

DiRW: Path-Aware Digraph Learning for Heterophily

Daohan Su
dhsu@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Xunkai Li
cs.xunkai.li@gmail.com
Beijing Institute of Technology
Beijing, China

Zhenjun Li
lizhenjun@szsit.edu.cn
Shenzhen Institute of Technology
Shenzhen, China

Yinping Liao
liaoyinping@szcp.edu.cn
Shenzhen City Polytechnic
Shenzhen, China

Rong-Hua Li
lironghuabit@126.com
Beijing Institute of Technology
Beijing, China

Guoren Wang
wanggrbit@gmail.com
Beijing Institute of Technology
Beijing, China

ABSTRACT

Recently, graph neural network (GNN) has emerged as a powerful representation learning tool for graph-structured data. However, most approaches are tailored for undirected graphs, neglecting the abundant information in the edges of directed graphs (digraphs). In fact, digraphs are widely applied in the real world and confirmed to address heterophily challenges. Despite recent advancements, existing spatial- and spectral-based DiGNNs have limitations due to their complex learning mechanisms and reliance on high-quality topology, resulting in low efficiency and unstable performance. To address these issues, we propose Directed Random Walk (DiRW), a plug-and-play strategy for most spatial-based DiGNNs and also an innovative model which offers a new digraph learning paradigm. Specifically, it utilizes a direction-aware path sampler optimized from the perspectives of walk probability, length, and number in a weight-free manner by considering node profiles and topologies. Building upon this, DiRW incorporates a node-wise learnable path aggregator for generalized node representations. Extensive experiments on 9 datasets demonstrate that DiRW: ① enhances most spatial-based methods as a plug-and-play strategy; ② achieves SOTA performance as a new digraph learning paradigm. The source code and data are available at <https://github.com/dhsuu/DiRW>.

CCS CONCEPTS

• **Computing methodologies** → **Semi-supervised learning settings**; **Neural networks**.

KEYWORDS

Graph Neural Network; Directed Graph; Random Walk

ACM Reference Format:

Daohan Su, Xunkai Li, Zhenjun Li, Yinping Liao, Rong-Hua Li, and Guoren Wang. 2025. DiRW: Path-Aware Digraph Learning for Heterophily. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*, November 10–14, 2025, Seoul, Republic of Korea. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3746252.3761090>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '25, November 10–14, 2025, Seoul, Republic of Korea.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2040-6/2025/11...\$15.00
<https://doi.org/10.1145/3746252.3761090>

1 INTRODUCTION

Graph neural networks (GNNs) have been widely used across node- [16, 35, 37], link- [15, 30, 44], and graph-level tasks [17, 20, 31] and achieve satisfactory performance. Therefore, this graph-based deep learning technique holds great potential for applications, such as recommendation [27, 41], financial analysis [4, 10, 25], and health-care [2, 23]. Despite their effectiveness, existing methods often overlook edge direction in natural graphs, leading to inevitable information loss and limited performance upper bound.

Compared to undirected representations, digraphs are crucial for modeling complex real-world topologies (e.g., web flow monitoring and bioinformatics), capturing more intricate node relationships. Additionally, the recently proposed A2DUG [22], Dir-GNN [26] and ADPA [28] reveal a key insight: *Edge direction offers a new perspective for addressing the topological heterophily challenges that plague graph learning*. Despite growing attention, DiGNNs are still in their infancy and face the following inherent limitations: ① Spatial-based methods often stack multiple convolution layers with separate learnable parameters for out-edges and in-edges, resulting in over-smoothing concerns [40] and high computational costs [8, 13, 47]. ② Spectral-based methods heavily rely on high-quality directed topology [32, 45], and without this, extreme eigenvalues inevitably lead to sub-optimal performance. Strict theoretical assumptions also limit practical deployment in complex scenarios. Therefore, developing a more efficient paradigm for digraph learning is urgent.

To enhance the usability of DiGNNs, this paper focuses on spatial-based methods and proposes a novel (directed) path-based learning mechanism (Di)PathGNNs. As we all know, the entanglement of homophily and heterophily, where connected nodes exhibit intricate feature distributions and labels, has recently posed a significant challenge [19, 21, 24, 46]. Researchers strive to achieve robust learning within this complex topology. In this context, compared to the traditional message-passing (i.e., neighbor expansion with strict spatial symmetry that disregards edge direction), we highlight the following advantages of DiRW to further clarify the motivation of our study: ① **Edge Direction and Node Order**. The core of DiPathGNNs lies in performing well-designed random walks for each node, treating the paths as node-wise sequences. *Advantage: DiRW fully considers edge direction and preserves the order of nodes within the walking paths, which is crucial for capturing structural insights.* ② **Adaptive Expansion of Node Receptive Fields**. DiPathGNNs adaptively extend hop-based neighbors to path-based neighbors by considering the characteristics of random walk, incorporating more homophilous signals. *Advantage: DiRW ensures*

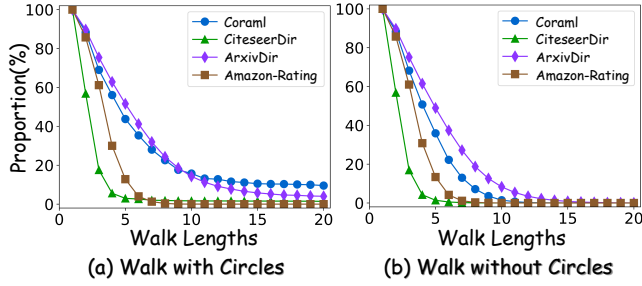


Figure 1: Interruption issue of DiSRW on digraphs.

message aggregation among nodes with the same label, equivalent to data augmentation and highlighting label-specific positive signals for prediction. **② Path-based Message Aggregation.** Based on the above homophily-aware paths, DiPathGNNs aggregate these messages for predicting each node through a learnable mechanism. *Advantage: DiRW facilitates the dense aggregation of direction-aware homophilous information, thus enhancing node prediction.*

Despite recent advancements in PathGNNs [11, 29, 38], their walk strategies are tailored for undirected graphs, lacking generalizability. Furthermore, the complex relationships inherent in directed topology pose unique challenges to the naive random walks, necessitating further investigations to develop fine-grained walk rules. To further elucidate, we present empirical analyses in Fig. 1-2 that demonstrate these limitations and offer key insights.

◆ **Limitation 1: Neglect of edge directions.** PathGNNs often fail to account for the edge direction. Utilizing directed simple random walks (DiSRW) on digraphs inevitably leads to interruptions when encountering nodes lack outgoing edges. To investigate and visualize this limitation, we employ DiSRW and initiate a walk from each node on four digraphs. As the walk length increases, we record the proportion of complete paths out of the total sequences, as depicted in Fig. 1(a). To mitigate the impact of cycles within the digraph, we also devise an DiSRW that excludes cycles and conduct the same experiment, as shown in Fig. 1(b).

✧ **Key Insight 1:** *Walking strictly along the edge directions leads to severe walk interruption problems.* The non-strongly connected nature of digraphs causes most walks to show a sharp decline in complete paths after just five steps, indicating that they fail to gather extensive information beyond the immediate neighborhood of the starting node. Additionally, removing the influence of cycles results in an even greater decline in uninterrupted sequences.

