u, v in V and let $f^{(u,v)}$ and $g^{(u,v)}$ be two node labeling functions based on (u, v) defined on $G^{(u,v)}$. For simplicity, we denote these functions as f and g, respectively. Denote the edge-weight functions derived from f and g as $W(f)$ and $W(g)$, respectively. The power set $\mathbb{X}$ of the vertex set V is an abstract simplicial complex. Consider the Rips filtration function $h_f : \mathbb{X} \to \mathbb{R}$ defined as

$$h_f(\sigma) = \max\{W(f)(w, z) \mid \text{edge } \{w, z\} \subseteq \sigma\}$$

whenever $p \geq 1$ for p-simplex $\sigma \in \mathbb{X}$, and defined as $h_f(z) = 0$ for any 0-simplex z. Denote $h_f^{-1}((-\infty, \epsilon])$ as K_ϵ for $\epsilon \in \mathbb{R}$. Then, the Rips filtration is obtained as $K_{\epsilon_1} \hookrightarrow K_{\epsilon_2} \hookrightarrow \cdots \hookrightarrow K_{\epsilon_m} = \mathbb{X}$ for $\epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_m$. By the stability theorem of the persistence diagram [16], we know that $D_B(dgm_p(f), dgm_p(g)) \leq \|h_f - h_g\|_\infty$.

Now, we have $\|h_f - h_g\|_\infty = |(h_f - h_g)(\sigma_*)|$ for a p-simplex $\sigma_* \in \mathbb{X}$. By the maximality of h_f and h_g, there are edges (w_f, z_f) and $(w_g, z_g) \subseteq \sigma_*$ such that $h_f(\sigma_*) = W(f)(w_f, z_f)$ and $h_g(\sigma_*) = W(g)(w_g, z_g)$. Consider an interpolation map $\mathcal{I}(t) = (1 - t)W(f) + tW(g)$ for a real number t, where $\mathcal{I}(0) = W(f)$ and $\mathcal{I}(1) = W(g)$. Then, define a function $\mathcal{H}(t) = \mathcal{I}(t)(w_f, z_f) - \mathcal{I}(t)(w_g, z_g)$ for real number t. By the maximality of the functions h_f and h_g, we have $\mathcal{H}(0) \geq 0$ and $\mathcal{H}(1) \leq 0$. Given that $\mathcal{H}$ is continuous on the closed interval $[0, 1]$, the intermediate value theorem guarantees the existence of a real number t_0 within $[0, 1]$ such that $\mathcal{H}(t_0) = 0$. That is, if we write $\mathcal{I}(t_0)$ as W_*, we have $W_*(w_f, z_f) = W_*(w_g, z_g)$. Now, observe that

$$\begin{aligned}
\|h_f - h_g\|_\infty &= |(h_f - h_g)(\sigma_*)| \\
&= |W(f)(w_f, z_f) - W(g)(w_g, z_g)| \\
&\leq |W(f)(w_f, z_f) - W_*(w_f, z_f)| \\
&\quad + |W_*(w_g, z_g) - W(g)(w_g, z_g)| \\
&\leq \|W(f) - W_*\|_\infty + \|W_* - W(g)\|_\infty \\
&= \|W(f) - W(g)\|_\infty
\end{aligned}$$

The last equality is derived from the expression $W_* = (1 - t_0)W(f) + t_0\, W(g)$.

By the definition of infinity norm, there exist two vertices w and z such that $\|W(f) - W(g)\|_\infty = |W(f)(w, z) - W(g)(w, z)|$. Without loss of generality, we can assume $f(w) > f(z)$. Suppose $g(w) > g(z)$. Then we have $\|W(f) - W(g)\|_\infty = |W(f)(w, z) - W(g)(w, z)| = |f(w) - g(w)| \leq \|f - g\|_\infty$.

Next, suppose $g(w) \leq g(z)$. Then we have $\|W(f) - W(g)\|_\infty = |f(w) - g(z)|$. If $f(w) - g(z) > 0$, then $|f(w) - g(z)| = f(w) - g(z) \leq f(w) - g(w) \leq \|f - g\|_\infty$. If $f(w) - g(z) \leq 0$, then $|f(w) - g(z)| = g(z) - f(w) \leq g(z) - f(z) \leq \|f - g\|_\infty$. Thus we have $\|W(f) - W(g)\|_\infty \leq \|f - g\|_\infty$ and conclude $D_B(dgm_p(f), dgm_p(g)) \leq \|f - g\|_\infty$. This concludes the proof.

Appendix B. Algorithm of PHLP

We present the training algorithms for the PHLP and MA-PHLP, as shown in Algorithms 1 and 2.

Algorithm 1 PHLP Training algorithm.

Input: Observed graph $G = (V, E)$;
Classifier Φ with He-initialized weights θ;
Angle hop parameters (k, l), maximum epoch $Epochs$, batch size m;
Output: Trained classifier Φ with optimized weights θ;
Sample true target links $\{x_1, x_2, \ldots, x_n\}$ from observed links E and label them as 1;
Sample false target links $\{x_{n+1}, x_{n+2}, \ldots, x_{2n}\}$ from $V \times V \setminus E$ and label them as 0;
for $i = 1 \cdots 2n$ **do**
    Extract (k, l)-angle hop subgraph $\mathcal{N}_{x_i}^{(k,l)}$ for target link x_i;
    Compute Degree DRNL and assign edge weights to graphs with and without the target link, $\mathcal{N}^+$ and $\mathcal{N}^-$;
    Compute persistence diagrams D^+ and D^- for $\mathcal{N}^+$ and $\mathcal{N}^-$ using edge weights as filtration values;
    Vectorize D^+ and D^- into persistence images PI^+ and PI^-;
    If $k \neq l$, repeat the process for (l, k)-angle hop subgraph and average the two vectors;
    Concatenate PI^+ and PI^- to form the final vector PI_i;
end for
Form training dataset $\mathcal{X} = \{PI_1, PI_2, \ldots, PI_{2n}\}$ and corresponding labels $Y = \{1, \ldots, 1, 0, \ldots, 0\}$;
for $e = 1 \cdots Epochs$ **do**
    Sample a random batch $\{pi_1, pi_2, \ldots, pi_m\}$ from $\mathcal{X}$ and corresponding labels $\{y_1, y_2, \ldots, y_m\}$;
    Predict labels from classifier Φ: $p_1, p_2, \ldots, p_m$;
    Train the classifier using binary cross-entropy loss BCE and update weights θ:

$$\nabla_\theta \Big(\sum_{pi_i \in \mathcal{X}} BCE(p_i, y_i) \Big);$$

end for

Algorithm 2 MA-PHLP Training algorithm.

