DGLP: INCORPORATING ORIENTATION INFORMATION FOR ENHANCED LINK PREDICTION IN DIRECTED GRAPHS

Yusen Zhang [1,*], *Yusong Tan* [1,*], *Songlei Jian* [1,✉], *Qingbo Wu* [1], *Kenli Li* [2]

[1] College of Computer Science and Technology, National University of Defense Technology, China
[2] College of Information Science and Engineering, Hunan University, China
{zhangys04, ystan, jiansonglei}@nudt.edu.cn, qingbo.wu@pcl.ac.cn, lkl@hnu.edu.cn

ABSTRACT

Link prediction in directed graphs offers a solution for uncovering detailed and accurate relationships among distinct entities. Unlike conventional link prediction in undirected graphs, the task becomes more intricate in directed graphs as it involves predicting both associations and orientations. Existing methods simply apply classic graph embedding techniques to learn node representations, followed by mapping representations of corresponding node pairs into probabilities indicating potential links. However, the inadequate capture of orientation information and sole reliance on node representations for prediction hinder the effective differentiation of orientation, thereby impeding the link prediction accuracy. In response, we introduce DGLP, an orientation-aware link prediction method tailored for directed graphs. DGLP utilizes the incidence matrix to learn both node and edge representations, effectively capturing structural and orientation information. By leveraging edge representations, DGLP achieves accurate link prediction in directed graphs without relying solely on implicit node representations. Experiments across six datasets demonstrate the effectiveness of DGLP, achieving a 1.2x improvement in prediction results.

***Index Terms*—** Link prediction, complex networks, edge representation learning, directed graph, graph neural network

1. INTRODUCTION

Link prediction in graphs is a fundamental task aiming to predict missing or potential edges between distinct nodes, enabling a comprehensive understanding of various complex systems [1]. In directed graphs, edges represent oriented relationships from source to target nodes, providing a detailed depiction of associations within diverse systems, including dialogue systems [2, 3], social networks [4, 5], recommendation systems [6, 7], and more. Accurately predicting oriented connections is essential for gaining insights into information propagation, influence pathways, and decision-making processes in complex systems [1]. As a result, link prediction in directed graphs has become a fascinating field of research.

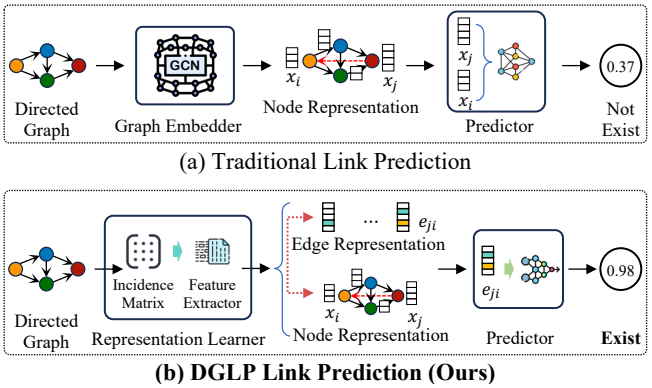

Fig. 1. Illustration of link prediction: (a) Traditional methods individually embed nodes and map node pair representations to probabilities for potential links. (b) DGLP leverages the incidence matrix to capture structural and orientation information concurrently, learning node and edge representations, and conducting link prediction based on edge representation.

However, the presence of asymmetric connections in directed graphs introduces complexities to link prediction. It necessitates identifying associations among distinct entities and the inherent orientation within these relationships [8]. This dual requirement adds complexity to the task, emphasizing the importance of capturing orientation information effectively. Existing link prediction methods, as depicted in Fig. 1(a), often follow a common scheme [1]. They individually embed nodes and use representations of corresponding node pairs to predict potential links. Nevertheless, this approach may have limitations when applied to directed graphs.

Most existing methods simply apply conventional graph embedding techniques to learn node representations, resulting in suboptimal link prediction results. Some of these methods rely on Laplacian smoothing, requiring the adjacency matrices of input graphs to be semi-definite [9, 10]. To fulfill this requirement, directed graphs are often preprocessed to render their adjacency matrices symmetric, inevitably discarding critical orientation information. Furthermore, certain methods aggregate features from neighboring nodes, exacerbating asymmetry within node representations [11, 12]. This inten-

[*] Equal contribution, ✉ Corresponding author

sification presents challenges for association prediction, as it relies on similar representations for connected nodes.