◆ **Limitation 2: Coarse-grained walk strategies.** PathGNNs treat walk number and length as hyperparameters, uniformly applying them across all nodes. This one-size-fits-all approach overlooks the distinct contexts of nodes within complex topology. To further investigate this, we conducted an in-depth analysis of the effects of varying walk numbers (Fig. 2(a)) and walk lengths (Fig. 2(b)) on node classification. We utilize undirected SRW as the sampling strategy, and utilize two MLPs to generate path embeddings from the nodes along the sampled paths and aggregate node embeddings from multiple paths associated with the same node. Node homophily is calculated as the ratio of the number of first-order neighboring nodes sharing the same label to the total number of first-order neighbors, with the top 50% of nodes by homophily scores classified as homophilous nodes and the remains as heterophilous nodes.

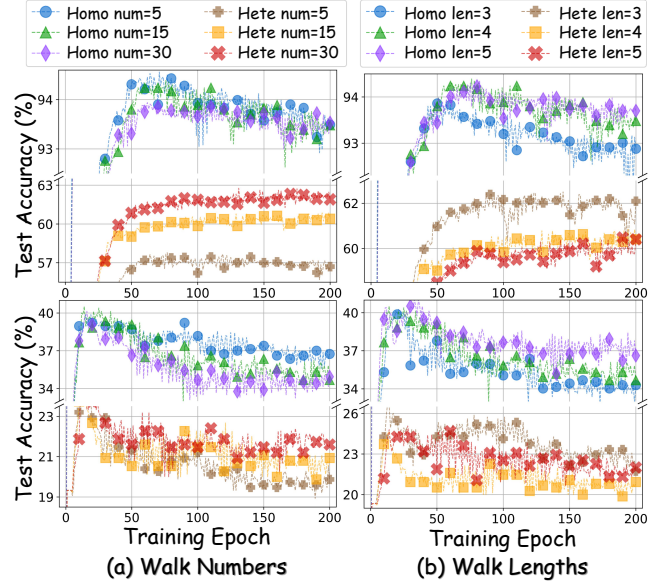


Figure 2: Performance of SRW on digraphs WikiCS (Homophilous, Upper) and Actor (Heterophilous, Lower).

✧ **Key Insight 2:** *Higher walk numbers facilitate heterophilous nodes, while longer walk lengths benefit homophilous nodes.* Increasing walk numbers enhances predictions for heterophilous nodes by enabling a comprehensive exploration of their complex topologies. In contrast, homophilous nodes can be accurately represented with fewer walks due to their uniform neighborhood structure. Excessive walks in such case raise concerns regarding over-smoothing and time and space complexity. Regarding walk length, heterophilous nodes are often surrounded by noisy environments, where longer walks risk incorporating irrelevant information. Conversely, homophilous nodes benefit from longer walks as they gather a broader range of relevant information, offering a clearer surrounding views.

Inspired by the above key insights, we propose Directed Random Walk (DiRW), which can be viewed as a plug-and-play directed walk strategy for PathGNNs or a new digraph learning paradigm. Specifically, DiRW first designs a direction-aware walk strategy to identify potential neighbor relationships (motivated by **Key Insight 1**). It then fine-tunes walk numbers and lengths on a node-adaptive basis (motivated by **Key Insight 2**). Finally, DiRW employs a node-wise learnable path aggregator to represent node embeddings, breaking the constraints of message-passing mechanisms.

Our contributions. ① *Novel Perspective.* This paper is the first to introduce DiPathGNN and highlight its advantages, emphasizing the need for fine-grained walking strategies in digraphs through valuable empirical studies. ② *Plug-and-play Strategy.* We present DiRW, which customizes walk probabilities, lengths, and numbers for each node, seamlessly integrating with any DiPathGNN to enhance performance. ③ *New DiGNN.* DiRW also serves as a new learning architecture for digraphs, featuring a pre-processed optimized path sampler and a well-designed node-wise learnable path aggregator. ④ *SOTA Performance.* Experiments demonstrate that DiRW improves node classification by 2.81% as a plug-and-play strategy and enhances link prediction by 0.82% as a novel DiGNN.

2 PRELIMINARIES

2.1 Notation and Problem Formulation

We consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ nodes and $|\mathcal{E}| = m$ edges, where each node has a feature of size f and a one-hot label of size c . The feature and label matrices represented as $\mathbf{X} \in \mathbb{R}^{n \times f}$ and $\mathbf{Y} \in \mathbb{R}^{n \times c}$. The \mathcal{G} is described by an asymmetrical adjacency matrix \mathbf{A} , where $\mathbf{A}(u, v) = 1$ if $(u, v) \in \mathcal{E}$, and 0 otherwise. The goal of the semi-supervised node classification is to predict the labels for unlabeled nodes with the supervision of labeled nodes.

2.2 Directed Graph Neural Networks

2.2.1 Spatial-based Methods. To capture the asymmetric topology of digraphs, some spatial-based methods follow the strict symmetric message-passing paradigm in the undirected setting [5, 9, 39]. However, it is crucial to account for the directionality of the edges when aggregating messages. Specifically, for the current node $u \in \mathcal{V}$, the model learns weights to combine the representations of its out-neighbors ($u \rightarrow v$) and in-neighbors ($v \rightarrow u$) independently:

$$\begin{aligned} \mathbf{H}_{u \rightarrow}^{(l)} &= \text{Agg} \left(\mathbf{W}_{\rightarrow}^{(l)}, \text{Prop} \left(\mathbf{X}_u^{(l-1)}, \left\{ \mathbf{X}_v^{(l-1)}, \forall (u, v) \in \mathcal{E} \right\} \right) \right), \\ \mathbf{H}_{u \leftarrow}^{(l)} &= \text{Agg} \left(\mathbf{W}_{\leftarrow}^{(l)}, \text{Prop} \left(\mathbf{X}_u^{(l-1)}, \left\{ \mathbf{X}_v^{(l-1)}, \forall (v, u) \in \mathcal{E} \right\} \right) \right), \\ \mathbf{X}_u^{(l)} &= \text{Com} \left(\mathbf{W}^{(l)}, \mathbf{X}_u^{(l-1)}, \mathbf{H}_{u \rightarrow}^{(l)}, \mathbf{H}_{u \leftarrow}^{(l)} \right), \end{aligned} \quad (1)$$

where the Propagation function $\text{Prop}(\cdot)$ gathers and distributes feature from neighbors, and the Aggregation function $\text{Agg}(\cdot)$ combines the feature depending on the learnable parameter matrix $\mathbf{W}^{(l)}$. Building upon this, recent advances in DiGNNs further refine the message-passing scheme to capture the inherent directionality of the digraph. DGCN [33] incorporates both first- and second-order neighbor proximity into the message aggregation process. DIMPA [8] expands the receptive field by aggregating features from K -hop neighborhoods. Inspired by the 1-WL graph isomorphism test, NSTE [13] tailors the message propagation to the directed nature of the graph. DiGCN [32] leverages neighbor proximity to increase the receptive field and proposes a digraph Laplacian based on personalized PageRank. ADPA [28] adaptively explores directed patterns to conduct weight-free propagation and employs two hierarchical node-wise attention mechanisms to learn representations.

