








Can Self Supervision Rejuvenate Similarity-Based Link Prediction?

Chenhan Zhang¹ , Weiqi Wang^(✉)² , Zhiyi Tian² , James Yu³ , Dali Kaafar¹ , An Liu⁴ , and Shui Yu² 

¹ Macquarie University, Sydney, Australia
{chzhang,dali.kaafar}@mq.edu.au

² University of Technology Sydney, Sydney, Australia
{weiqi.wang,zhiyi.tian,shui.yu}@uts.edu.au

³ University of York, York, UK
jqyu@ieee.org

⁴ Soochow University, Suzhou, China
anliu@suda.edu.cn

Abstract. Although recent advancements in end-to-end learning-based link prediction (LP) methods have shown remarkable capabilities, the significance of traditional similarity-based LP methods persists in unsupervised scenarios where there are no known link labels. However, the selection of node features for similarity computation in similarity-based LP can be challenging. Less informative node features can result in suboptimal LP performance. To address these challenges, we integrate self-supervised graph learning techniques into similarity-based LP and propose a novel method: **Self-Supervised Similarity-based LP (3SLP)**. 3SLP is suitable for the unsupervised condition of similarity-based LP without the assistance of known link labels. Specifically, 3SLP introduces a dual-view contrastive node representation learning (DCNRL) with crafted data augmentation and node representation learning. DCNRL is dedicated to developing more informative node representations, replacing the node attributes as inputs in the similarity-based LP backbone. Extensive experiments over benchmark datasets demonstrate the salient improvement of 3SLP, outperforming the baseline of traditional similarity-based LP by up to 21.2% (AUC).

Keywords: Link prediction · Graph neural networks · Self-supervised learning.

1 Introduction

Link prediction (LP) is one of the most intriguing and enduring challenges in the field of graph mining, contributing to a myriad of real-world applications such as friend recommendation in social networks, drug-drug interaction prediction, and knowledge graph completion [28]. LP methods can be broadly classified into two categories: similarity-based LP and learning-based LP. While learning-based LP has shown strong capabilities, there remains a significant role for similarity-based

LP methods. On the one hand, they heavily rely on known link information as labels to supervise the learning process [30,33]. However, in some realistic applications, link information can be absent, where we only have an *edgeless graph*. For example, in the “cold start” phase of the drug-drug interaction, where link labels may not be readily available [11]. Similarity-based LP assists in hypothesis generation by suggesting potential interactions derived from the attributes of drugs, where this label information can be further utilized in unsupervised learning on the data. On the other hand, the “black box” nature of widely adopted end-to-end neural network-based LP methods makes it difficult to gain insights into what inclusive modules or specific data features are pivotal in determining LP. Comparatively, similarity-based is a transparent and intuitive method that is more conducive to the analysis of results.

Research Gap. However, similarity-based LP methods are heuristic due to their predefined similarity measures [15]. Consequently, the quality of input node features can significantly impact the final LP performance. This study reexamines similarity-based LP and explores further advancement within the existing framework. We believe that the similarity-based LP methods can be enhanced by improving the representation of the original node attributes. Following this vein, a further hypothesis is that node-level features enriched with topological information can enhance prediction accuracy, considering that link existence depends on both individual node features and their relative positions and neighborhood structures. Nevertheless, as previously mentioned, similarity-based LP nowadays is often used in contexts when there are no or limited known links to label information. This condition impedes the utilization of topological information.

Motivation. The effectiveness of similarity-based link prediction (LP) methods largely depends on the quality of node features, particularly those enriched with topological information, as link existence is influenced by individual node attributes and their positions and neighborhood structures in the graph. Traditional LP methods rely on original or reduced-dimensional node attributes, which capture only individual information and are thus suboptimal [15]. Graph representation learning (GRL) offers a way to encode complex topological information into node features [12], but the lack of link-label information in similarity-based LP often limits its applicability. Self-supervised learning (SSL) has emerged as an effective approach for learning expressive representations from unlabeled data by employing data augmentation and pretext tasks [4,9]. Recent advancements in SSL have enhanced GRL techniques to capture latent topological information and improve node representations [24,20]. Thus, leveraging SSL offers a promising avenue to enrich node features for similarity-based LP.