Moreover, existing methods also encounter challenges in effectively distinguishing orientations when mapping the representation of corresponding node pairs to probabilities indicating potential links. Some methods attempt to predict potential links using similarity measurements, but this approach is inherently symmetric and cannot discern orientation information [8]. Additionally, certain methods employ predictors that determine link existence between nodes based on their combined representations [13, 14]. Nonetheless, relying solely on node representations remains challenging for predicting connection orientation, as the permutations of node representations cannot fully distinguish orientations.

In response to these challenges, we introduce DGLP, a novel orientation-aware approach tailored for **D**irected **G**raph **L**ink **P**rediction. Obviously, edges in directed graphs convey not only associations but also specific orientations among nodes. Drawing from this insight, DGLP prioritizes the utilization of information from the directed edges themselves for link prediction. As illustrated in Fig. 1(b), DGLP leverages the incidence matrix to learn separate representations for nodes and edges, then conducts link prediction using edge representations. Benefiting from the capability of the incidence matrix to describe interactions between edges and nodes, DGLP could effectively capture associations and their corresponding orientations. Consequently, DGLP seamlessly integrates structural information and orientation details of relevant edges, resulting in enhanced prediction results.

In summary, our contributions are as follows: (i) We introduce an edge representation learning method utilizing the incidence matrix to fully extract topology and orientation information; (ii) We present an innovative orientation-aware approach that enables accurate link prediction in directed graphs, offering deeper insights into the intricate structure and dynamics of various networks. Extensive experiments over six datasets demonstrate the effectiveness of DGLP in directed graph link prediction, achieving a notable 1.2x improvement in prediction accuracy compared to other methods.

2. METHODOLOGY

2.1. Overview

The objective of DGLP is to unsupervisedly learn edge representations that comprehensively capture both structural and orientation information, enhancing prediction results. Inspired by Variational Autoencoders (VAE) [15], DGLP consists mainly of a *Graph Encoder* and a *Graph Decoder*, as illustrated in Fig. 2. Firstly, the input graph G is mapped into distributions $\mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\sigma}_v)$ and $\mathcal{N}(\boldsymbol{\mu}_e, \boldsymbol{\sigma}_e)$ in the latent space. Then, latent variables $\boldsymbol{Z}_v$ and $\boldsymbol{Z}_e$ are sampled from their respective distributions, and the incidence matrix is reconstructed by assessing the affinity between nodes and edges.

2.2. Directed Graph Encoding

Given a directed graph $G = (V, E)$ containing $m = |E|$ edges and $n = |V|$ nodes, the corresponding incidence matrix is denoted as $\boldsymbol{B}$, with rows and columns indexed by edges and nodes, respectively. The node feature matrix, denoted as $\mathbf{X} \in \mathbb{R}^{n \times d}$, is usually provided with the input graph. If not available, DGLP initializes it using the degrees of each node. On the other hand, common datasets often lack initial edge features $\mathbf{E}$, and DGLP initializes the edge features as below:

$$\mathbf{E} = \boldsymbol{B}\mathbf{X}\boldsymbol{\Theta} = \{\boldsymbol{\varepsilon}_0, \ldots, \boldsymbol{\varepsilon}_{m-1}\}, \tag{1}$$

where $\boldsymbol{\Theta}$ is the learnable weight matrix, and $\boldsymbol{\varepsilon}_i \in \mathbb{R}^d$ is the feature vector of edge e_i. Subsequently, subcomponents including the node feature encoder and the edge feature encoder are employed to map G into multivariate distributions in the latent space, guided by $\boldsymbol{B}$. This process is specified as:

$$\begin{aligned} q_{\boldsymbol{\theta}_v}(\boldsymbol{Z}_v | \mathbf{X}, \boldsymbol{B}) &= \prod_{i=0}^{n-1} q_{\boldsymbol{\theta}_v}\left(\boldsymbol{z}_v^{(i)} \middle| \mathbf{X}, \boldsymbol{B}\right), \\ q_{\boldsymbol{\theta}_e}(\boldsymbol{Z}_e | \mathbf{E}, \boldsymbol{B}) &= \prod_{i=0}^{m-1} q_{\boldsymbol{\theta}_e}\left(\boldsymbol{z}_e^{(i)} \middle| \mathbf{E}, \boldsymbol{B}\right), \end{aligned} \tag{2}$$