Input: Observed graph $G = (V, E)$;
Max hop $k_{\max}$, maximum epochs $Epochs$, batch size m;
Classifiers $\Phi = (\Phi_1, \dots, \Phi_N)$ with He-initialized weights $\theta = (\theta_1, \dots, \theta_N)$;
Trainable parameter $\alpha = (\alpha_1, \dots, \alpha_N)$, where $N = |\{(k,l) \in \mathbb{Z}^2 : 0 \le l \le k \le k_{\max}, k > 0\}|$;
Output: Trained classifiers Φ with optimized weights θ;
Optimized parameter α;
Sample true target links $\{x_1, x_2, \dots, x_n\}$ from observed links E and label them as 1;
Sample false target links $\{x_{n+1}, x_{n+2}, \dots, x_{2n}\}$ from $V \times V \setminus E$ and label them as 0;
for $i = 1 \cdots 2n$ **do**
  for $k = 1 \cdots k_{\max}$ **do**
    for $l = k \cdots k_{\max}$ **do**
      Extract (k,l)-angle hop subgraph $\mathcal{N}_{x_i}^{(k,l)}$ for target link x_i;
      Compute Degree DRNL and assign edge weights to graphs with and without the target link, $\mathcal{N}^+$ and $\mathcal{N}^-$;
      Compute persistence diagrams D^+ and D^- for $\mathcal{N}^+$ and $\mathcal{N}^-$ using edge weights as filtration values;
      Vectorize D^+ and D^- into persistence images PI^+ and PI^-;
      If $k \neq l$, repeat the same process with the (l,k)-angle hop subgraph and average the two vectors;
      Concatenate PI^+ and PI^- to form $PI_{(k,l)}$;
    end for
  end for
  $PIS_i = \{PI_{(k,l)} : 0 \le l \le k \le k_{\max}, k > 0\} = \{PI_1, PI_2, \dots, PI_N\}$;
end for
Form training dataset $\mathcal{X} = \{PIS_1, PIS_2, \dots, PIS_{2n}\}$ and corresponding labels $Y = \{1, \dots, 1, 0, \dots, 0\}$;
for $e = 1 \cdots Epochs$ **do**
  Sample a random batch $\{pis_1, pis_2, \dots, pis_m\}$ from $\mathcal{X}$ and corresponding labels $\{y_1, y_2, \dots, y_m\}$;
  Predict labels from classifiers as $p_i = \sum_{j=1}^{N} \alpha_j \Phi_j(pi_j)$, where $pis_i = (pi_1, \dots, pi_N)$;
  Train the classifiers and update α using binary cross-entropy loss BCE:

$$\nabla_{\theta,\alpha}\Big(\sum_{pis_i \in \mathcal{X}} BCE(p_i, y_i)\Big);$$

end for

Appendix C. Statistical significance analyses of hybrid methods

We computed paired t-test p-values [48] and 95 % bootstrap confidence intervals [53] between the base models (SEAL / WP) and their hybrid variants incorporating our topological features (MA-PHLP + SEAL/WP), as reported in Tables 12 and 13. When the topological features were added to SEAL, most datasets (e.g., USAir, NS, Yeast, Power, Router, E.coli) exhibited p-values below 0.05, indicating statistically significant improvements. In particular, the Power dataset achieved a confidence interval of $(7.118, 10.786)$ and $p < 10^{-5}$, demonstrating a strong effect of the topological features. Only PB and C.elegans showed non-significant differences ($p > 0.05$), implying that the improvement is not consistent across all datasets. Similarly, when applied to WP, the improvements were smaller in magnitude. While most datasets yielded p-values above 0.05, both PB and Power

Table 12
Statistical significance analysis between SEAL and MA-PHLP + SEAL (Paired t-test and 95 % CI).

Dataset	SEAL	MA-PHLP + SEAL	paired t-test (p)	Confidence intervals
USAir	97.10 ± 0.87	$\mathbf{97.41 \pm 0.62}$	0.05	$(0.08, 0.59)$
NS	98.25 ± 0.61	$\mathbf{98.97 \pm 0.30}$	3.1e-03	$(0.39, 1.05)$
PB	95.07 ± 0.39	$\mathbf{95.14 \pm 0.39}$	0.12	$(-0.01, 0.14)$
Yeast	97.60 ± 0.33	$\mathbf{97.93 \pm 0.18}$	0.05	$(0.05, 0.59)$
C.ele	89.54 ± 1.23	$\mathbf{89.61 \pm 1.12}$	0.83	$(-0.52, 0.59)$
Power	86.21 ± 2.89	$\mathbf{95.53 \pm 0.33}$	9.3e-06	$(7.12, 10.79)$
Router	95.07 ± 1.63	$\mathbf{96.15 \pm 1.26}$	4.7e-03	$(0.83, 2.48)$
E.coli	97.57 ± 0.30	$\mathbf{97.93 \pm 0.34}$	1.8e-03	$(0.21, 0.51)$

Table 13
Statistical significance analysis between WP and MA-PHLP + WP (Paired t-test and 95 % CI).

Dataset	WP	MA-PHLP + WP	paired t-test (p)	Confidence intervals
USAir	98.20 ± 0.57	$\mathbf{98.27 \pm 0.53}$	0.57	$(-0.12, 0.29)$
NS	99.12 ± 0.45	$\mathbf{99.24 \pm 0.32}$	0.13	$(0.00, 0.23)$
PB	95.42 ± 0.25	$\mathbf{95.58 \pm 0.32}$	2.3e-03	$(0.08, 0.22)$
Yeast	98.21 ± 0.17	$\mathbf{98.25 \pm 0.18}$	0.12	$(-0.0, 0.1)$
C.ele	93.30 ± 0.91	$\mathbf{93.32 \pm 0.71}$	0.64	$(-0.13, 0.21)$
Power	92.11 ± 0.76	$\mathbf{96.09 \pm 0.38}$	1.4e-08	$(3.64, 4.42)$
Router	97.15 ± 0.29	$\mathbf{97.18 \pm 0.24}$	0.43	$(-0.05, 0.13)$
E.coli	98.54 ± 0.19	$\mathbf{98.57 \pm 0.20}$	0.40	$(-0.02, 0.07)$

again demonstrated statistically significant gains ($p < 0.01$). These results suggest that the proposed topological features provide statistically verifiable performance enhancements in particular datasets.

Appendix D. Robustness analysis of the proposed method

Experiments with a reduced percentage of observed links. Following prior work, we conducted additional experiments with only 50 % of the links observed. The results are presented in Table 14. Except for the results of LGLP, MPLP, and MA-PHLP (dim 0), which we reproduce, all others are directly taken from the WP paper. Below each score, we report the drop in AUC scores (▼) from the 90 % to the 50 % observed setting. Reducing the number of observed links substantially degrades the performance of all models. Notably, MA-PHLP (dim 0) continues to achieve the best performance on the Power dataset, even under the reduced setting, demonstrating strong robustness.

Gaussian noise on node labeling. Theoretically, the stability of persistent homology under variations in the filtration function on simplicial complexes is well established. In this paper, we further prove that similar stability results can be extended to perturbations in the node labeling. This theoretical guarantee is formalized in Theorem 1, with a detailed proof provided in Appendix A.