While advanced end-to-end learning-based LP methods dominate the field [19,34,6], our study focuses on enhancing classic similarity-based LP methods due to their practicality in certain scenarios. Our approach is orthogonal to end-to-end methods and can complement them, for instance, by generating pseudo labels during cold-start phases. Rather than improving similarity metrics or algorithms (e.g., clustering), we aim to enhance input node features. The backbone used in

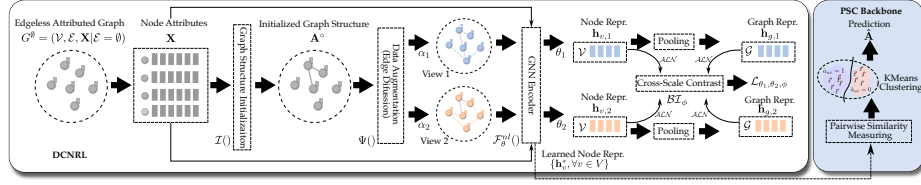


Fig. 1: Schematic of 3SLP.

our study, detailed in Section 2.3, builds on widely evaluated similarity-based LP methods [15,10,2].

Contribution. In this paper, we propose a novel approach to enhance similarity-based LP: **Self-Supervised Similarity-Based LP (3SLP)**. Leveraging self-supervised learning (SSL) techniques for unsupervised graph representation learning [20,24], 3SLP integrates SSL into node representation learning, using the learned representations for similarity-based LP. We adopt a widely used pairwise similarity-based clustering (PSC) backbone [10,31] as the testbed. This unsupervised backbone requires only node features as inputs, providing a practical scheme for similarity-based LP. To improve performance, we design dual-view contrastive node representation learning (DCNRL), which replaces original node attributes with informative representations. DCNRL addresses the lack of structural information by initializing a graph structure and employs graph diffusion to generate two augmented views during data augmentation. Inspired by [23,8], we apply cross-scale contrasting on node-level and graph-level representations to enhance node features. Experimental results show that 3SLP outperforms the baseline by up to 21.2%. Code is available at <https://anonymous.4open.science/r/3LSP-main-86BC/README.md>.

2 Our Method

2.1 Usage Scenario of Similarity-Based LP

The link prediction (LP) task determines whether a given pair of nodes are linked (i.e., an edge exists) [15]. Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed graph, where \mathcal{V} is the node set, \mathcal{E} is the edge set, and $\mathbf{X} = \mathbf{x}_i \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the node attribute matrix with d as the attribute dimension. The adjacency matrix $\mathbf{A} = a_{ij} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ has $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Let \mathcal{E}^k and \mathcal{E}^r denote the known and real edge sets, respectively. To replicate scenarios where only node attributes are accessible (e.g., early-stage drug-drug interactions), we consider edgeless attributed graphs, defined as $G^0 = (\mathcal{V}, \mathcal{E}^k, \mathbf{X} | \mathcal{E}^k = \emptyset)$. The objective of similarity-based LP in this context is $\mathbf{X} \rightarrow \mathcal{E}^r$. This setup presents a challenging test for LP methods, requiring them to extract subtle attribute differences for accurate predictions.

2.2 Method Overview

In this section, we introduce our proposed 3SLP. The core architecture of 3SLP is two-fold: (1) pairwise similarity-based clustering backbone (PSC) (empirical

backbone) and (2) dual-view contrastive node representation learning (DCNRL) (our design). Specifically, 3SLP first leverages DCNRL to develop informative node representations from the edgeless attributed graph. Then, the PSC backbone takes these node representations as input to predict the target links. A schematic view of 3SLP is given in Figure 1.

2.3 Pairwise Similarity-Based Clustering Backbone

We first introduce pairwise similarity-based clustering (PSC) based on the previous construction in the literature [10,31]. PSC forms a concise yet effective unsupervised clustering method, serving as the backbone of our method that implements similarity-based LP. The PSC backbone mainly incorporates two steps. The first step is node similarity measuring between pairwise input node features. For any two nodes u and v , and their respective features \mathbf{i}_u and \mathbf{i}_v , the node similarity measuring can be formulated as:

$$s_{uv} = \mathcal{T}(\mathbf{i}_u, \mathbf{i}_v), \quad (1)$$

where \mathcal{T} represents the adopted similarity measurer. In this work, we involve five commonly-used symmetric similarity metrics: (1) cosine similarity; (2) cosine distance; (3) Euclidean distance; (4) Manhattan distance; and (5) correlation distance. By node similarity measuring, we can obtain $|\mathcal{V}| \times |\mathcal{V}|$ pairs of node representation similarity score, denoted as \mathcal{S} . Next, PSC applies an unsupervised clustering algorithm to perform binary classification on the computed similarity scores. Specifically, we adopt the K-Means algorithm. The number of clusters is set to 2, corresponding to positive (link) and negative (no link) node pairs. Treating it as a ranking problem here, PSC calculates the average similarity score for each cluster—based on the adopted metrics, the cluster whose node pairs with a lower (higher) average score is considered linked (not linked)⁵. The clustering process can be written as:

$$\begin{aligned} \mathcal{S}_1^*, \mathcal{S}_2^* &\leftarrow \text{KMeans}(\{s_{uv} \in \mathcal{S}\}) \\ \begin{cases} \hat{a}_{uv} \in \mathcal{S}_1^* = 1, \hat{a}_{uv} \in \mathcal{S}_2^* = 0 & \text{if } \text{AVG}(\mathcal{S}_1^*) > \text{AVG}(\mathcal{S}_2^*) \\ \hat{a}_{uv} \in \mathcal{S}_1^* = 0, \hat{a}_{uv} \in \mathcal{S}_2^* = 1 & \text{else.} \end{cases} \end{aligned} \quad (2)$$