with

$$\begin{aligned} q_{\boldsymbol{\theta}_v}\left(\boldsymbol{z}_v^{(i)} \middle| \mathbf{X}, \boldsymbol{B}\right) &= \mathcal{N}\left(\boldsymbol{\mu}_v^{(i)}, \boldsymbol{\sigma}_v^{(i)}\right), \\ q_{\boldsymbol{\theta}_e}\left(\boldsymbol{z}_e^{(i)} \middle| \mathbf{E}, \boldsymbol{B}\right) &= \mathcal{N}\left(\boldsymbol{\mu}_e^{(i)}, \boldsymbol{\sigma}_e^{(i)}\right), \end{aligned} \tag{3}$$

where $\boldsymbol{Z}_e \in \mathbb{R}^{m \times d^*}$ refers to the edge representations, it can be obtained by sampling from the approximated posterior distribution $q_{\boldsymbol{\theta}_e}(\boldsymbol{Z}_e | \mathbf{E}, \boldsymbol{B})$. The dimension d^* is a hyperparameter and is set according to the specific task. And $q_{\boldsymbol{\theta}_e}(\cdot)$ is designed as a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_e$ and variance $\boldsymbol{\sigma}_e$, which can be formalized as:

$$\boldsymbol{\mu}_e = \mathcal{F}_{\mu_e}(\mathcal{H}_e(\mathbf{E}, \boldsymbol{B})), \ \boldsymbol{\sigma}_e = \mathcal{F}_{\sigma_e}(\mathcal{H}_e(\mathbf{E}, \boldsymbol{B})), \tag{4}$$

where $\mathcal{F}_{\mu_e}(\cdot)$ and $\mathcal{F}_{\sigma_e}(\cdot)$ are set to be MLP, although it is not limited to this. Meanwhile, $\mathcal{H}_e(\cdot)$ is a feature extraction function based on the incidence matrix and is defined as:

$$\mathcal{H}_e(\mathbf{E}, \boldsymbol{B}) = \widetilde{\boldsymbol{B}} \cdot \left(\widetilde{\boldsymbol{B}}^{\top} \cdot \mathbf{E}\boldsymbol{\mathcal{W}}_e\right)\boldsymbol{\mathcal{W}}_v, \tag{5}$$

where $\widetilde{\boldsymbol{B}} = \boldsymbol{D}_e^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{D}_v^{-\frac{1}{2}}$ is the normalized incidence matrix, $\boldsymbol{D}_v(k,k) = \sum_i \boldsymbol{B}(i,k)$ and $\boldsymbol{D}_e(k,k) = \sum_j \boldsymbol{B}(k,j)$ are degree matrices of nodes and edges, respectively. $\boldsymbol{\mathcal{W}}_e \in \mathbb{R}^{d \times d^*}$ and $\boldsymbol{\mathcal{W}}_v \in \mathbb{R}^{d^* \times d^*}$ are learnable parameters. All these functions together make up the edge feature encoder with parameter $\boldsymbol{\theta}_e$. The node feature encoder follows a similar approach.

2.3. Incidence Matrix Reconstruction

The latent variables $\boldsymbol{Z}_v$ and $\boldsymbol{Z}_e$ sampled from the distributions $q_{\boldsymbol{\theta}_v}(\boldsymbol{Z}_v | \mathbf{X}, \boldsymbol{B})$ and $q_{\boldsymbol{\theta}_e}(\boldsymbol{Z}_e | \mathbf{E}, \boldsymbol{B})$ are discrete and non-differentiable. To overcome this challenge, the Reparameterization [15, 16] technique is utilized, involving a continuous reparameterization step. This entails sampling a noise variable ϵ from an independent marginal distribution $p(\boldsymbol{\epsilon}) \sim$