Our model computes edge weights based on node labels obtained via the Degree DRNL function. As such, perturbing node labels directly affects the computed edge weights and hence the persistent homology computations. To test the model's robustness to this kind of noise, we add Gaussian noise to each node label individually. Formally, given the Degree DRNL labeling function $f : V \to \mathbb{R}$, we define the perturbed labeling $\tilde{f}_\lambda$ as

$$\tilde{f}_\lambda(v) = f(v) + \lambda \cdot \epsilon_v, \quad \epsilon_v \sim \mathcal{N}(0,1) \tag{12}$$

for each vertex $v \in V$, where $\lambda \in \{0.1, 0.5, 1, 5, 10\}$ is a scalar coefficient controlling the noise level.

Table 15 reports the results of this experiment, where we evaluate MA-PHLP (dim 0) across all benchmark datasets under 5 different random seeds $1, 2, \ldots, 5$. The performance remains stable for small scalar coefficients $\lambda = 0.1, 0.5, 1$, while a gradual performance degradation is observed in most cases as the scalar coefficients increase to $\lambda = 5$ and $\lambda = 10$, suggesting that the model is robust to moderate label perturbations.

Table 14
AUC scores at 50 % observed links (mean ± std) and drop in AUC scores (▼) across datasets.

Dataset	USAir	NS	PB	Yeast	C. ele	Power	Router	E. coli
AA	88.61 ± 0.40 ▼6.45	77.13 ± 0.75 ▼17.32	87.06 ± 0.17 ▼5.30	82.63 ± 0.27 ▼6.80	73.37 ± 0.80 ▼13.58	53.38 ± 0.22 ▼5.41	52.94 ± 0.28 ▼3.49	87.66 ± 0.56 ▼7.70
Katz	88.91 ± 0.51 ▼3.97	82.30 ± 0.93 ▼12.55	91.25 ± 0.22 ▼1.67	88.87 ± 0.28 ▼3.37	79.99 ± 0.59 ▼6.35	57.34 ± 0.51 ▼8.05	54.39 ± 0.38 ▼15.77	89.81 ± 0.46 ▼3.69
PR	90.57 ± 0.62 ▼4.10	82.32 ± 0.94 ▼12.57	92.23 ± 0.21 ▼1.31	89.35 ± 0.29 ▼3.41	84.95 ± 0.58 ▼5.37	57.34 ± 0.52 ▼8.66	54.44 ± 0.38 ▼15.68	92.96 ± 0.43 ▼2.61
WLK	91.93 ± 0.71 ▼4.70	87.27 ± 1.71 ▼11.30	92.54 ± 0.33 ▼1.29	91.15 ± 0.35 ▼4.71	83.29 ± 0.89 ▼6.43	63.44 ± 1.29 ▼18.97	71.25 ± 4.37 ▼16.17	92.38 ± 0.46 ▼4.56
WLNM	91.42 ± 0.95 ▼4.53	87.61 ± 1.63 ▼11.00	90.93 ± 0.23 ▼2.56	92.22 ± 0.32 ▼3.40	75.72 ± 1.33 ▼10.46	64.09 ± 0.76 ▼20.67	86.10 ± 0.52 ▼8.31	92.81 ± 0.30 ▼4.40
N2V	84.63 ± 1.58 ▼6.81	80.29 ± 1.20 ▼11.23	79.29 ± 0.67 ▼6.50	90.18 ± 0.17 ▼3.49	75.53 ± 1.23 ▼8.58	55.40 ± 0.84 ▼20.82	62.45 ± 0.81 ▼3.01	84.73 ± 0.81 ▼6.09
SPC	65.42 ± 3.41 ▼8.80	79.63 ± 1.34 ▼10.31	78.06 ± 1.00 ▼5.10	89.73 ± 0.28 ▼3.52	47.30 ± 0.91 ▼4.60	56.51 ± 0.94 ▼9.33	53.87 ± 1.33 ▼14.92	92.00 ± 0.50 ▼2.92
MF	91.28 ± 0.71 ▼2.80	62.95 ± 1.03 ▼11.60	93.27 ± 0.16 ▼1.03	84.99 ± 0.49 ▼5.29	78.49 ± 1.73 ▼7.41	50.53 ± 0.60 ▼0.10	77.49 ± 0.64 ▼0.54	91.75 ± 0.33 ▼1.99
LINE	72.51 ± 12.19 ▼8.96	65.96 ± 1.60 ▼14.67	75.53 ± 1.78 ▼1.42	79.44 ± 7.90 ▼8.01	59.46 ± 7.08 ▼9.75	53.44 ± 1.83 ▼2.19	62.43 ± 3.10 ▼4.72	74.50 ± 11.10 ▼7.88
SEAL	93.36 ± 0.67 ▼3.73	90.88 ± 1.18 ▼7.97	93.79 ± 0.25 ▼1.22	93.90 ± 0.54 ▼4.01	82.33 ± 2.31 ▼7.97	65.84 ± 1.10 ▼21.77	86.64 ± 1.58 ▼9.74	94.18 ± 0.41 ▼3.46
WP	95.16 ± 0.70 ▼3.18	90.68 ± 1.04 ▼7.98	94.50 ± 0.20 ▼1.03	94.89 ± 0.22 ▼3.37	87.83 ± 0.83 ▼5.17	67.03 ± 0.77 ▼24.84	88.09 ± 0.52 ▼9.14	95.37 ± 0.22 ▼3.25
LGLP	94.97 ± 0.12 ▼2.12	91.19 ± 0.11 ▼7.93	94.15 ± 0.05 ▼0.55	94.61 ± 0.13 ▼2.92	83.53 ± 0.12 ▼5.11	66.48 ± 0.27 ▼19.15	85.88 ± 0.26 ▼9.63	95.01 ± 0.09 ▼3.38
MPLP	93.24 ± 1.36 ▼3.77	86.98 ± 0.77 ▼9.19	93.15 ± 0.47 ▼0.91	90.34 ± 0.30 ▼3.91	86.55 ± 0.88 ▼3.93	58.06 ± 0.70 ▼15.65	85.91 ± 0.58 ▼5.99	94.09 ± 0.24 ▼2.58
MA-PHLP (dim 0)	93.79 ± 0.71 ▼3.31	89.14 ± 0.89 ▼9.74	94.32 ± 0.14 ▼0.78	94.07 ± 0.32 ▼3.91	83.62 ± 1.44 ▼6.71	72.32 ± 1.13 ▼21.05	87.96 ± 0.65 ▼8.41	94.48 ± 0.22 ▼3.24

Table 15
AUC scores of MA-PHLP (dim 0) using noisy node labels $\tilde{f}_\lambda$ with varying $\lambda \in \{0.1, 0.5, 1, 5, 10\}$.