2.2.2 Spectral-based Methods. Spectral-based approaches for the DiGNN depart from the strict symmetric message-passing used for undirected graphs [7, 36]. The core of spectral-based DiGNNs is to leverage a symmetric or conjugated digraph Laplacian \mathbf{L}_d , which is constructed based on the directed adjacency matrix \mathbf{A}_d . This symmetric Laplacian \mathbf{L}_d allows the application of spectral convolution operations, which can be formally represented as a function of the eigenvalues and eigenvectors of \mathbf{L}_d . Specifically, the layer-wise node embeddings $\mathbf{X}^{(l)}$ are computed via a first-order approximation of Chebyshev polynomials, leveraging the spectral decomposition of the symmetric digraph Laplacian.

$$\begin{aligned} \mathbf{L}_d &= \text{DGS}(\mathbf{A}_d, \alpha, q), \\ \mathbf{X}^{(l+1)} &= \text{Poly}(\mathbf{L}_d) \text{MLP}(\mathbf{X}^{(l)}), \end{aligned} \quad (2)$$

where $\text{DGS}(\cdot)$ is the digraph generalized symmetric function with parameters and $\text{Poly}(\cdot)$ is a polynomial-based approximation method.

Building on this, DiGCN [32] proposes a α -parameterized stable state distribution based on the personalized PageRank to achieve the digraph convolution. MagNet [45] utilizes q -parameterized complex Hermitian matrix to model directed information in digraphs. MGC [43] adopts a truncated variant of PageRank, designing low-pass and high-pass filters tailored for homogeneous and heterogeneous digraphs. LightDiC [15] decouples graph propagation and feature aggregation for scalability in large-scale scenarios.

2.3 Path-based Graph Neural Networks

PathGNNs offer an effective approach to capture the intricate graph patterns by sampling and aggregating information along paths. For instance, GeniePath [18] introduces an adaptive path layer that navigates the exploration of both the breadth and depth of the node's receptive fields. SPAGAN [42] leverages the shortest paths and applies path-based attention mechanisms to obtain node embeddings. PathNet [29] utilizes a maximal entropy-based random walk strategy to capture the heterophilous and structural information. RAWGNN [11] employs Node2Vec [6] to simulate both BFS and DFS, capturing both homophily and heterophily information. PathMLP [38] designs a similarity-based path sampling strategy to capture smooth paths containing high-order homophily.

3 METHOD

Building upon the key insights discussed in Sec. 1, we now present our DiRW model, which is composed of two principal components: the optimized path sampler and the node-wise learnable path aggregator. The architecture of the model is depicted in Fig. 3. Specifically, drawing upon **Key Insight 1**, DiRW is initiated with a direction-aware path sampling strategy in Sec. 3.1.1. Furthermore, we introduce a multi-order walk probability scheme in Sec. 3.1.2. Guided by **Key Insight 2**, we evaluate the quality of walk sequences by introducing the homophily entropy to customize the walk length in Sec. 3.1.3. In Sec. 3.1.4, we propose a weight-free aggregation mechanism for the path embedding. We also tailor walk numbers in Sec. 3.1.5 by assessing the information richness of sampled walks. In the learning phase, we integrate an attention mechanism to learn the importance of each walk sequence in Sec. 3.2.1. Culminating in Sec. 3.2.2, we utilize a linear layer to distill the final node representations, applied to node classification and link prediction.

3.1 Optimized Path Sampler

3.1.1 Direction-aware Path Sampler. In DiRW, edge directionality governs the walk trajectory through the direction-aware sampling protocol that dynamically adapts to node connectivity patterns. This critical stage is tasked with determining the destination set for each walk, which is essential for calculating the node probabilities and ensuring the continuity of the walk without interruptions.

Specifically, when the walk encounters a sink node (node with empty in-neighborhoods), it strategically transitions to its out-neighbors. Conversely, when encountering a source node (node lacking out-edges), the walk remains confined to its in-neighborhood. This approach effectively circumvents the issue of walk termination at nodes with unidirectional connectivity, which is mentioned in **Key Insight 1**. For nodes that possess both incoming and outgoing edges, a more nuanced strategy is needed to balance the

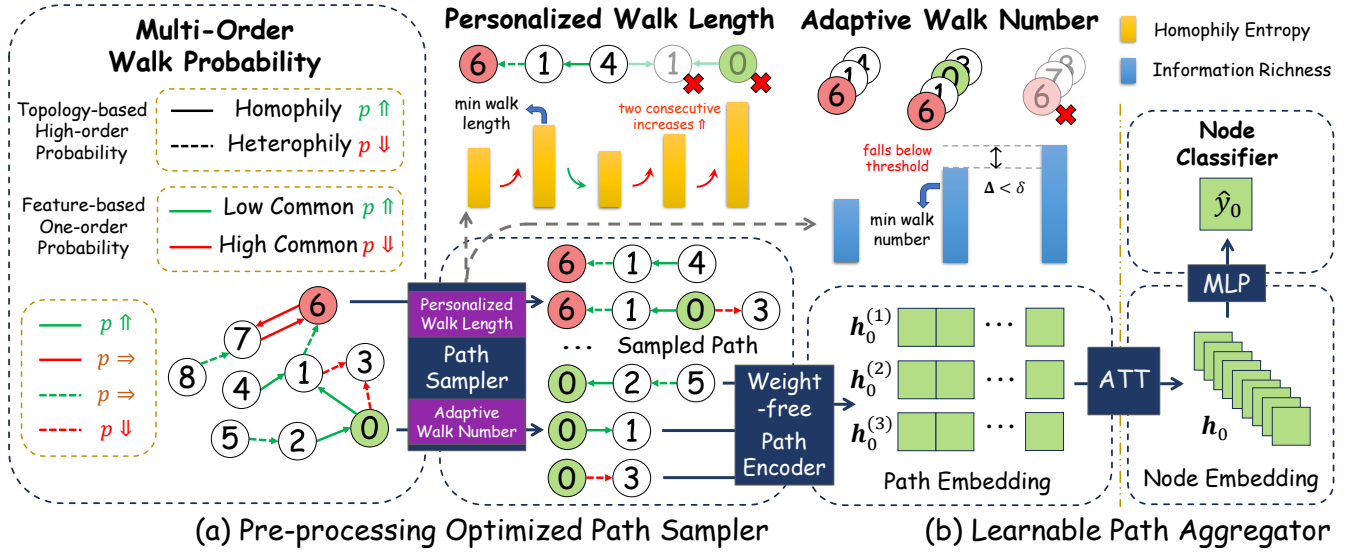


Figure 3: Overview of our DiRW, including (a) sampling optimized sequences and calculating the path embeddings in a weight-free manner; (b) learning the importance of different paths to obtain accurate node predictions by node-wise path aggregator.

edge presence and directionality. Drawing inspiration from the magnetic Laplacian [45], DiRW introduces a direction control parameter $q \in [0, 1]$ with a uniformly sampled threshold $r \in [0, 1]$ to determine the successor selection. If $r > q$, the walk proceeds directionally to out-neighbors, whereas $r \leq q$ permits bidirectional exploration considering both the in-neighbors and out-neighbors. This mechanism interpolates between strict directionality preservation ($q = 0$) and undirected-like exploration ($q = 1$), while preventing walk termination at unidirectionally connected nodes.

3.1.2 Multi-order Walk Probability. The computation of walk probabilities in DiRW is a nuanced process that seamlessly integrates both the topological structure and node profiles, enabling DiRW to delve into the complexities of both high-order and one-order homophily within heterophilous graphs.