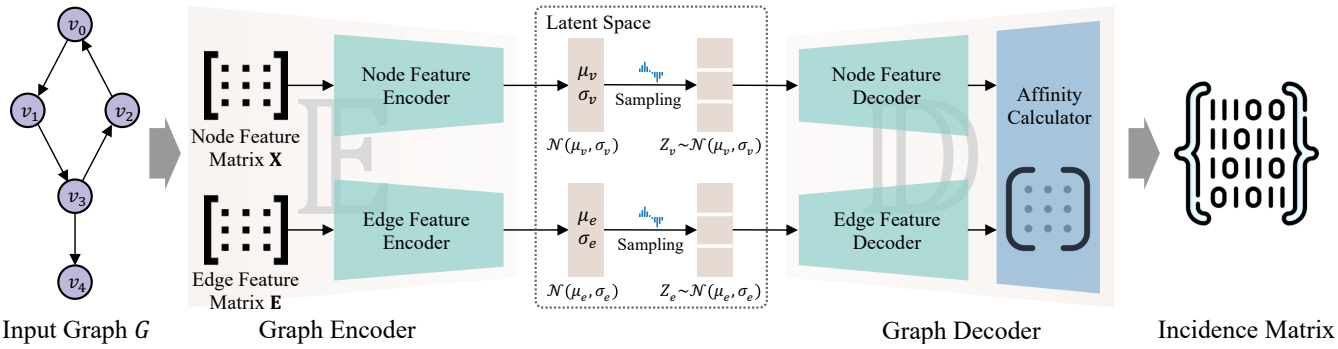

Fig. 2. The overview of the DGLP framework, including a Graph Encoder and a Graph Decoder. The input graph is mapped into posterior probability distributions $\mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\sigma}_v)$ and $\mathcal{N}(\boldsymbol{\mu}_e, \boldsymbol{\sigma}_e)$ in the latent space. Latent variables $\boldsymbol{Z}_v$ and $\boldsymbol{Z}_e$ are sampled accordingly. The incidence matrix of the input graph is then reconstructed using calculated affinities between nodes and edges.

$\mathcal{N}(0, I)$ and then calculating $\boldsymbol{Z}_e = \boldsymbol{\mu}_e + \boldsymbol{\sigma}_e^{1/2} \odot \boldsymbol{\epsilon}$. The Reparameterization technique is not the only solution, alternative methods such as Gumbel sampling [17] can also be employed.

Subsequently, DGLP reconstructs the incidence matrix of the input graph by calculating the affinity between edges and nodes individually. This process can be described as follows:

$$q_{\boldsymbol{\phi}}\left(\boldsymbol{B} \middle| \boldsymbol{Z}_e, \boldsymbol{Z}_v\right) = \prod_{i=0}^{m-1} \prod_{j=0}^{n-1} q_{\boldsymbol{\phi}}\left(B_{ij} \middle| \boldsymbol{Z}_e, \boldsymbol{Z}_v\right), \tag{6}$$

with

$$q_{\boldsymbol{\phi}}\left(B_{ij} | \boldsymbol{Z}_e, \boldsymbol{Z}_v\right) = \frac{\exp\left(\Gamma\left(\boldsymbol{w}^\top \left[\boldsymbol{W}\boldsymbol{z}_e^{(i)} \middle\| \boldsymbol{W}\boldsymbol{z}_v^{(j)}\right]\right)\right)}{\sum_{k=0}^{n-1} \exp\left(\Gamma\left(\boldsymbol{w}^\top \left[\boldsymbol{W}\boldsymbol{z}_e^{(i)} \middle\| \boldsymbol{W}\boldsymbol{z}_v^{(k)}\right]\right)\right)}, \tag{7}$$

where $[\cdot\|\cdot]$ denotes the concatenation operation. The learnable parameters $\boldsymbol{W} \in \mathbb{R}^{d' \times d^*}$ and $\boldsymbol{w} \in \mathbb{R}^{2d'}$ are used to model the affinity between edges and nodes, enabling capture both association and orientation information. $\Gamma(\cdot)$ is the LeakyReLU function used to introduce non-linearity. It is worth noting that $q_{\boldsymbol{\phi}}\left(B_{ij} \middle| \boldsymbol{Z}_e, \boldsymbol{Z}_v\right)$ is a modified attention drawing from GAT [18]. As a result, the incidence matrix can be reconstructed from node and edge representations.