Dataset	Degree DRNL	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$
USAir	96.56 ± 0.61	96.65 ± 0.66	96.49 ± 0.57	96.30 ± 0.56	93.71 ± 1.12	92.63 ± 1.32
NS	99.00 ± 0.46	99.03 ± 0.47	99.07 ± 0.41	98.94 ± 0.53	98.03 ± 0.41	97.65 ± 0.53
PB	94.96 ± 0.19	95.00 ± 0.21	94.94 ± 0.21	94.79 ± 0.21	94.65 ± 0.25	95.33 ± 0.27
Yeast	97.93 ± 0.12	97.93 ± 0.15	97.84 ± 0.17	97.70 ± 0.20	96.91 ± 0.27	96.41 ± 0.18
C.ele	89.56 ± 1.17	89.51 ± 1.30	89.47 ± 1.35	89.02 ± 1.21	84.27 ± 1.02	82.73 ± 2.09
Power	86.31 ± 1.03	86.29 ± 1.08	85.86 ± 1.12	85.44 ± 1.10	83.95 ± 1.26	83.18 ± 1.25
Router	95.86 ± 0.49	95.69 ± 0.48	95.60 ± 0.42	95.44 ± 0.49	93.52 ± 0.64	92.69 ± 0.49
E.coli	97.66 ± 0.20	97.58 ± 0.17	97.49 ± 0.28	97.37 ± 0.31	96.30 ± 0.34	95.98 ± 0.37

Appendix E. Computational efficiency, memory usage, and scalability analysis

This section presents the analysis of computational efficiency, memory usage, and performance scalability of our proposed models. All experiments in this section were conducted on a server equipped with an AMD EPYC 7513 32-Core CPU and an NVIDIA A100 80GB PCIe GPU. For preprocessing and persistence image computation, we utilized all 32 CPU cores through multiprocessing.

Computation time and memory usage. While persistent homology is often considered computationally expensive, our implementation addresses this cost using three different strategies.

First, the PI vectors are computed only once for each subgraph prior to training, as part of a preprocessing step. These features are fixed and reused throughout training, avoiding repeated computations. Table 16 reports the computation time and memory usage required for generating PI vectors in the MA-PHLP (dim 0) model, measured on the validation set of each benchmark. The first row, **Avg. time per input (s)**, shows the average time required to compute the PI vector for a single input, measured using a single CPU core. The second row, **Total time (s)**, reports the total time taken to compute PI vectors for the validation set of each benchmark, measured using 32 CPU cores with multiprocessing. The third row, **Memory per input (KB)**, indicates the fixed memory required to store the PI vector for one input. The last row, **Total memory (MB)**, shows the total memory usage for storing all PI vectors in the validation set.

Second, the trainable part of our model consists solely of an MLP classifier, rather than a GNN architecture with an MLP classifier. This significantly reduces the computational complexity of our method. In contrast, WP uses attention during message passing, which makes it substantially more computationally expensive. Table 17 presents the time required for one training epoch for both the GNN-based models and our proposed MA-PHLP model.

Finally, we found that zero-dimensional features alone are sufficient to achieve strong performance. While persistent homology is generally known to be computationally expensive, zero-dimensional homology can be computed efficiently using only Kruskal's minimum spanning tree algorithm [32] with a union-find [52], and is therefore not computationally heavy. Moreover, we parallelize the PI extraction process using CPU-based multiprocessing, which significantly accelerates preprocessing and makes it practical even for large-scale datasets. Overall, the preprocessing time remains acceptable in our experimental settings.

Performance scale. Regarding the effect of angle hops, Table 6 shows how varying the number of angle hops impacts performance within our PHLP framework. We observe that performance improves with larger hops in some datasets. Furthermore, Table 9 illustrates a comparative analysis on the Power dataset using a hybrid model. It shows that increasing the hop size in PHLP leads to a more significant performance gain than in GNN-based models. For the Power dataset, we employed up to 7-hop neighborhoods. Thus, for this dataset, we measured the computation time for each (i, j)-angle hop setting. Table 18 reports the average PI generation time (in seconds) per (i, j)-angle pair, measured over 100 input samples from the Power dataset, computed using a single CPU core. Despite the increase in hop size, we observe a relatively moderate and predictable growth in computation time. This behavior is expected, as zero-dimensional persistent homology is computed via the minimum spanning tree algorithm.

Table 16
Computation time and memory usage for persistence image extraction using MA-PHLP (dim 0) across validation sets.

	USAir	NS	PB	Yeast	C.ele	Power†	Router	E.coli
Avg. time per input (s)	2.08	0.09	3.18	1.64	1.87	0.9	0.34	4.62
Total time (s)	89.10	9.85	901.52	412.81	89.09	334.19	72.13	862.52
Memory per input (KB)	36	36	36	36	36	140	36	36
Total memory (MB)	7.45	9.63	58.71	41.06	7.52	89.96	21.94	51.54

† The max hop was set to 7 for Power dataset.

Table 17
Training time per epoch (in seconds) for each model on different datasets.

	USAir	NS	PB	Yeast	C.ele	Power†	Router	E.coli
SEAL	2.48	2.18	63.65	12.45	2.41	4.82	5.52	59.86
WP	10.82	6.61	59.64	67.16	9.84	14.20	18.36	55.27
LGLP	6.92	1.02	35.74	31.85	1.86	2.25	2.67	65.90
MPLP	0.09	0.11	0.21	0.22	0.10	0.15	0.16	0.27
MA-PHLP	0.38	0.46	2.62	1.80	0.38	2.20	0.95	2.62
MA-PHLP (dim 0)	0.33	0.39	2.30	1.56	0.34	1.58	0.83	2.24

† The maximum hop was set to 7 for Power dataset.

Table 18
Average persistence image generation time (seconds) per (i, j)-angle pair over 100 input samples for the Power dataset in the MA-PHLP (dim 0) model, computed using a single CPU core.

(i, j)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	–	0.1935	0.2728	0.4046	0.6266	0.9873	1.8200	2.3139
1	0.1962	0.2286	0.3082	0.4405	0.6642	1.0243	1.5635	2.3626
2	0.2725	0.3080	0.3979	0.5036	0.7449	1.0816	1.6356	2.4320
3	0.4215	0.4568	0.5493	0.6773	0.8646	1.2090	1.7544	2.5546
4	0.6574	0.7030	0.7640	0.8974	1.1934	1.4156	1.9698	2.7662
5	1.0201	1.0412	1.1135	1.2574	1.4329	1.7507	2.2819	3.1010
6	1.5325	1.5305	1.6013	1.7204	1.9350	2.2339	2.7658	3.5680
7	2.2194	2.2461	2.3202	2.4557	2.6504	2.9761	3.4822	4.3073

Appendix F. Additional visualizations of persistence image features

We perform additional visualizations using standard visualization techniques, t-SNE (Figs. 10 and 11) and PCA (Figs. 12 and 13).