Topology-based High-order Probability. DiRW uncovers high-order homophily information which is often embedded within heterophilous graphs [38] by strategically prioritizing nodes with few common neighbors, thus bridging to distant areas of the graph. The topology-based high-order probability P_u^{topo} is inversely proportional to the number of common neighbors $\text{Com}(u, v)$ and normalized by the degree $\text{Deg}(v)$ of the candidate node v :

$$P_u^{topo}(v) = 1 - \frac{\text{Com}(u, v) + 1}{\text{Deg}(v)}, \text{ if } (u, v) \in \mathcal{E}. \quad (3)$$

Feature-based One-order Probability. From the feature standpoint, DiRW captures one-order homophily by favoring walks towards neighbors with more similar features, which indicate stronger homophily. The feature-based one-order probability P_u^{feat} is calculated based on the cosine similarity between the features of the current node u and candidate node v :

$$P_u^{feat}(v) = \cos(X_u, X_v), \text{ if } (u, v) \in \mathcal{E}. \quad (4)$$

The transition probability from node u to node v in a walk step is determined by combining the topology-based and feature-based probabilities, normalized by a softmax function:

$$P_u^{dst}(v) = \text{Softmax}\left(P_u^{topo}(v) + P_u^{feat}(v)\right). \quad (5)$$

Recognizing that the softmax scales vary with the walk directions, and we pre-compute P^{topo} and P^{feat} to ensure efficiency.

3.1.3 Homophily Entropy-based Personalized Walk Length. In DiRW, the walk length is dynamically determined based on homophily entropy rather than being predefined and fixed. Traditional homophily metrics predominantly focus on immediate neighborhood information, providing a limited view of the walk sequence’s quality. To address this limitation, we introduce a novel metric *Homophily Entropy*, which assesses the walk sequence quality based on feature similarity among nodes in the sequence. Our objective is to ensure sampled nodes in the walk sequence have a strong homophilous relationship with the originating node, enriching the aggregated information with relevance to it. We first convert the walk sequence into a homophily label sequence, represented as:

$$S^{homo}(i) = \mathbb{I}\left(X_{S^{walk}(i)} \neq X_{S^{walk}(0)}\right) \cdot i, \quad (6)$$

where $\mathbb{I}(\text{condition})$ equals 1 if the condition is met and 0 otherwise. $S^{walk}(i)$ denotes the i -th node in the walk sequence, and $S^{homo}(i)$ is the corresponding homophily label. By applying the Shannon entropy [14] on S^{homo} , we derive the homophily entropy:

$$H_{homo}(S^{walk}) = H(S^{homo}) = -\sum_{i=1}^k p(i) \log p(i), \quad (7)$$

where $p(i)$ represents the probability of encountering $S^{homo}(i)$ in S^{homo} and k denotes the length of the sequence. Our analysis indicates that a lower homophily entropy corresponds to a higher walk sequence quality.

Table 1: The statistician of the experimental datasets.

Characteristics	Datasets	#Nodes	#Edges	#Features	#Node Classes	#Train/Val/Test	#Description
Directed-Homophily	CoraML	2,995	8,416	2,879	7	140/500/2,355	citation network
	CiteSeer	3,312	4,591	3,703	6	120/500/2,692	citation network
	WikiCS	11,701	290,519	300	10	580/1,769/5,847	weblink network
	Amazon-Computers	13,752	287,209	767	10	200/300/12,881	co-purchase network
	ogbn-arxiv	169,343	2,315,598	128	40	91k/30k/48k	citation network
UnDirected-Homophily	ogbn-products	2,449,029	61,859,140	100	47	196k/49k/2204k	co-purchase network
Directed-Heterophily	Chameleon	890	13,584	2,325	5	48%/32%/20%	wiki-page network
	Actor	7,600	26,659	932	5	48%/32%/20%	actor network
	Rating	24,492	93,050	300	5	50%/25%/25%	rating network

This entropy metric serves for dynamically determining node-specific walk lengths. We start with a predefined minimum walk length l_{min} to guarantee the adequate information acquisition. Subsequent walk persists until two consecutive homophily entropy increments, signaling the heterophilous neighborhood. This adaptive scheme aligns with **Key Insight 2**, which explicitly accommodates nodes with homophilous neighborhoods.

3.1.4 Weight-free Path Encoding. To address the varying influence of nodes along a path, DiRW uses an exponential decay function to assign weights and compute the path embedding:

$$\mathbf{h}_u^{(l)} = \sum_{i=1}^k \frac{\gamma^i}{\sum_{j=1}^k \gamma^j} \mathbf{X}_{P_u^{(l)}(i)}, \quad (8)$$

where $\gamma \in (0, 1)$ represents the decay parameter and $P_u^{(l)}$ denotes the l -th sampled path of node u . This weight-free aggregation mechanism allows for the integration of the sampling into the pre-processing stage, which substantially reduces the computational complexity during training. Moreover, by giving greater significance to nodes closer to the originating node, the exponential decay mechanism effectively captures the sequential information inherent in the walk sequence, overcoming the constraints of uniform weighting often found in message-passing approaches [12].

3.1.5 Adaptive Walk Numbers. In DiRW, gathering comprehensive context is essential, which influences the walk numbers per node. This is determined by assessing the *Information Richness*, measured by the average of sampled path embeddings. To avoid early walk termination, a minimum walk number n_{min} is set. After each walk, DiRW computes the L2 norm of the Information Richness difference between the current and previous sampled sequences:

$$\Delta = \left\| \frac{1}{L} \sum_{l=1}^L \mathbf{h}_u^{(l)} - \frac{1}{L-1} \sum_{l=1}^{L-1} \mathbf{h}_u^{(l)} \right\|_2, \quad L > n_{min}, \quad (9)$$

where L is the number of sampled path sequences.

If Δ falls below the predefined threshold δ , it indicates that the sampling has reached a saturation point, beyond which additional walks are unlikely to contribute substantially to the information richness. In such cases, the walk is stopped to avoid further unnecessary and redundant sampling, improving the information gathering efficiency and reflecting **Key Insight 2**: heterophilous nodes are often characterized by more intricate neighborhoods and necessitate more walks to amass sufficiently rich information.

3.2 Learnable Path Aggregator

3.2.1 Attention-based Node Encoding. After the pre-processing sampling phase, DiRW harnesses an attention mechanism to discern and weigh the informativeness of various path embeddings. This process is pivotal for constructing node embeddings that are rich in relevant contextual information.

Specifically, for each node u , DiRW initiates the encoding of the sampled paths \mathbf{h}_u through a pair of linear layers, interspersed with a LeakyReLU activation function. The encoded representation \mathbf{e}_u is subsequently processed through a softmax function to yield attention scores α_u . The final node embedding \mathbf{z}_u is then computed as a weighted aggregation of the path embeddings $\mathbf{h}_u^{(i)}$, with the weights being the attention scores α_u^i , articulated as follows:

$$\begin{aligned} \mathbf{e}_u &= \text{MLP}_2(\text{LeakyReLU}(\text{MLP}_1(\mathbf{h}_u))), \\ \alpha_u &= \text{Softmax}(\mathbf{e}_u), \quad \mathbf{z}_u = \sum_{l=1}^L \alpha_u^l \mathbf{h}_u^{(l)}. \end{aligned} \quad (10)$$

3.2.2 Node Classifier. With the node embeddings \mathbf{z}_u at hand, DiRW deploys an MLP to tackle the node classification task. The training process is directed by the cross-entropy loss as follows:

$$\hat{\mathbf{y}}_u = \text{Softmax}(\text{MLP}_3(\mathbf{z}_u)), \quad \mathcal{L} = -\frac{1}{|\mathcal{V}_l|} \sum_{i \in \mathcal{V}_l} \mathbf{Y}_i \log \hat{\mathbf{y}}_i, \quad (11)$$

where \mathcal{V}_l represents the training set. \mathbf{Y}_i and $\hat{\mathbf{y}}_i$ are the one-hot encoded true label and the predicted label of node i , respectively.