Given that the PSC is primarily used as the testbed for our new designs, further refinement, such as balancing positive and negative samples (class imbalance issue), falls outside the scope of this study. Building upon the PSC backbone, we can utilize the informative node feature to prompt LP. A naive approach directly takes the original node attribute information \mathbf{x} as input (PSC-NA). In this work, we treat PSC-NA as the major baseline.

⁵ An exception is when cosine similarity metric is adopted where we consider the cluster with higher value the linked one.

2.4 Dual-View Contrast Node Representation Learning (DCNRL)

Here, we propose DCNRL to generate informative node features for the PSC backbone using a dual-view contrastive learning framework. To address the specific needs of edgeless attributed graphs and similarity-based LP, DCNRL incorporates tailored data augmentation and contrastive learning modules. Details of these components are elaborated in the following sections.

Within the edgeless attributed graph, no graph structural (i.e., links) information is known. We cannot use any known graph structural information for data augmentation and further node representation learning. Therefore, we propose first to initialize a graph structure. Particularly, we adopt the cosine similarity metric to obtain the similarity between a node and all other nodes and select the nodes with top- k nearest nodes to connect with. This method resembles the creation of weak labels based on the attribute homophily [29]. Denote a graph initializer as \mathcal{I} , this step can be formulated as:

$$\mathbf{A}^\circ = \mathcal{I}(\mathbf{X}) \quad (3)$$

where \mathbf{A}° represents the adjacency matrix for the initialized graph structure.

However, the initialized graph structure may deviate significantly from the real graph structures, resulting in a noisy and insufficient configuration that may prove ineffectual for further GNN encoding. It has been shown that *diffusion* processes can enhance GNN performance on such noisy graph structures [14]. Therefore, we employ edge diffusion as a data augmentation strategy on the initialized graph structure. This approach can imitate higher-order proximities among nodes, thus empowering GNN encoders to capture more intricate graph properties within the SSL framework. Specifically, we employ the Personalized PageRank (PPR) method for edge diffusion. [1,16] have demonstrated PPR can smooth out the neighborhood over a noisy graph and restore more pertinent connections. Given the adjacency matrix of the initialized graph \mathbf{A}° and diffuser Ψ , we can develop edge diffused graph structures using a closed-form expression for PPR derived from [14], that is:

$$\tilde{\mathbf{A}}^\circ = \Psi(\mathbf{A}^\circ) := \alpha \left(\mathbf{I} - (1 - \alpha)\mathbf{D}^{-1/2}\mathbf{A}^\circ\mathbf{D}^{-1/2} \right)^{-1}, \quad (4)$$

where \mathbf{D} is the diagonal degree matrix of \mathbf{A}° and α is the teleportation probability. Moreover, we apply two different teleportation probabilities α_1 and α_2 to the diffuser to create two distinct views:

$$\begin{aligned} \text{VIEW1 : } \tilde{\mathbf{A}}_1^\circ &= \Psi_{\alpha_1}(\mathbf{A}^\circ) \\ \text{VIEW2 : } \tilde{\mathbf{A}}_2^\circ &= \Psi_{\alpha_2}(\mathbf{A}^\circ) \end{aligned} \quad (5)$$

These two views, characterized by varying diffusion levels, furnish a progressive topological transition from a high-order perspective. Consequently, the subsequent learning process is able to encode more comprehensive topological information within the node-level representation.

For the LP purpose, we aim to develop node-level representations infused with topological information; however, these representations may fall short of being

highly indicative due to the inherent limitations of the local aggregation function within a GNN encoder. Drawing inspiration from [8], we propose to apply a cross-scale strategy during the contrasting phase. By juxtaposing features at different scales, we can extract pattern and structural information across multiple dimensions [26]. This allows our GNN encoder to capture how the representation varies from different scales and thus learn a more robust representation. Specifically, we first employ two respective GNN encoders for two views. While sharing the same architecture, the two encoders do not share parameters (with θ_1 and θ_2 , respectively). Then, we use the GNN encoder to develop two scales for each view progressively, namely, node-level representation (by \mathcal{F}_θ^{nl}) and graph-level representation (by \mathcal{F}^{gl}), respectively:

$$\begin{aligned} \text{VIEW1} : & \begin{cases} \mathbf{H}_{v,1} = \mathcal{ALN} \left(\mathcal{F}_{\theta_1}^{nl} \left(\mathbf{X}, \tilde{\mathbf{A}}_1^\circ \right) \right) \\ \mathbf{H}_{g,1} = \mathcal{ALN} \left(\mathcal{F}^{gl} \left(\mathbf{H}_{v,1} \right) \right) \end{cases} \\ \text{VIEW2} : & \begin{cases} \mathbf{H}_{v,2} = \mathcal{ALN} \left(\mathcal{F}_{\theta_2}^{nl} \left(\mathbf{X}, \tilde{\mathbf{A}}_2^\circ \right) \right) \\ \mathbf{H}_{g,2} = \mathcal{ALN} \left(\mathcal{F}^{gl} \left(\mathbf{H}_{v,2} \right) \right), \end{cases} \end{aligned} \quad (6)$$

where \mathcal{ALN} represents the dimension alignment process to make each developed representation in the same dimension $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times h}$. The corresponding negative samples for each view are generated by our corruption function by randomly shuffling the node attributes $\mathcal{C} : \mathbf{X} \rightarrow \tilde{\mathbf{X}}$ and we use $\tilde{\mathbf{H}}_v = \mathcal{ALN} \left(\mathcal{F}_\theta^{nl} \left(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}_1^\circ \right) \right)$ to denote the representation developed by $\tilde{\mathbf{X}}$. Note that we do not choose to generate negative samples by corrupting the structural input $\tilde{\mathbf{A}}^\circ$, as their included augmented edges are not the actual instances and thus may not render effective contrasting.

To better maximize the concordance between two positive views while minimizing that between positive and negative views, we compute the mutual information between their corresponding node-level and graph-level representations. Inspired by [23], we adopt neural network-based mutual information estimation, making this step more tractable. In this way, the final contrastive learning objective can be expressed as:

$$\min_{\theta_1, \theta_2, \phi} \underbrace{\mathcal{BI}_\phi(\mathbf{H}_{g,1}, \mathbf{H}_{v,2})}_{\text{positive}}; \underbrace{\tilde{\mathbf{H}}_{g,1}, \tilde{\mathbf{H}}_{v,2}}_{\text{negative}} + \mathcal{BI}_\phi(\underbrace{\mathbf{H}_{g,2}, \mathbf{H}_{v,1}}_{\text{positive}}; \underbrace{\tilde{\mathbf{H}}_{g,2}, \tilde{\mathbf{H}}_{v,1}}_{\text{negative}}), \quad (7)$$

where \mathcal{BI}_ϕ represents a bilinear scoring function with shared parameters between the two views. The promise of the design is that we can obtain *globally relevant* node-level representations, capturing information across the entire graph. Such node representations are designed to preserve similarity for all node pairs, even for those that are distant [5, 23]. Subsequently, to obtain the final node representation that will be used as input to the PSC backbone, we compute the average of the learned node representations derived from the two encoders, which can be expressed as:

$$\{\mathbf{h}_v^*, \forall v \in V\} = \frac{1}{2} (\mathcal{F}_{\theta_1}^{nl}(\mathbf{X}, \tilde{\mathbf{A}}_1^\circ) + \mathcal{F}_{\theta_2}^{nl}(\mathbf{X}, \tilde{\mathbf{A}}_2^\circ)). \quad (8)$$