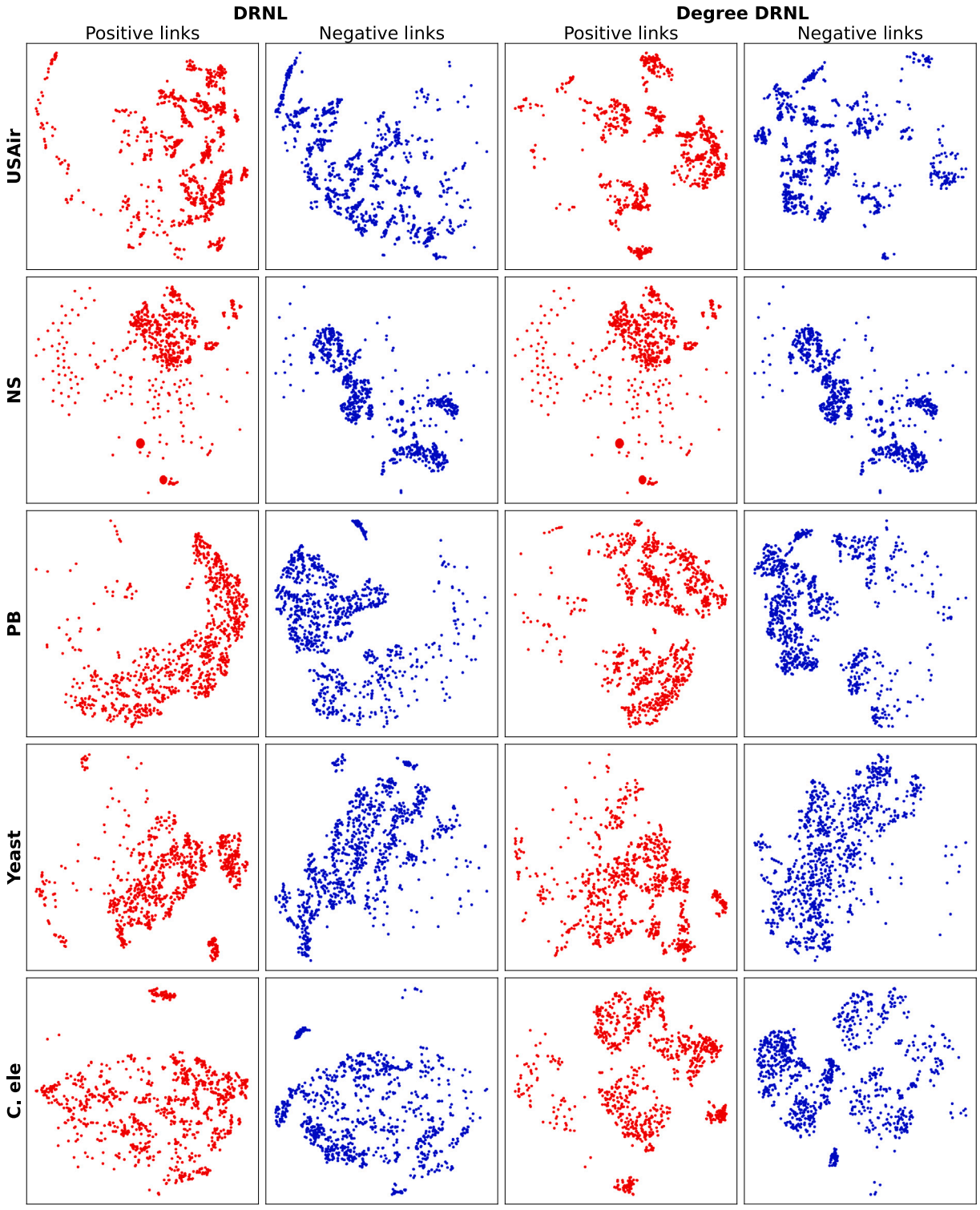

Fig. 10. Visualization of persistence image (PI) vectors calculated using MA-PHLP (dim 0). For each dataset, the first and second columns depict the visualization of PIs using t-SNE when double radius node labeling (DRNL) is applied for node labeling, and the third and fourth columns represent the values obtained when Degree DRNL is applied. The first and third columns plot the values produced from positive links (i.e., target nodes labeled 1), and the second and fourth columns plot the values produced from negative links (i.e., target nodes labeled 0).

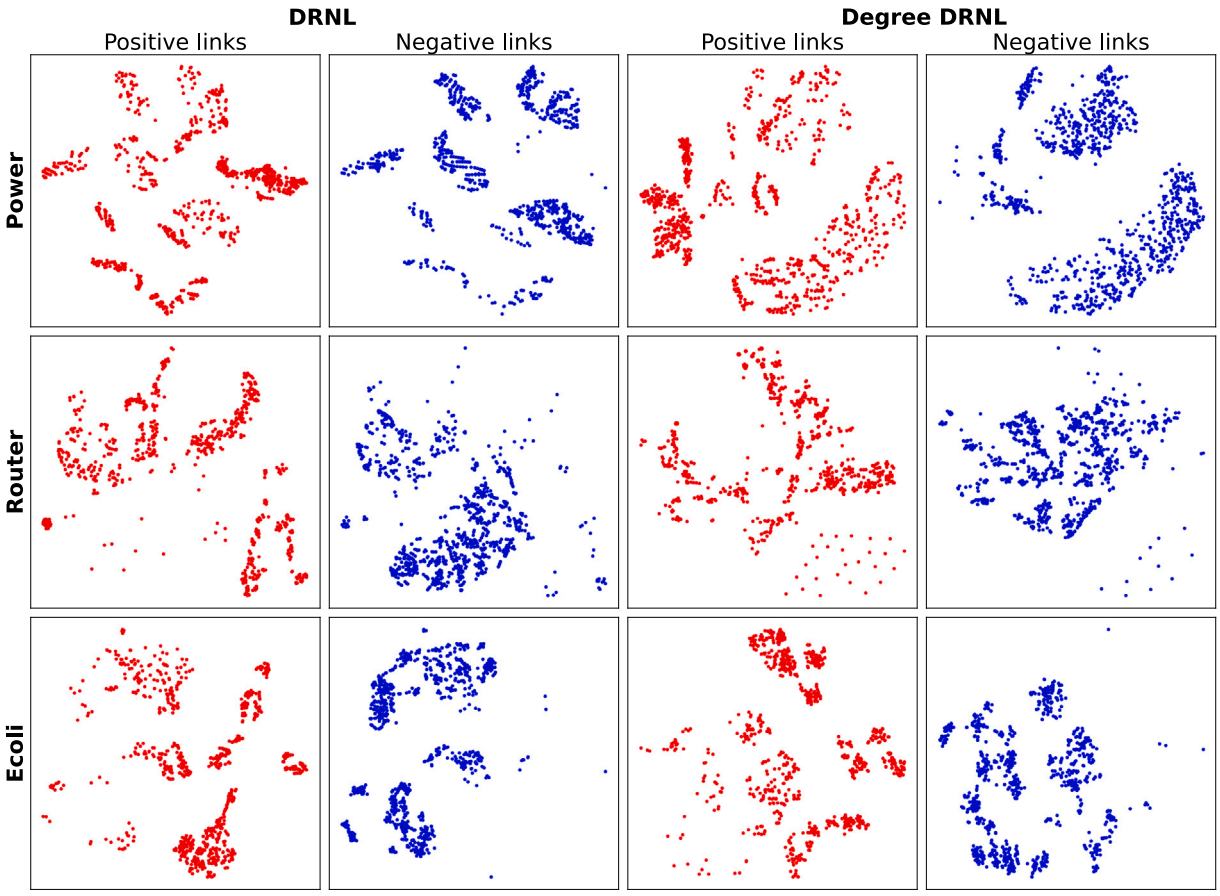

Fig. 11. Visualization of persistence image vectors calculated using MA-PHLP (dim 0) via t-SNE.