By minimizing the reconstruction error between the reconstructed incidence matrix and the original one, the graph encoder and decoder can be trained effectively without any annotations. Referring to the ELBO of VAE, the unsupervised reconstruction objective of DGLP can be defined as:

$$\begin{aligned} &\arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}_{\text{re}} = \mathcal{D}\left[q_{\boldsymbol{\theta}_v}\left(\boldsymbol{Z}_v \middle| \mathbf{X}, \boldsymbol{B}\right) \| p(\boldsymbol{Z}_v)\right] \\ &+ \mathcal{D}\left[q_{\boldsymbol{\theta}_e}\left(\boldsymbol{Z}_e \middle| \mathbf{E}, \boldsymbol{B}\right) \| p(\boldsymbol{Z}_e)\right] - \mathbb{E}\left[\log q_{\boldsymbol{\phi}}\left(\boldsymbol{B} \middle| \boldsymbol{Z}_e, \boldsymbol{Z}_v\right)\right], \end{aligned} \tag{8}$$

where $\mathbb{E}[\cdot]$ stands for $\mathbb{E}_{q_{\boldsymbol{\theta}_e}\left(\boldsymbol{Z}_e | \mathbf{E}, \boldsymbol{B}\right), q_{\boldsymbol{\theta}_v}\left(\boldsymbol{Z}_v | \mathbf{X}, \boldsymbol{B}\right)}[\cdot]$, $\boldsymbol{\theta}_e, \boldsymbol{\theta}_v \in \boldsymbol{\theta}$, and $\mathcal{D}[\cdot]$ is the Kullback-Leibler divergence.

2.4. Link Prediction

The main objective of DGLP is to comprehensively capture the structural and orientation information inherent in directed graphs, thereby enhancing the precision of link prediction results. Once the graph encoder and decoder have been trained to converge, DGLP can generate effective representations for nodes and edges. Subsequently, the link prediction for edge $e_{ij} : v_i \mapsto v_j$ can be formalized as follows:

$$\boldsymbol{y}^* = \text{Softmax}(\mathcal{F}_e(\boldsymbol{Z}_{e_{ij}})), \tag{9}$$

where $\mathcal{F}_e(\cdot)$ is implemented as an MLP serving as the link predictor. This selection offers flexibility and can be tailored to the specific task. The output $\boldsymbol{y}^*$ is a probability vector indicating the presence or absence of the link from node v_i to v_j. Then the binary Cross-Entropy loss function is employed to train the predictor, formalized as:

$$\arg\min_{\omega} \mathcal{L}_{\text{bce}}(\boldsymbol{y}^*, \boldsymbol{y}), \tag{10}$$

where $\boldsymbol{y}$ represents the ground truth labels for directed links and ω denotes the parameter of the predictor.

3. EXPERIMENTS

3.1. Datasets and Settings

To demonstrate the effectiveness of DGLP, we conduct comparative experiments on six diverse directed graphs, meticulously grouped into two distinct categories. The first category encompasses citation networks, including Cora, Citeseer, and WikiCS. Meanwhile, the second category pertains to hyperlink networks among web pages associated with computer science departments at Cornell, Texas, and Wisconsin universities. Each of these graphs is endowed with node features, while edge features are initialized using Equation (1).

The process of edge representation learning is accomplished in an unsupervised manner by reconstructing the incidence matrix. The subsequent link predictor training process follows a semi-supervised strategy, with 20% of edges randomly selected as training samples. To address the challenges posed by imbalanced classification, we ensure an equal distribution of positive and negative samples during predictor training. Furthermore, we introduce synthetic edges by selectively removing existing edges and incorporating non-existent edges in a randomized manner.

3.2. Baseline Methods

We compare DGLP with ten baseline methods, which could be categorized into three groups: (i) Spectral-based methods

Table 1. Link prediction results for various methods on six datasets. ACC represents prediction accuracy, while F1 denotes the F1 score. Higher accuracy and F1 scores indicate superior prediction performance. The **NaN** value indicates non-convergence.