Method	Aux. Kn.	Metric	Cora			Citeseer			PubMed		
			AUC (%)	AP (%)	Δ	AUC (%)	AP (%)	Δ	AUC (%)	AP (%)	Δ
PSC-NA (Major baseline)	-	Cos sim.	61.3 \pm 0.1	61.3 \pm 0.1	-	64.6 \pm 0.1	64.7 \pm 0.1	-	79.7 \pm 0.1	75.6 \pm 0.1	-
		Cos dis.	61.3 \pm 0.1	61.3 \pm 0.1	-	64.6 \pm 0.1	64.6 \pm 0.1	-	79.7 \pm 0.1	75.4 \pm 0.1	-
		Eucl.	57.3 \pm 3.6	56.5 \pm 4.4	-	63.7 \pm 0.1	63.6 \pm 0.1	-	61.7 \pm 7.5	61.2 \pm 3.2	-
		Corr.	61.3 \pm 0.1	61.3 \pm 0.1	-	64.5 \pm 0.1	64.5 \pm 0.1	-	79.5 \pm 0.2	76.0 \pm 0.1	-
PSC-Repr (Zhang <i>et al.</i> [32])	✓	Cos sim.	63.3 \pm 1.1	66.5 \pm 2.4	-	64.7 \pm 3.8	58.7 \pm 2.7	-	50.2 \pm 0.1	50.0 \pm 0.1	-
		Cos dis.	66.3 \pm 5.2	67.9 \pm 1.4	-	60.0 \pm 1.0	55.5 \pm 0.7	-	50.2 \pm 0.1	50.0 \pm 0.1	-
		Eucl.	64.2 \pm 1.2	60.2 \pm 0.6	-	65.7 \pm 0.9	59.4 \pm 0.9	-	34.5 \pm 0.7	43.8 \pm 0.1	-
		Corr.	74.3 \pm 2.8	67.8 \pm 2.3	-	65.8 \pm 2.5	59.4 \pm 1.8	-	50.2 \pm 0.1	50.1 \pm 0.1	-
PSC-Pos (He <i>et al.</i> [10])	✓	Cos sim.	74.6 \pm 2.6	66.5 \pm 2.4	-	70.8 \pm 1.5	63.1 \pm 1.2	-	50.2 \pm 0.1	50.1 \pm 0.1	-
		Cos dis.	76.2 \pm 1.5	67.9 \pm 1.4	-	71.6 \pm 1.3	63.8 \pm 1.0	-	50.2 \pm 0.1	50.1 \pm 0.1	-
		Eucl.	66.6 \pm 0.4	60.2 \pm 0.5	-	66.7 \pm 0.2	60.1 \pm 0.1	-	34.0 \pm 0.5	43.8 \pm 1.6	-
		Corr.	76.0 \pm 2.5	67.8 \pm 2.3	-	72.1 \pm 2.8	64.1 \pm 2.4	-	50.2 \pm 0.1	50.1 \pm 0.1	-
3SLP (ours)	-	Cos sim.	77.1 \pm 1.2	70.9 \pm 1.7	17.0 \uparrow	84.0 \pm 1.6	76.4 \pm 1.7	19.4 \uparrow	79.4 \pm 1.5	73.3 \pm 1.1	0.3 \downarrow
		Cos dis.	78.7 \pm 0.2	71.3 \pm 0.1	17.4 \uparrow	85.8 \pm 0.9	81.7 \pm 1.2	21.2 \uparrow	80.5 \pm 2.0	76.3 \pm 2.5	0.8 \uparrow
		Eucl.	76.7 \pm 0.3	70.0 \pm 0.2	19.4 \uparrow	82.3 \pm 1.1	75.8 \pm 1.9	18.6 \uparrow	76.2 \pm 0.1	70.4 \pm 0.5	14.5 \uparrow
		Corr.	78.6 \pm 0.6	71.0 \pm 0.4	17.3 \uparrow	84.6 \pm 0.8	77.3 \pm 1.1	20.1 \uparrow	79.6 \pm 2.3	76.3 \pm 2.5	0.1 \uparrow

Table 1: Performance comparison. The best results are highlighted in **bold**. “Aux. Kn.” is short for “auxiliary knowledge”. Δ represents the performance improvements (AUC) from PSC-NA by 3SLP.

GNN Encoder	Cora	Δ	Citeseer	Δ
GCN (default)	73.9 \pm 2.6	12.6 \uparrow	85.8 \pm 0.9	21.2 \uparrow
GIN	71.3 \pm 1.4	10.0 \uparrow	82.6 \pm 1.2	18.0 \uparrow
SGC	74.1 \pm 0.5	12.8 \uparrow	74.4 \pm 5.7	9.8 \uparrow
GraphSAGE	67.0 \pm 4.9	5.7 \uparrow	63.4 \pm 8.0	1.2 \downarrow

Table 3: Comparison of LP performance (AUC (%)) among 3SLP configuring different GNN encoders. Δ represents the improvements measured from the baseline.

3 Evaluation

3.1 Experimental Setup

Datasets: We include four real-world and widely-adopted graph datasets, namely, Cora [21], Citeseer [21], PubMed [17], and Reddit [7] in our experiments.

Model and Parameter Settings. We adopt single-layer GCN [13] as the default GNN encoder, and the hidden size is set to 512. We use Adam as the optimizer for the SSL where we have the number of training epoch $T = 200$ and the learning rate $\eta = 0.001$. In the graph initialization, we set $k = 5$ for the top- k nearest algorithm. The teleportation probabilities for the two diffusers are set to $\alpha_1 = 0.2$ and $\alpha_1 = 0.4$. We consider cosine distance as the default similarity metric for PSC backbone. The optimal values for some of these hyperparameters are discussed later. Experiments for each setting are run repeatedly five times to eliminate randomness.

Metrics. We adopt area under the *ROC curve* (AUC), and *average precision* (AP) as our metrics, which is widely adopted to measure the performance of binary classification in a range of thresholds [3,32]. We primarily refer to AUC as did in [30,10]. Additionally, we introduce two metrics, Attribute Assortativity Coefficient (AAC) and Degree Assortativity Coefficient (DAC) [18], for the graph homophily analysis.