In the link prediction task, DiRW estimates the probability of edges between node pairs using another MLP. The input to this MLP consists of the concatenated embeddings of the node pairs, allowing the model to leverage the feature information from both nodes. The training process is also guided by the binary cross-entropy loss.

4 EXPERIMENTS

In this section, we conduct a comprehensive evaluation of our DiRW, structured to address five critical research dimensions: **Q1**: Can DiRW establish superior performance as both a new DiGNN and a plug-and-play strategy? **Q2**: If DiRW proves effective, what architectural characteristics contribute to its enhanced performance? **Q3**: How does DiRW’s time complexity compare with baseline methods in practical deployment scenarios? **Q4**: How does the hyperparameters sensitivity and robustness of DiRW manifest? **Q5**: How does a comparative analysis of path quality between DiRW and existing PathGNNs inform its structural learning advantages?

Table 2: Model performance (%) as a new DiGNN in node classification. The best result is **bold. The second result is underlined.**

Type	Models	CoraML	CiteSeer	WikiCS	Amazon Computers	Chameleon	Actor	Rating	Arxiv	Products	Rank
Undirected GNNs	GCN	84.48±0.17	65.26±1.07	78.98±0.49	83.26±0.51	40.93±1.24	30.96±0.45	43.37±0.13	67.80±0.07	73.3±0.07	5.67
	GAT	83.73±0.47	63.12±1.06	79.35±0.28	80.16±2.32	40.82±2.46	30.29±0.37	45.03±0.50	<u>67.79±0.24</u>	OOM	7.44
Spatial DiGNNs	DGCN	81.27±0.58	64.66±0.65	78.71±0.19	82.95±1.13	41.96±1.00	30.87±0.38	44.24±0.28	OOT	OOM	8.00
	NSTE	80.06±0.89	62.99±1.08	77.58±0.29	78.50±1.87	40.31±1.87	31.14±0.46	45.31±0.15	64.40±0.94	OOT	9.78
	DIMPA	79.77±0.66	60.88±1.08	78.58±0.27	77.80±1.42	40.41±1.88	30.84±0.73	41.58±0.17	65.89±0.22	72.89±0.13	10.67
	Dir-GNN	79.92±1.26	63.74±0.87	79.04±0.26	81.34±0.92	41.96±1.95	30.12±0.65	43.74±0.10	64.26±0.48	72.93±0.17	8.33
	ADPA	79.10±1.56	65.01±2.29	<u>79.43±0.92</u>	73.25±4.26	41.66±3.07	35.29±0.82	44.90±0.62	64.28±0.56	OOT	7.67
Spectral DiGNNs	DiGCN	80.81±0.52	62.38±0.28	79.30±0.32	80.02±1.52	41.24±1.67	34.30±0.89	<u>45.99±0.28</u>	65.38±0.47	OOM	7.11
	DiGCNappr	80.65±0.40	62.05±0.59	78.54±0.32	81.53±1.27	40.93±2.49	30.99±0.49	40.46±0.20	61.57±0.56	73.51±0.05	9.33
	MagNet	80.45±0.46	<u>65.63±1.69</u>	79.19±0.20	<u>83.41±0.99</u>	<u>42.78±0.63</u>	31.39±0.53	45.97±0.71	68.22±0.25	<u>74.21±0.06</u>	3.78
	MGC	82.66±1.01	64.78±1.05	77.42±0.28	81.26±0.80	41.34±3.29	35.75±0.41	42.01±0.36	66.73±0.17	59.86±3.20	7.11
	LightDiC	82.67±1.10	64.36±0.35	78.02±0.21	79.87±0.68	42.68±1.61	30.59±0.50	40.53±0.48	63.41±0.26	64.85±0.07	8.78
PathGNNs	RAWGNN	78.35±0.90	59.94±0.38	76.76±0.31	72.76±0.67	41.24±3.34	34.71±0.80	44.07±0.32	OOM	OOM	11.89
	PathNet	66.61±6.83	61.41±1.93	75.71±0.66	68.78±2.13	38.44±3.21	36.67±0.92	39.15±0.40	OOM	OOM	13.22
	DiRW	<u>84.24±0.19</u>	65.88±1.41	79.87±0.08	83.50±0.68	43.92±0.23	<u>36.42±0.46</u>	46.13±0.39	68.51±0.18	75.19±0.47	1.22

Table 3: Model Performance (%) as a plug-and-play strategy on PathGNNs in node classification.

Models	CoraML	CiteSeer	WikiCS	Amazon Computers	Chameleon	Actor	Rating	Improvement
RAWGNN	78.35±0.90	59.94±0.38	76.76±0.31	72.76±0.67	41.24±3.34	34.71±0.80	44.07±0.32	
RAWGNN+DiRW	82.67±0.26	64.54±0.75	77.71±0.27	76.33±1.02	43.30±1.31	35.55±0.39	45.66±0.63	↑ 4.34%
PathNet	66.61±6.83	61.41±1.93	75.71±0.66	68.78±2.13	38.44±3.21	36.67±0.92	39.15±0.40	
PathNet+DiRW	68.25±0.77	62.66±0.89	76.21±0.35	69.45±0.97	38.88±1.04	37.14±0.87	39.22±0.24	↑ 1.25%
DiRW	84.24±0.19	65.88±1.41	79.87±0.08	83.50±0.68	43.92±0.23	36.42±0.46	46.13±0.39	—

4.1 Experimental Setup

4.1.1 Datasets. Our comprehensive evaluation spans 9 benchmark datasets encompassing both directed and undirected graphs with diverse homophily characteristics. We have documented the detailed information of these datasets in the Tab. 1.

4.1.2 Baselines. Our experiment leverages diverse GNNs as comparative benchmarks, which can be categorized as follows: (i) traditional undirected approaches: GCN [12], GAT [34]; (ii) directed spatial methods: DGCN [33], NSTE [13], DIMPA [8], Dir-GNN [26], ADPA [28]; (iii) directed spectral methods: DiGCN [32] and its variant DiGCNappr, MagNet [45], MGC [43], LightDiC [15]; (iiii) PathGNNs: RAWGNN [11] and PathNet [29]. To minimize randomness and ensure fair comparisons, we repeated each experiment 10 times to obtain unbiased performances. Moreover, we transform the digraph to undirected graph and feed it into undirected GNNs.

4.1.3 Hyperparameter Settings. The hyperparameters in the baseline GNNs are set following the original paper if available. Otherwise, we perform a hyperparameter search via the Optuna [3]. In DiRW, we perform a grid search for the minimum walking length ranging from 2 to 6, and for the minimum walk number ranging from 2 to 10. Furthermore, we fine-tune the walk direction coefficient q within the interval [0.5, 1].