Methods	Cora		Citeseer		WikiCS		Cornell		Texas		Wisconsin	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
GCN	0.866±0.008	0.871±0.008	0.906±0.004	0.911±0.004	0.746±0.042	0.743±0.049	0.762±0.059	0.763±0.064	0.725±0.057	0.728±0.052	0.789±0.044	0.789±0.046
ChebNet	0.715±0.007	0.723±0.007	0.773±0.010	0.772±0.012	0.763±0.057	0.763±0.056	0.616±0.034	0.613±0.034	0.637±0.028	0.640±0.026	0.680±0.028	0.669±0.030
SAGE	0.848±0.026	0.854±0.025	0.912±0.008	0.912±0.008	0.658±0.037	0.658±0.037	0.848±0.026	0.851±0.026	0.873±0.016	0.872±0.016	0.856±0.035	0.860±0.035
APPNP	0.883±0.007	0.881±0.007	0.851±0.014	0.851±0.014	0.739±0.038	0.738±0.038	0.738±0.052	0.743±0.055	0.724±0.029	0.731±0.027	0.741±0.058	0.742±0.061
GIN	0.919±0.005	0.921±0.005	0.961±0.003	0.961±0.003	0.763±0.039	0.765±0.037	0.876±0.047	0.880±0.046	0.876±0.028	0.875±0.032	0.831±0.132	0.791±0.274
GAT	0.825±0.010	0.825±0.011	0.844±0.014	0.844±0.010	0.709±0.023	0.709±0.023	0.819±0.037	0.823±0.037	0.820±0.053	0.822±0.053	0.823±0.027	0.821±0.027
DGCN	0.844±0.011	0.844±0.012	0.860±0.005	0.862±0.005	NaN	NaN	0.818±0.032	0.819±0.033	0.823±0.044	0.823±0.042	0.828±0.041	0.827±0.041
DiGCN	0.797±0.105	0.753±0.263	0.713±0.042	0.713±0.057	NaN	NaN	0.606±0.022	0.593±0.025	0.616±0.057	0.630±0.046	0.640±0.046	0.621±0.055
DiGCN-IB	0.903±0.004	0.901±0.004	0.879±0.015	0.881±0.015	NaN	NaN	0.640±0.029	0.640±0.029	0.656±0.026	0.662±0.025	0.702±0.040	0.703±0.041
MagNet	0.920±0.006	0.922±0.006	0.957±0.002	0.962±0.002	0.812±0.030	0.806±0.038	0.628±0.044	0.921±0.006	0.572±0.027	0.587±0.060	0.516±0.018	0.421±0.212
DiGAE	0.598±0.002	0.602±0.002	0.673±0.001	0.619±0.005	0.638±0.001	0.683±0.002	0.679±0.001	0.609±0.002	0.688±0.003	0.615±0.002	0.620±0.002	0.578±0.001
DGLP-N	0.907±0.052	0.908±0.057	0.913±0.012	0.912±0.003	0.880±0.029	0.884±0.026	0.987±0.005	0.981±0.014	0.908±0.015	0.918±0.054	0.975±0.011	0.974±0.012
DGLP-M	0.904±0.009	0.907±0.009	0.926±0.002	0.927±0.002	0.902±0.003	0.903±0.004	0.991±0.003	0.988±0.002	0.991±0.001	0.987±0.001	0.990±0.004	0.990±0.005
DGLP-E	0.486±0.002	0.029±0.019	0.518±0.004	0.242±0.058	NaN	NaN	0.581±0.044	0.663±0.031	0.528±0.012	0.327±0.042	0.537±0.026	0.413±0.189
DGLP	**0.997**±0.004	**0.991**±0.004	**0.998**±0.003	**0.998**±0.003	**0.913**±0.004	**0.912**±0.004	**0.992**±0.013	**0.991**±0.011	**0.996**±0.006	**0.997**±0.005	**0.995**±0.010	**0.993**±0.011

that generate node representations through Laplacian smoothing, including GCN [9] and ChebNet [10]. (ii) Spatial-based methods learn node representations based on aggregation information from the neighborhood, including SAGE [19], APPNP [20], GIN [12], and GAT [18]. (iii) Directed GNN-based methods designed specifically for directed graphs that incorporate orientation information to produce node representation, including DGCN [21], DiGCN [14], MagNet [13], and DiGAE [8]. For link prediction tasks, we follow the common settings [8, 13] to carry out this task.

3.3. Link Prediction

We evaluate prediction results using accuracy and F1 scores, where higher accuracy and larger F1 scores indicate better performance. To ensure fair comparisons, consistent predictors are employed across all methods. The prediction results, as presented in Table 1, reveal that spatial-based and directed GNN-based methods outperform spectral-based ones. Spectral-based methods often preprocess adjacency matrices to be semi-definite for Laplacian smoothing, inevitably neglecting orientation information. These findings underscore the importance of incorporating orientation information into representations for link prediction in directed graphs.

Moreover, spatial-based methods excel on hyperlink networks, while directed GNN-based methods perform well on citation networks. This suggests that spatial-based methods may struggle with complex graphs, whereas directed GNN-based methods may not effectively capture orientation information on simple graphs. In contrast, DGLP consistently performs well across all directed graphs, highlighting its effectiveness in capturing both structural and orientation information for enhanced link prediction accuracy. Additionally, this also demonstrates the benefits of learning effective edge representations for link prediction in directed graphs.