Fig. 2: Utility of predicted links on node classification tasks.

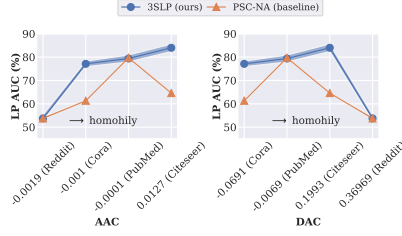


Fig. 3: Graph assortativity coefficients versus LP performance (using cosine similarity in PSC). AAC measures graph’s attribute homophily. DAC measures graph’s topology homophily.

3.2 Results

Comparison with Baselines. We first compare the LP performance of the proposed method 3SLP with the baselines. As aforementioned, we consider pairwise node attribute similarity-based clustering (PSC-NA) the major baseline. From the results shown in Table 1, we can observe that 3SLP pronouncedly surpasses the baseline in most cases. For example, within the same PSC metric, 3SLP can exceed PSC-NA by 10.2% (AUC) on the Cora dataset and 21.2% on the Citeseer dataset. Through self-supervision, 3SLP can develop node features that are more contributive than original node attributes with respect to the LP within the PSC backbone. However, we also notice that the performance improvement by our method on the PubMed dataset is not as pronounced as the other two datasets. We recognize that this pertains to graph homophily, which will be discussed later.

Additionally, we compare our method with the method involving auxiliary knowledge within the PSC backbone: (1) node representations (PSC-Repr) [32]: the node representations developed from the trained GNN on node classification tasks in a supervised manner; (2) node posteriors (PSC-Pos) [10]: the node prediction posteriors in node classification tasks developed from the trained GNN in a supervised manner. As shown in Table 1, 3SLP markedly outperforms these methods, implying the contribution of its developed node representations.

Source	LP AUC (%)	Target		
		Cora	Citeseer	PubMed
Cora	89.7 ± 0.9	51.0 ± 1.5	58.1 ± 6.0	
Citeseer	53.8 ± 6.5	88.0 ± 2.6	62.9 ± 4.4	
PubMed	57.0 ± 4.2	50.5 ± 6.3	93.2 ± 2.4	
Trans. Avg.	55.4 ± 5.4	50.8 ± 3.9	60.5 ± 5.2	
3SLP	73.9 ± 2.6	85.8 ± 0.9	80.5 ± 2.0	

Table 2: LP Performance comparison with the dataset knowledge transfer-based GAE. The diagonal (in shadow) shows the non-transfer results of GAE.

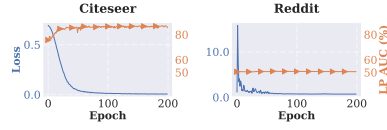


Fig. 4: Learning performance comparison between datasets Citeseer (homophilic) and Reddit (heterophilic). Training loss per epoch (blue) and validation performance per epoch (orange).

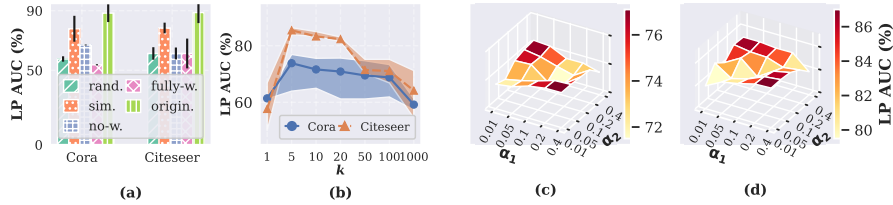


Fig. 5: Ablation and hyperparameter tests. (a): Influence of graph structure initialization. (b): Influence of k in similarity wiring. (c) and (d): Influence different teleportation probability α on Cora and Citeseer respectively.

Comparison with Knowledge Transfer-Based Methods. Given knowledge transfer is a useful technique for handling unsupervised tasks, we compare 3SLP with a parameter-sharing knowledge transfer method [22] using a Graph AutoEncoder (GAE) as the carrier. Each dataset serves as both source and target, yielding 9 combinations. When the source and target datasets are the same, it reduces to regular supervised learning (see Table 2, diagonal). Results in Table 2 show that 3SLP outperforms other methods on all three datasets, exceeding transferred GAE by an average of 19.1% on Cora and 28.6% on Citeseer. 3SLP even matches or surpasses non-transferred GAE in some cases. While transferred GAE’s effectiveness depends on the source dataset’s quality, sometimes leading to negative transfer, 3SLP leverages self-supervision to avoid performance degradation, ensuring robust learning.