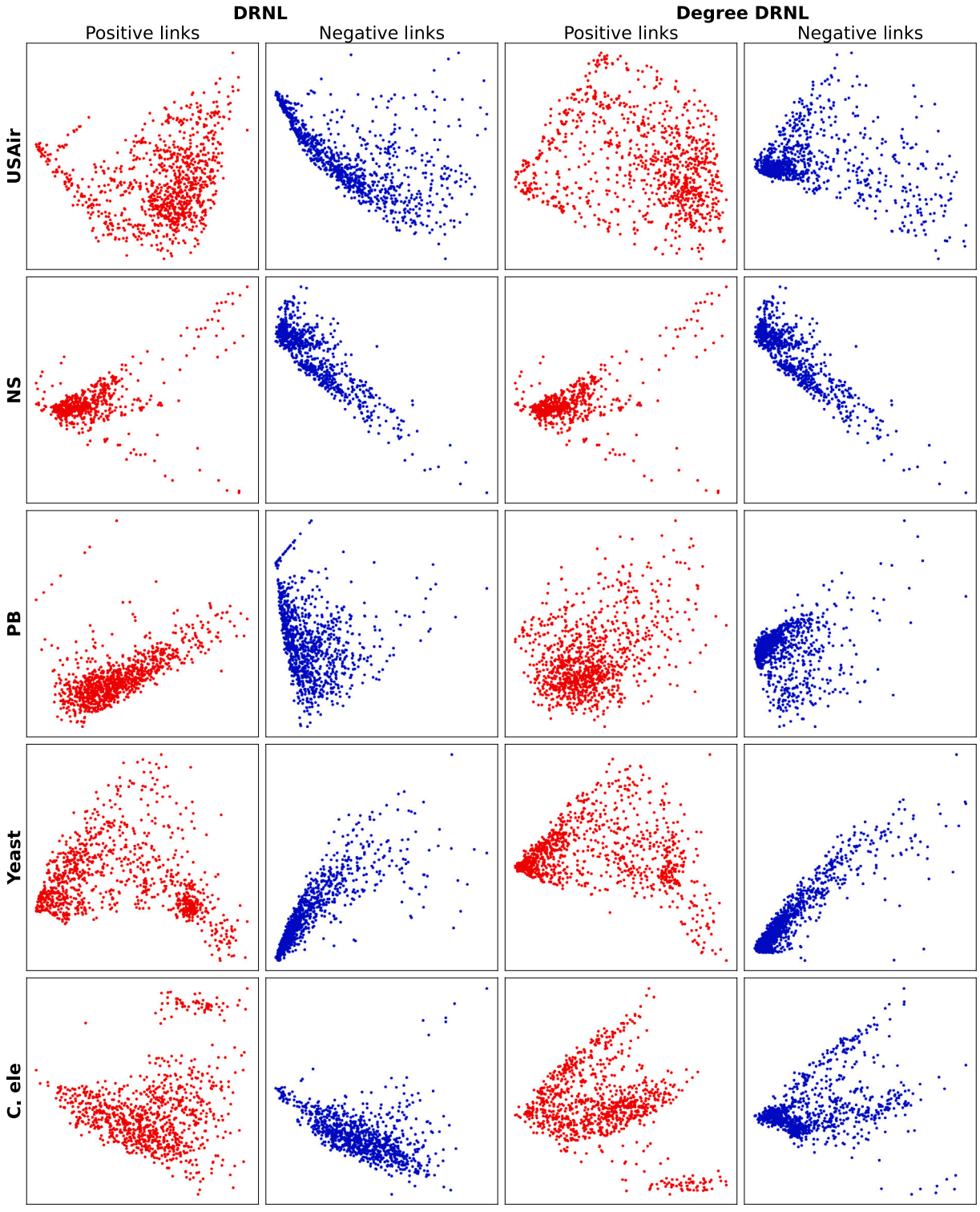

Fig. 12. Visualization of persistence image vectors calculated using MA-PHLP (dim 0) via PCA.