3.4. Ablation Study

In order to further assess the capabilities of DGLP, we conducted an ablation study encompassing three variants: *DGLP-N*, *DGLP-M*, and *DGLP-E*. In the DGLP-N variant, link prediction is solely accomplished through node representations. Additionally, DGLP-M involves learning edge representations through MLPs without the use of the incidence matrix. Finally, DGLP-E randomly initializes edge features from random noises without using Equation (1). The results of the ablation study are present in the bottom part of Table 1.

Comparing the results of DGLP-E with those of DGLP-N and DGLP-M, we can conclude that learning effective edge representations is crucial for link prediction in directed graphs. Moreover, DGLP-M exhibits a slight performance improvement over DGLP-N, emphasizing the advantages of edge representation learning. Notably, DGLP consistently outperforms all variants, further reaffirming its effectiveness in learning edge representations to enhance prediction results.

4. CONCLUSION

This paper introduces DGLP, an orientation-aware link prediction method tailored for directed graphs. DGLP focuses on directly and comprehensively learning edge representations through the utilization of the incidence matrix. This approach enables the effective capture of both structural and orientation information, consequently enhancing link prediction accuracy. Through extensive experiments conducted on six widely used directed graphs, DGLP consistently outperforms various baseline methods.

5. ACKNOWLEDGMENT

This work is the results of the research project funded by the NSFC (U19A2060, 62002371), the Foundation of National University of Defense Technology (ZK21-17), and the Natural Science Youth Foundation of Hunan (2021JJ40678).

6. REFERENCES

[1] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas, "Link prediction techniques, applications, and performance: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 553, pp. 124289, 2020.

[2] Wei Peng, Yue Hu, Luxi Xing, Yuqiang Xie, Xingsheng Zhang, and Yajing Sun, "Modeling intention, emotion and external world in dialogue systems," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7042–7046.

[3] Djallel Bouneffouf, Mayank Agarwal, and Irina Rish, "Dialogue system with missing observation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[4] Guimu Guo, Da Yan, Lyuheng Yuan, Jalal Khalil, Cheng Long, Zhe Jiang, and Yang Zhou, "Maximal directed quasi-clique mining," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 1900–1913.

[5] Sandra Siby, Umar Iqbal, Steven Englehardt, Zubair Shafiq, and Carmela Troncoso, "Webgraph: Capturing advertising and tracking information flows for robust blocking," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2875–2892.

[6] Yuanxing Zhang, Pengyu Zhao, Yushuo Guan, Lin Chen, Kaigui Bian, Lingyang Song, Bin Cui, and Xiaoming Li, "Preference-aware mask for session-based recommendation with bidirectional transformer," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3412–3416.

[7] Chenyuan Feng, Zuozhu Liu, Shaowei Lin, and Tony QS Quek, "Attention-based graph convolutional network for recommendation system," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7560–7564.

[8] Georgios Kollias, Vasileios Kalantzis, Tsuyoshi Idé, Aurélie Lozano, and Naoki Abe, "Directed graph autoencoders," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 7211–7219.

[9] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR*, 2017.

[10] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[11] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang, "Neural bellman-ford networks: A general graph neural network framework for link prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29476–29490, 2021.

[12] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka, "How powerful are graph neural networks?," in *7th International Conference on Learning Representations, ICLR*, 2019.

[13] Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn, "Magnet: A neural network for directed graphs," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[14] Zekun Tong, Yuxuan Liang, Changsheng Sun, Xinke Li, David Rosenblum, and Andrew Lim, "Digraph inception convolutional networks," *Advances in neural information processing systems*, vol. 33, pp. 17907–17918, 2020.

[15] Diederik P. Kingma and Max Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR*, 2014.

[16] Carl Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[17] Iris AM Huijben, Wouter Kool, Max B Paulus, and Ruud JG Van Sloun, "A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1353–1371, 2022.

[18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, "Graph attention networks," *arXiv:1710.10903*, 2017.

[19] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[20] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[21] Zekun Tong, Yuxuan Liang, Changsheng Sun, David S Rosenblum, and A. Lim, "Directed graph convolutional network," *arXiv:2004.13970*, 2020.