Utility of Predicted Links in Cold-Start Tasks We explore whether the predicted links offer utility for further graph analysis, particularly in cold-start scenarios with limited information. Specifically, we evaluate if these links can train an accurate GNN classifier for node classification tasks. To do so, we train a GCN classifier using predicted links and node labels in a supervised manner and assess its performance on a test set, comparing it with other knowledge-based methods. As shown in Figure 2, our method achieves satisfactory results, particularly in cold-start scenarios. The classification accuracy derived from 3SLP’s predicted links consistently outperforms other methods. Notably, on Citeseer, the accuracy using the predicted graph structure surpasses that of the original graph structure, highlighting 3SLP’s ability to construct informative links for diverse applications, including cold-start assistance.

3.3 Analysis on Graph Homophily

When handling LP tasks, it is imperative to circumvent the homophilous assumption. Thereby, we investigate the influence of graph homophily on 3SLP’s performance. We introduce the AAC and the DAC, which measure the attribute homophily and topology homophily of the graph, respectively [18]. We include the relatively heterophilic dataset, Reddit, in this test to facilitate a more pronounced comparison. Figure 3 depicts a noticeable positive correlation between the LP performance of 3SLP and AAC: the higher the AAC, the better the LP performance of 3SLP; however, it does not show a similar trend between the

LP performance of 3SLP and DAC. We further compare 3SLP’s learning performance between the relatively homophilous dataset Citeseer and the heterophilic dataset Reddit (see Figure 4). We find that 3SLP fails to learn informative information for the LP task in the heterophilic dataset. These results imply a highly possible correlation between 3SLP’s effectiveness and attribute homophily assumption, which we will further explore in future work.

3.4 Hyperparameter and Ablation Study

Influence of Structural Initialization Methods. We evaluate the performance of 3SLP using four initialization methods for data augmentation, as shown in Figure 5(a). Similarity wiring outperforms others, as it initializes semantically meaningful graph structures based on node attributes. Next, we investigate the hyperparameter k in similarity wiring, setting $k \in 1, 5, 10, 20, 50, 100, 1000$ to assess LP performance. Note that $k = 0$ and $k = |\mathcal{V}|$ correspond to no wiring and fully wiring, respectively. As shown in Figure 5(b), the best performance occurs at $k = 5$, with a decreasing trend as k increases. These results highlight the importance of the initialization method and hyperparameter selection.

Influence of Edge Diffusion Levels. We evaluate the sensitivity of the proposed method to different edge diffusion levels by comparing the LP performance with different teleportation probabilities of PPR $\alpha_1, \alpha_2 \in \{0.01, 0.05, 0.1, 0.2, 0.4\}$. The results are shown in Figure 5(c) and (d). We can observe that the proposed method performs better when the difference between α_1 and α_2 is larger. This result suggests that when contrasting the two views, a more distinguished diffusion difference between the two views makes richer topological information can be captured, improving the quality of learned node representations.

Influence of GNN Encoders. Our design enables 3SLP to be model-agnostic regarding the GNN encoder. Beyond the default GCN, we extend our evaluation of 3SLP’s performance to three additional GNN encoders: SGC [25], GIN [27], and GraphSAGE [7]. As shown in Table 3, 3SLP exhibits consistently satisfying performance across different GNN encoders, demonstrating its compatibility. However, we notice that the performance with GraphSAGE is somewhat subpar, potentially due to the limitations inherent in its adopted neighborhood sampling strategy.

Influence of Similarity Metrics in PSC. Similarity metrics are pivotal in the PSC backbone, influencing the ultimate LP outcomes. In Table 1, we showcase the performance across all five introduced metrics for a comprehensive comparison. The results show that the best metric varies from dataset, except the Manhattan distance, which consistently underperforms compared to others. Furthermore, our repetitive tests also suggest that the cosine distance metric achieves more consistently good performance, making it our preferable configuration.

4 Conclusion

In this paper, we propose 3SLP to rejuvenate the similarity-based LP methods. We integrate self-supervised graph representation learning techniques into 3SLP

to enable it to handle the realistic use case of similarity-based LP. Without the supervision of edge labels, 3SLP can develop informative node representations as inputs instead of original node attributes to the pairwise similarity-based clustering backbone to prompt LP performance on attributed homophilic graphs. We empirically demonstrate the superiority of 3SLP compared to the baselines. Our extensive analysis highlights the crucial impact of graph homophily on the efficacy of 3SLP. In future research endeavors, we aim to explore the applicability of the proposed method to heterophilic graphs.