4.1.4 Experimental Environment. Our experiments are conducted on the machine with Intel(R) Xeon(R) Platinum 8468V, NVIDIA H800 PCIe, and CUDA 12.2. The operating system is Ubuntu 20.04.6. As for the software, we use Python 3.8 and Pytorch 2.2.1.

4.2 Overall Performance

4.2.1 An Innovative Learning Architecture. To answer **Q1**, we conduct comparative experiments to evaluate the performance of DiRW in node classification. The results presented in Tab. 2 demonstrate that, as a novel DiGNN, DiRW achieves exceptional performance across all datasets, outperforming the leading DiGNN, MagNet, by an average of 3.2%. This improvement is attributed to its effective modeling of digraphs and heterophilous relationships.

In contrast, traditional undirected GNNs perform well in homophilous scenarios but struggle in heterophilous digraphs due to their reliance on simplistic undirected adjacency matrix approaches. While Dir-GNN and ADPA attempt to address heterophily through digraph modeling, their methods exhibit limitations in generalizing effectively to homophilous digraphs. Furthermore, although undirected PathGNNs show potential in modeling heterophilous graphs, their performance is significantly compromised by neglect of edge direction and simplistic walk strategies. Notably, they fail to account for edge directionality in digraphs, and their simplistic walk strategies limit their effectiveness in complex digraph structures.

Emphasize that the sophisticated model architectures like RNNs in PathGNNs cause scalability issues, leading to out-of-memory (OOM) errors with large-scale graphs like ogbn-arxiv and ogbn-products. Similarly, the intricate message-passing paradigm with multiple convolutional layers in DGCN and NSTE often leads to incomplete training within 12 hours, resulting in out-of-time (OOT) errors. In contrast, DiRW demonstrates high efficiency and superior performance on large-scale digraphs through its weight-free path sampling strategy and lightweight learning mechanism.

Table 4: Model performance (%) in link direction prediction. The best result is **bold. The second result is underlined.**

Models	CoraML	CiteSeer	WikiCS	Amazon Computers	Chameleon	Actor	Rating	Arxiv
GCN	83.63±0.74	74.29±0.73	72.97±0.23	92.42±0.14	87.98±0.48	58.37±1.34	50.16±0.74	82.24±0.12
GAT	81.28±3.40	79.33±2.24	70.18±9.71	67.35±0.85	85.80±1.83	54.13±0.42	50.84±0.88	76.55±4.46
DGCN	89.47±0.25	85.09±0.29	87.02±0.08	95.67±0.13	91.38±0.58	82.58±0.66	78.96±0.10	OOT
DiGCN	88.15±0.57	87.99±1.77	85.38±0.19	96.45±0.09	89.84±0.61	82.77±0.44	80.92±0.05	92.71±0.04
DiGCNapp	84.96±0.23	82.50±0.51	83.41±0.10	95.72±0.07	91.54±1.25	81.60±0.25	78.92±0.06	91.71±0.01
NSTE	90.10±0.59	87.86±0.60	87.69±0.15	95.49±0.56	91.28±0.69	79.74±0.20	80.29±0.39	92.31±0.07
DIMPA	<u>90.48±0.33</u>	86.83±0.50	82.54±0.15	95.65±0.16	89.68±0.69	82.33±0.24	83.06±0.06	93.56±0.12
MagNet	89.25±0.34	85.13±0.60	89.07±0.10	96.73±0.07	90.64±1.18	83.69±0.39	82.75±0.18	93.91±0.04
MGC	89.97±0.64	81.79±1.63	86.66±0.17	86.86±0.46	89.15±0.97	67.84±0.37	72.39±0.17	87.65±4.27
LightDiC	88.87±0.46	87.99±0.46	88.77±0.06	95.30±0.05	90.27±0.36	82.39±0.35	81.59±0.07	92.73±0.01
Dir-GNN	88.95±0.45	86.34±0.99	88.87±0.05	<u>96.93±0.03</u>	<u>91.97±0.63</u>	85.89±0.17	<u>85.01±0.05</u>	<u>94.73±0.05</u>
ADPA	90.28±0.78	<u>88.57±0.68</u>	<u>88.85±0.14</u>	96.66±0.06	89.52±0.83	84.88±0.34	82.69±0.24	93.81±0.04
RAWGNN	89.92±1.21	87.01±1.04	83.48±0.8	94.93±0.18	89.95±1.21	85.34±0.46	73.26±0.43	OOM
DiRW	91.05±0.57	88.96±1.38	88.79±0.49	97.15±0.24	92.19±1.17	<u>85.56±0.39</u>	85.49±0.34	95.27±0.08

4.2.2 A Plug-and-Play Approach. In addition to evaluating DiRW as an innovative neural architecture, we also integrate its optimized path sampler with other PathGNNs, specifically combining DiRW with RAWGNN and PathNet for digraph modeling. However, due to the complicated path and node embedding learning rules designed in these two models, it is tough for the weight-free adaptive walking length and number in DiRW to generalize to them. Therefore, we only incorporated the direction-aware sampling strategy and walking probabilities that consider both topological structure and node profiles into these two models to assess their performance. The results are demonstrated in Tab. 3.

The results provide strong evidence that our optimized path sampler significantly enhances the performance of both RAWGNN and PathNet. The performance gains stem from the shortcomings of the original PathGNNs, which fail to consider adequately the directionality of edges in digraphs, leading to a substantial loss of valuable information. DiRW corrects it by balancing the edge directionality and existence into its sampling process, effectively capturing the intricate topological structures inherent in digraphs. Moreover, our walking probabilities, which account for topological structure and node profiles, outperform those of RAWGNN and PathNet, which focus solely on topology. By modeling node similarity, DiRW effectively captures one-hop homophily in heterophilous contexts. This integration not only enhances performance when DiRW is used as a plug-and-play module with other PathGNNs but also highlights the adaptability and robustness of our approach.

4.2.3 Performance in Link Prediction. We extend the evaluation of DiRW through a multi-task framework, investigating both node classification and directional link prediction capabilities, with the results presented in Tab. 4. We directly leverage the node embeddings derived from Eq. (10) and concatenate them to form discriminative edge embeddings, thereby bypassing the need for additional node classifier as specified in Eq. (11). This concatenated vector serves as a comprehensive representation of the edges, preserving the geometric relationships encoded during the walk aggregation phase while maintaining architectural simplicity.

Table 5: Ablation study performance (%).

Models	CiteSeer	WikiCS	Chameleon	Rating
w/o Dir	65.11±1.77	79.65±0.18	42.47±1.07	45.59±0.16
w/o Topo	64.26±0.17	79.37±0.13	38.97±2.64	45.79±0.18
w/o Feat	65.77±1.28	79.27±0.13	41.65±1.73	45.62±0.25
w/o Att	65.02±0.75	79.26±0.32	39.59±2.23	42.55±0.39
DiRW-Gate	65.65±0.84	79.52±0.28	38.76±3.56	42.68±0.15
DiRW-JK	63.88±1.02	79.31±0.25	40.21±3.82	42.74±0.33
DiRW	65.88±1.41	79.87±0.08	43.92±0.23	46.13±0.39

The experimental outcomes provide compelling evidence that DiRW achieves an average performance gain of 0.82% over the second leading model Dir-GNN. The efficacy of DiRW in link prediction is a testament to its ability to capture the subtleties of graph structure and the nuanced relationships between nodes. This capability stems from the model’s sophisticated path sampling strategies and attention-based aggregation mechanism, which allow for a deep understanding of the digraph’s topology and node features.