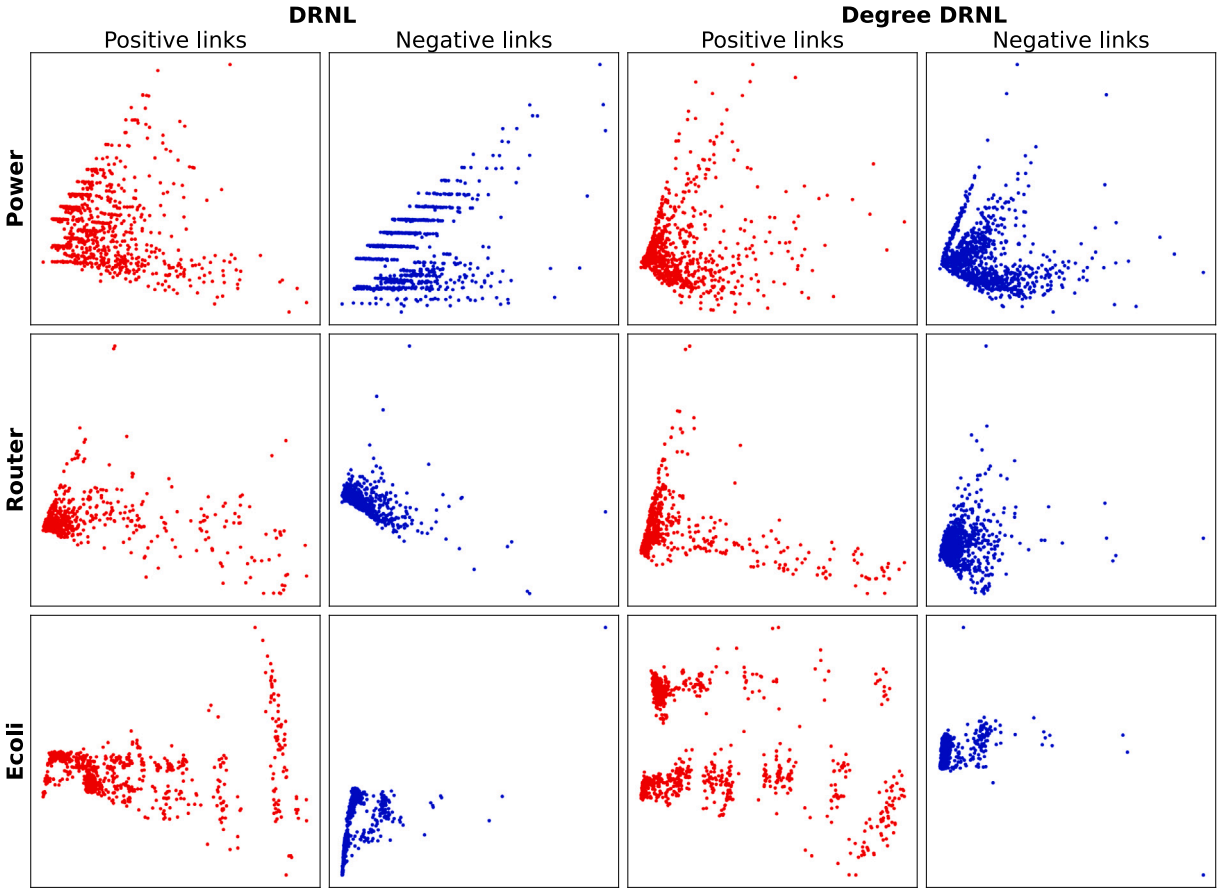

Fig. 13. Visualization of persistence image vectors calculated using MA-PHLP (dim 0) via PCA.

Data availability

Data will be made available on request.

References

[1] R. Ackland, et al., Mapping the US political blogosphere: are conservative bloggers more prominent? in: BlogTalk Downunder 2005 Conference, Sydney, BlogTalk Downunder 2005 Conference, Sydney, 2005.
[2] L.A. Adamic, E. Adar, Friends and neighbors on the web, Soc. Netw. 25 (2003) 211–230.
[3] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, L. Ziegelmeier, Persistence images: a stable vector representation of persistent homology, J. Mach. Learn. Res. 18 (2017).
[4] A.L. Barabási, R. Albert, Emergence of scaling in random networks, science 286 (1999) 509–512.
[5] V. Batagelj, A. Mrvar, Pajek datasets, 2006, http://vlado.fmf.uni-lj.si/pub/networks/data/.
[6] S. Bhatia, B. Chatterjee, D. Nathani, M. Kaul, A persistent homology perspective to the link prediction problem, in: International Conference on Complex Networks and Their Applications, Springer, 2019, pp. 27–39.
[7] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognit. 30 (1997) 1145–1159.
[8] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Netw. ISDN Syst. 30 (1998) 107–117.
[9] S. Brin, L. Page, Reprint of: The anatomy of a large-scale hypertextual web search engine, Comput. Netw. 56 (2012) 3825–3833.
[10] A. Brintrup, P. Wichmann, P. Woodall, D. McFarlane, E. Nicks, W. Krechel, Predicting hidden links in supply networks, Complexity 2018 (2018) 1–12.
[11] N. Brockmann, E. Elson Kosasih, A. Brintrup, Supply chain link prediction on uncertain knowledge graph, ACM SIGKDD Explor. Newsl. 24 (2022) 124–130.
[12] L. Cai, J. Li, J. Wang, S. Ji, Line graph neural networks for link prediction, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2022) 5103–5113.
[13] M. Carrière, F. Chazal, Y. Ike, T. Lacombe, M. Royer, Y. Umeda, Perslay: a neural network layer for persistence diagrams and new graph topological signatures, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2786–2796.
[14] B.P. Chamberlain, S. Shirobokov, E. Rossi, F. Frasca, T. Markovich, N.Y. Hammerla, M.M. Bronstein, M. Hansmire, Graph neural networks for link prediction with subgraph sketching, in: The Eleventh International Conference on Learning Representations, 2023, https://openreview.net/forum?id=m1oqEOAozQU.
[15] Y. Chen, B. Coskunuzer, Y. Gel, Topological relational learning on graphs, Adv. Neural Inf. Process. Syst. 34 (2021) 27029–27042.
[16] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Stability of persistence diagrams, in: Proceedings of the Twenty-First Annual Symposium on Computational Geometry, New York, NY, USA, Association for Computing Machinery, 2005, pp. 263–271, https://doi.org/10.1145/1064092.1064133
[17] T.K. Dey, Y. Wang, Computational Topology for Data Analysis, Cambridge University Press, 2022.
[18] K. Dong, Z. Guo, N.V. Chawla, Pure message passing can estimate common neighbor for link prediction, in: The Thirty-Eighth Annual Conference on Neural Information Processing Systems, 2024, https://openreview.net/forum?id=Xa3dVaolKo.
[19] Edelsbrunner, Letscher, Zomorodian, Topological persistence and simplification, Discret. Comput. Geom. 28 (2002) 511–533.
[20] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, Y. Elovici, Link prediction in social networks using computationally efficient topological features, in: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, IEEE, 2011, pp. 73–80.
[21] M. Gromov, Hyperbolic groups, in: Essays in Group Theory, Springer, 1987, pp. 75–263.
[22] A. Grover, J. Leskovec, Node2vec: scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.
[23] M. Horn, E. De Brouwer, M. Moor, Y. Moreau, B. Rieck, K. Borgwardt, Topological graph neural networks, arXiv preprint arXiv:2102.07835, 2021.
[24] S. Huber, Persistent homology in data science, in: Data Science–Analytics and Applications: Proceedings of the 3rd International Data Science Conference–iDSC2020, Springer, 2021, pp. 81–88.
[25] J. Immonen, A. Souza, V. Garg, Going beyond persistent homology using persistent homology, Adv. Neural Inf. Process. Syst. 36 (2024).
[26] G. Jeh, J. Widom, SimRank: a measure of structural-context similarity, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 538–543.
[27] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1953) 39–43.