References

1. Berberidis, D., Giannakis, G.B.: Node embedding with adaptive similarities for scalable learning over graphs. *IEEE Transactions on Knowledge and Data Engineering* **33**(2), 637–650 (2019)
2. Chen, C., Liu, Y.Y.: A survey on hyperlink prediction. *IEEE Transactions on Neural Networks and Learning Systems* (2023)
3. Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., Zhang, Y.: When machine unlearning jeopardizes privacy. In: *Proc. ACM SIGSAC Conference on Computer and Communications Security*. pp. 896–911 (2021)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: *Proc. International conference on machine learning*. pp. 1597–1607. PMLR (2020)
5. Donnat, C., Zitnik, M., Hallac, D., Leskovec, J.: Learning structural node embeddings via diffusion wavelets. In: *Proc. ACM SIGKDD international conference on knowledge discovery & data mining*. pp. 1320–1329 (2018)
6. Guo, Z., Shiao, W., Zhang, S., Liu, Y., Chawla, N.V., Shah, N., Zhao, T.: Linkless link prediction via relational distillation. In: *International Conference on Machine Learning*. pp. 12012–12033. PMLR (2023)
7. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. *Proc. Advances in neural information processing systems* **30** (2017)
8. Hassani, K., Khasahmadi, A.H.: Contrastive multi-view representation learning on graphs. In: *Proc. International Conference on Machine Learning*. pp. 4116–4126 (2020)
9. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: *Proc. IEEE/CVF conference on computer vision and pattern recognition*. pp. 9729–9738 (2020)
10. He, X., Jia, J., Backes, M., Gong, N.Z., Zhang, Y.: Stealing links from graph neural networks. In: *30th USENIX Security Symposium (USENIX Security 21)*. pp. 2669–2686 (2021)
11. Huang, L., Lin, J., Liu, R., Zheng, Z., Meng, L., Chen, X., Li, X., Wong, K.C.: Coadti: multi-modal co-attention based framework for drug–target interaction annotation. *Briefings in Bioinformatics* **23**(6), bbac446 (2022)
12. Jiao, L., Chen, J., Liu, F., Yang, S., You, C., Liu, X., Li, L., Hou, B.: Graph representation learning meets computer vision: A survey. *IEEE Transactions on Artificial Intelligence* **4**(1), 2–22 (2022)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *Proc. International Conference on Learning Representations* (2017)
14. Klicpera, J., Welfenberger, S., Günnemann, S.: Diffusion improves graph learning. *arXiv preprint arXiv:1911.05485* (2019)

15. Kumar, A., Singh, S.S., Singh, K., Biswas, B.: Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications* **553**, 124289 (2020)
16. Li, P., Chien, I., Milenkovic, O.: Optimizing generalized pagerank methods for seed-expansion community detection. *Proc. Advances in Neural Information Processing Systems* **32** (2019)
17. Namata, G., London, B., Getoor, L., Huang, B., Edu, U.: Query-driven active surveying for collective classification. In: 10th International Workshop on Mining and Learning with Graphs. vol. 8, p. 1 (2012)
18. Newman, M.E.: Mixing patterns in networks. *Physical review E* **67**(2), 026126 (2003)
19. Pan, L., Shi, C., Dokmanić, I.: Neural link prediction with walk pooling. *arXiv preprint arXiv:2110.04375* (2021)
20. Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., Huang, J.: Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems* **33**, 12559–12571 (2020)
21. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. *AI magazine* **29**(3), 93–93 (2008)
22. Sun, X., Panda, R., Feris, R., Saenko, K.: Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems* **33**, 8728–8740 (2020)
23. Velickovic, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. In: *Proc. International Conference on Learning Representations* (2019)
24. Wang, S., Zeng, Z., Yang, X., Zhang, X.: Self-supervised graph learning for long-tailed cognitive diagnosis. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 37, pp. 110–118 (2023)
25. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: *Proc. International conference on machine learning*. pp. 6861–6871 (2019)
26. Wu, L., Lin, H., Tan, C., Gao, Z., Li, S.Z.: Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering* (2021)
27. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: *Proc. International Conference on Learning Representations* (2018)
28. Yang, C., Wang, C., Lu, Y., Gong, X., Shi, C., Wang, W., Zhang, X.: Few-shot link prediction in dynamic networks. In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. pp. 1245–1255 (2022)
29. Yang, L., Li, M., Liu, L., Wang, C., Cao, X., Guo, Y., et al.: Diverse message passing for attribute with heterophily. *Advances in Neural Information Processing Systems* **34**, 4751–4763 (2021)
30. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. *Advances in neural information processing systems* **31** (2018)
31. Zhang, Z., Liu, Q., Huang, Z., Wang, H., Lu, C., Liu, C., Chen, E.: Graphmi: Extracting private graph data from graph neural networks. In: *Proc. International Joint Conference on Artificial Intelligence*. pp. 3749–3755 (2021)
32. Zhang, Z., Chen, M., Backes, M., Shen, Y., Zhang, Y.: Inference attacks against graph neural networks. In: *Proc. USENIX Security* (2022)
33. Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D.: Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems* **33**, 7793–7804 (2020)

34. Zhu, Z., Zhang, Z., Xhonneux, L.P., Tang, J.: Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems* **34**, 29476–29490 (2021)