It is important to note that traditional undirected GNNs have exhibited subpar performance and suffer from structural isomorphism issues in direction-sensitive tasks. The primary reason for this deficit lies in the direction-agnostic message-passing in undirected architectures, which fails to discriminate between $(u \rightarrow v)$ and $(v \rightarrow u)$ edge semantics. In contrast, DiRW’s design explicitly accounts for edge directionality, providing it with a distinct advantage in tasks where the direction of relationships is pivotal. This comparative analysis underscores the importance of adopting models that can accommodate the directed nature of graphs, thereby offering a more accurate representation of the underlying structure and relationships within the data.

4.3 Ablation Study

To answer Q2, we have conducted a series of ablation studies to ascertain the individual contributions of the five core components that constitute the backbone of DiRW.

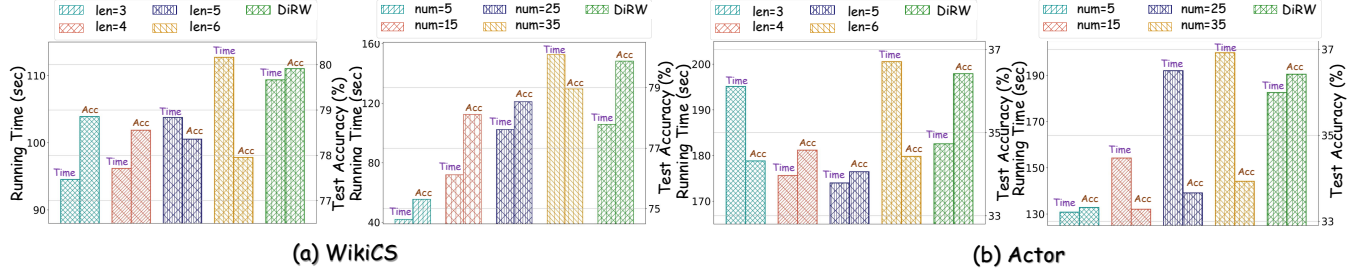


Figure 4: Performance and running time of DiRW and its variants w/o personalized walk length and adaptive walk number.

We began with the direction-aware path sampler by setting the walk direction coefficient q to 1, simulating a scenario without directionality (w/o Dir). Then, we conducted separate ablations for the topology-based (w/o Topo) and feature-based (w/o Feat) walk probabilities to assess the relative importance of node profiles and topological structure in guiding the walk. Additionally, we conducted an ablation experiment on our node-wise attention mechanisms, replacing them with direct averaging of path embeddings (w/o Att), Gate attention [1] (DiRW-Gate), and JK attention [39] (DiRW-JK). The outcomes are detailed in Tab. 5.

Furthermore, to assess the adaptive walk length and numbers, we replace them with predefined and fixed hyperparameters and depict the outcomes in Fig. 4. We measured the performance and running time across varying walk lengths and numbers while keeping other hyperparameters consistent with DiRW. From these results, we draw the following conclusions:

Direction-aware path sampler. By effectively balancing the direction and existence of edges, the direction-aware path sampler has significantly enhanced the model’s capacity to capture the intricate topological structures of digraphs, leading to an average performance improvement of 1.5%.

Multi-order walk probability. The topology-based high-order walk probability shows a clear advantage in detecting homophilous patterns, leading to significant improvements in predictive performance. Moreover, the feature-based one-order walk probability achieves an average performance increase of 1.9%, underscoring feature information’s importance in guiding walk preferences.

Node-wise learnable path aggregator. By utilizing our attention mechanism, DiRW effectively distinguishes between productive and unproductive sampling sequences, leading to high-quality node embeddings and an average performance boost of 5.36%. The node-wise attention mechanism employs a dual MLP, which captures complex nuances in path embeddings more effectively than the single MLP used in the JK and Gate attention mechanisms.

Homophily entropy-based personalized walk length. The personalized walk length greatly enhances DiRW’s performance, outperforming static walk length models that require significant pre-processing and often yield sub-optimal results. In contrast, DiRW leverages the homophily entropy to sample high-quality paths, streamlining the sampling process for optimal effectiveness.

Adaptive walk number. Echoing the success of the personalized walk length, the adaptive walk number also demonstrates a significant performance improvement over models that utilize a rigid walk number. It comes from our systematic evaluation of the information richness in sampling more informative walk sequences.

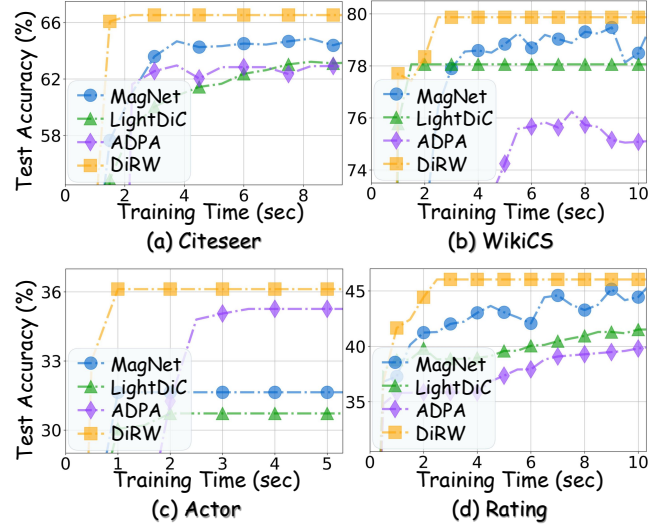


Figure 5: The Time-Accuracy performance comparison of DiRW and other DiGNNs.

4.4 Efficiency Analysis

To address Q3, we conducted experiments to evaluate the efficiency of DiRW. We selected a spatial-based baseline, ADPA, and two spectral-based baselines, Magnet and LightDiC, to benchmark their performance and efficiency. The metric for efficiency was established based on the total time required to run each model five times, with the results graphically depicted in Fig. 5.

From the visual representation, it is evident that the superiority of DiRW lies in both predictive accuracy and operational efficiency. Although optimal adaptive walk lengths and numbers vary between datasets, causing some fluctuations in computational time, DiRW achieves the best performance and efficiency due to its lightweight learning mechanism. ADPA is consistently the most time-consuming model, largely due to its intricate architecture, featuring hierarchical attention mechanisms and multiple convolution layers, which increases its computational and resource complexity.

Notably, the sampling stage constitutes the primary computational bottleneck in DiRW’s workflow, accounting for the majority of its running time. By adopting a pre-processing strategy analogous to PathNet, where we precompute and store sampled sequences, we can substantially optimize the training pipeline’s efficiency. This preparatory approach synergizes effectively with DiRW’s dedicated rapid learning mechanism, facilitating more accelerated training.