[28] S.M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, Adv. Neural Inf. Process. Syst. 31 (2018).
[29] T.N. Kipf, M. Welling, Variational graph auto-encoders, arXiv preprint arXiv:1611.07308, 2016.
[30] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37.
[31] I.A. Kovács, K. Luck, K. Spirohn, Y. Wang, C. Pollis, S. Schlabach, W. Bian, D.K. Kim, N. Kishore, T. Hao, et al., Network-based prediction of protein interactions, Nat. Commun. 10 (2019) 1240.
[32] J.B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, Proc. Am. Math. Soc. 7 (1956) 48–50.
[33] C. Lei, J. Ruan, A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity, Bioinformatics 29 (2013) 355–364.
[34] H. Li, W. Jin, G. Skenderi, H. Shomer, W. Tang, W. Fan, J. Tang, Sub-graph based diffusion model for link prediction, in: The Third Learning on Graphs Conference, 2024, https://openreview.net/forum?id=RM2SAf5dd1.
[35] L. Lü, T. Zhou, Link prediction in complex networks: a survey, Phys. A 390 (2011) 1150–1170.
[36] C. Mavromatis, G. Karypis, Graph infoClust: Leveraging cluster-level node information for unsupervised graph representation learning, arXiv preprint arXiv:2009.06946, 2020.
[37] E. Nasiri, K. Berahmand, M. Rostami, M. Dabiri, A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding, Comput. Biol. Med. 137 (2021) 104772.
[38] M. Nayyeri, G.M. Cil, S. Vahdati, F. Osborne, A. Kravchenko, S. Angioni, A. Salatino, D.R. Recupero, E. Motta, J. Lehmann, Link prediction of weighted triples for knowledge graph completion within the scholarly domain, IEEE Access 9 (2021) 116002–116014.
[39] M.E. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (2006) 036104.
[40] M. Nickel, X. Jiang, V. Tresp, Reducing the rank in relational factorization models by including observable patterns, Adv. Neural Inf. Process. Syst. 27 (2014).
[41] M. Nie, D. Chen, D. Wang, H. Chen, Local optimization policy for link prediction via reinforcement learning, IEEE Trans. Netw. Sci. Eng. (2025).
[42] L. Pan, C. Shi, I. Dokmanić, Neural link prediction with walk pooling, in: International Conference on Learning Representations, 2022, https://openreview.net/forum?id=CCu6RcUMwK0.
[43] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
[44] L.F. Ribeiro, P.H. Saverese, D.R. Figueiredo, Struc2vec: learning node representations from structural identity, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 385–394.
[45] N. Shervashidze, P. Schweitzer, E.J. Van Leeuwen, K. Mehlhorn, K.M. Borgwardt, Weisfeiler-Lehman graph kernels, J. Mach. Learn. Res. 12 (2011).
[46] N. Spring, R. Mahajan, D. Wetherall, Measuring ISP topologies with Rocketfuel, ACM SIGCOMM Comput. Commun. Rev. 32 (2002) 133–145.
[47] Z. Stanfield, M. Coşkun, M. Koyutürk, Drug response prediction as a link prediction problem, Sci. Rep. 7 (2017) 40321.
[48] Student, The probable error of a mean, Biometrika (1908) 1–25.
[49] F.M. Taiwo, U. Islambekov, C.G. Akcora, Explaining the power of topological data analysis in graph machine learning, arXiv preprint arXiv:2401.04250, 2024.
[50] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1067–1077.
[51] L. Tang, H. Liu, Leveraging social media networks for classification, Data Min. Knowl. Discov. 23 (2011) 447–478.
[52] R.E. Tarjan, A class of algorithms which require nonlinear time to maintain disjoint sets, J. Comput. Syst. Sci. 18 (1979) 110–127.
[53] R.J. Tibshirani, B. Efron, An introduction to the bootstrap, Monogr. Stat. Appl. Probab. 57 (1993) 1–436.
[54] L. Vietoris, Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen, Math. Ann. 97 (1927) 454–472.
[55] C. Von Mering, R. Krause, B. Snel, M. Cornell, S.G. Oliver, S. Fields, P. Bork, Comparative assessment of large-scale data sets of protein–protein interactions, Nature 417 (2002) 399–403.
[56] Y. Wang, T. Zhao, Y. Zhao, Y. Liu, X. Cheng, N. Shah, T. Derr, A topological perspective on demystifying GNN-based link prediction performance, in: The Twelfth International Conference on Learning Representations, 2024, https://openreview.net/forum?id=apA6SSXx2e.
[57] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world'networks, nature 393 (1998) 440–442.
[58] T. Wen, E. Chen, Y. Chen, Tensor-view topological graph neural network, arXiv preprint arXiv:2401.12007, 2024.
[59] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (2020) 4–24.
[60] Z. Yan, T. Ma, L. Gao, Z. Tang, C. Chen, Link prediction with persistent homology: an interactive view, in: International Conference on Machine Learning, PMLR, 2021, pp. 11659–11669.
[61] L. Yao, L. Wang, L. Pan, K. Yao, Link prediction based on common-neighbors for dynamic social network, Proc. Comput. Sci. 83 (2016) 82–89.
[62] X. Ye, F. Sun, S. Xiang, TREPH: a plug-in topological layer for graph neural networks, Entropy 25 (2023) 331.
[63] C. Ying, X. Zhao, T. Yu, Boosting graph pooling with persistent homology, arXiv preprint arXiv:2402.16346, 2024.
[64] S. Yun, S. Kim, J. Lee, J. Kang, H.J. Kim, Neo-GNNs: neighborhood overlap-aware graph neural networks for link prediction, Adv. Neural Inf. Process. Syst. 34 (2021) 13683–13694.
[65] K. Zhang, J. Shen, G. He, Y. Sun, H. Ling, H. Zha, H. Li, J. Zhang, A transformative topological representation for link modeling, prediction and cross-domain network analysis, IEEE Trans. Pattern Anal. Mach. Intell. (2024).
[66] M. Zhang, Y. Chen, Weisfeiler-Lehman neural machine for link prediction, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 575–583.
[67] M. Zhang, Y. Chen, Link prediction based on graph neural networks, Adv. Neural Inf. Process. Syst. 31 (2018).
[68] M. Zhang, Z. Cui, S. Jiang, Y. Chen, Beyond link prediction: predicting hyperlinks in adjacency space, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
[69] Z. Zhang, P. Cui, W. Zhu, Deep learning on graphs: a survey, IEEE Trans. Knowl. Data Eng. 34 (2020) 249–270.
[70] Q. Zhao, Y. Wang, Learning metrics for persistence-based summaries and applications for graph classification, Adv. Neural Inf. Process. Syst. 32 (2019).
[71] Q. Zhao, Z. Ye, C. Chen, Y. Wang, Persistence enhanced graph neural network, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2896–2906.
[72] T. Zhou, L. Lü, Y.C. Zhang, Predicting missing links via local information, Eur. Phys. J. B 71 (2009) 623–630.

Author biography

Junwon You received the B.S. degree from the School of Undergraduate Studies, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Republic of Korea, in 2020. and the Ph.D. degree in the Department of Mathematics at Pohang University of Science and Technology (POSTECH), Republic of Korea, in 2025. He is working as post-doctoral researcher at the POSTECH Mathematical Institute for Data Science (POSTECH MINDS). His research interests include combining Topological Data Analysis and Deep Learning.

Eunwoo Heo received the B.S. degree from the Department of Mathematics Education, Pusan National University (PNU), Republic of Korea, in 2019, and the Ph.D. degree from the Department of Mathematics at Pohang University of Science and Technology (POSTECH), Republic of Korea, in 2025. He is currently a postdoctoral researcher at Ulsan National Institute of Science and Technology (UNIST) and a member of the LLM Innovation Research Center, Republic of Korea. His research interests include topological data analysis (TDA), multimodal representation learning, improving artificial intelligence models with TDA techniques, and theoretical research on the persistent homology (PH) of graph data.

Jae-Hun Jung obtained his Ph.D. degree in Applied Mathematics from Brown University. He has been a postdoctoral fellow at the University of British Columbia, an Assistant Professor of Mathematics at the University of Massachusetts Dartmouth and University at Buffalo, The State University of New York, an Associate (tenured) Professor of Mathematics at University at Buffalo SUNY and a Professor of Mathematics at Ajou University. He is currently a Professor of Mathematics at POSTECH and also serves as a founding director of POSTECH MINDS.