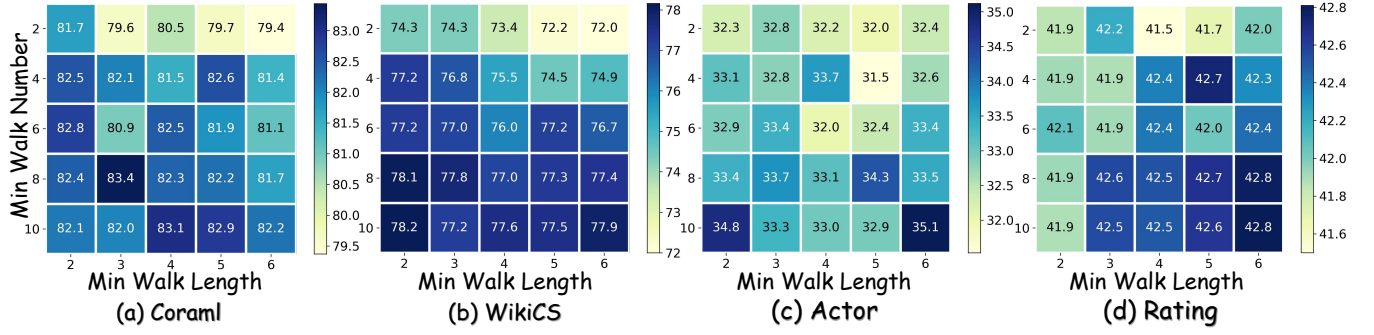


Figure 6: Sensitivity analysis of test accuracy to minimal walk length l_{min} and minimal walk number n_{min} on node classification.

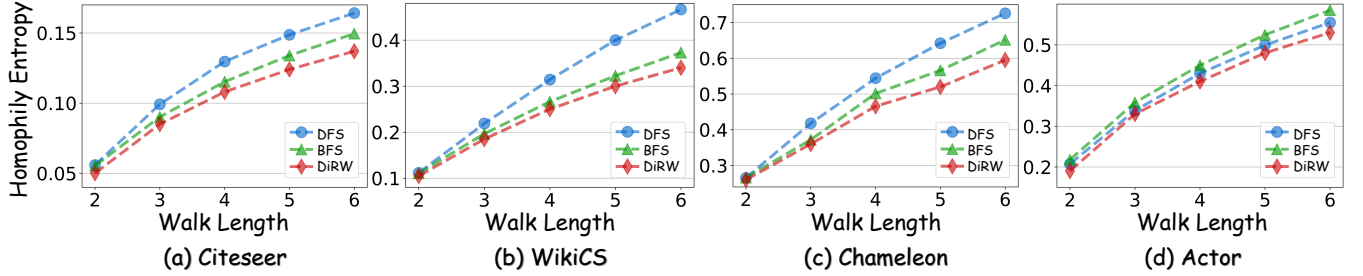


Figure 7: Path homophily entropy comparison of the optimized path sampler in DiRW and DFS/BFS in RAWGNN.

4.5 Sensitivity Analysis

To evaluate the impact of hyperparameters on performance, we conducted an in-depth analysis of the minimum walk length l_{min} and the minimum walk number n_{min} in Fig. 6. All other parameters were kept constant to isolate the effects of l_{min} and n_{min} .

Our analysis reveals a clear trend of improved performance as the minimum walk number n_{min} increases. However, while more walks enhance performance, this cannot be pursued indefinitely due to the accompanying rise in time and space complexity.

The impact of l_{min} is less straightforward. There is no obvious performance improvement or decrease as l_{min} increases. Notably, in homophilous digraphs (e.g., Coraml, WikiCS), shorter walk lengths yield better performance. Conversely, in heterophilous digraphs (e.g., Actor, Rating), longer walk lengths are linked to superior performance. This aligns with our earlier discussions on high-order homophily in heterophilous graphs, indicating that longer walks are crucial for capturing complex relationships within these structures.

4.6 Path Quality Analysis

We conducted an in-depth analysis comparing the path quality sampled by our optimized path sampler with those generated by the DFS and BFS utilized in RAWGNN, measured by the homophily entropy in Eq. (7). The experimental outcome is shown in Fig. 7.

The results demonstrate that the paths sampled by DiRW consistently exhibit lower homophily entropy compared to those produced by DFS and BFS. This superiority is primarily attributed to the integrated consideration of both topology and feature in the walk probability, coupled with the adaptive walk length guided by homophily entropy. Furthermore, the DFS demonstrated the highest homophily entropy in most scenarios because BFS explores the immediate neighborhood, which aligns well with the homophily

assumption. However, in scenarios characterized by strong heterophily (e.g., Actor), DFS outperforms BFS. DiRW achieves a harmonious balance between these two approaches, leveraging the strengths of both while mitigating their respective weaknesses.

5 CONCLUSION AND FUTURE WORK

In this work, we emphasize the necessity of DiGNNs for modeling digraph-structured data and introduce the DiPathGNN mechanism to capture inherent homophilous information in digraphs. Our empirical analysis reveals the limitations of current PathGNNs, particularly their overlook of edge direction and coarse-grained sampling strategies. To overcome these challenges, we propose DiRW, a novel path-based digraph learning method characterized by its direction-aware path sampler, which is fine-tuned based on walk probability, length, and number. Additionally, DiRW utilizes a node-wise learnable path aggregator to create nuanced node representations. Experiments show that DiRW achieves state-of-the-art performance in node- and link-level tasks, particularly in heterophilous scenarios, providing an efficient and robust solution for digraph learning. Looking ahead, several promising directions for future research emerge. First, we plan to explore more efficient sampling strategies. Then, developing adaptive sampling methods would also improve model performance. Lastly, designing improved learning mechanisms to capture the complex relationships in digraphs may yield more accurate predictions. These future efforts will deepen our understanding of digraph learning and could lead to breakthroughs in modeling complex digraph-structured data.

ACKNOWLEDGMENTS

This work is supported by the NSFC Grants U2241211, 62427808, and U24A20255. Rong-Hua Li is the corresponding author.

GENAI USAGE DISCLOSURE

We used ChatGPT exclusively for language polishing in this work. We take full responsibility for the integrity of the work’s intellectual content and its adherence to academic ethical standards.

REFERENCES

- [1] Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. GATE: graph attention transformer encoder for cross-lingual relation and event extraction. In *Proceedings of the Association for the Advancement of Artificial Intelligence, AAAI*.
- [2] Usman Ahmed, Jerry Chun-Wei Lin, and Gautam Srivastava. 2022. Hyper-Graph Attention Based Federated Learning Method For Mental Health Detection. *IEEE Journal of Biomedical and Health Informatics* (2022).
- [3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*.
- [4] Vicente Balmaseda, María Coronado, and Gonzalo de Cadenas-Santiago. 2023. Predicting Systemic Risk in Financial Systems Using Deep Graph Learning. *Intelligent Systems with Applications* (2023), 200240.
- [5] Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. 2020. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198* (2020).
- [6] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*.
- [7] Mingguo He, Zhewei Wei, Hongteng Xu, et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems, NeurIPS* (2021).
- [8] Yixuan He, Gesine Reinert, and Mihai Cucuringu. 2022. DIGRAC: Digraph Clustering Based on Flow Imbalance. In *Learning on Graphs Conference, LoG*.
- [9] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R Benson. 2021. Combining label propagation and simple models out-performs graph neural networks. *International Conference on Learning Representations, ICLR* (2021).
- [10] Woonchang Hyun, Jaehong Lee, and Bongwon Suh. 2023. Anti-Money Laundering in Cryptocurrency via Multi-Relational Graph Neural Network. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 118–130.
- [11] Di Jin, Rui Wang, Meng Ge, Dongxiao He, Xiang Li, Wei Lin, and Weixiong Zhang. 2022. RAW-GNN: RAndom Walk Aggregation based Graph Neural Network. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*.
- [12] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations, ICLR*.
- [13] Georgios Kollias, Vasileios Kalantzis, Tsuyoshi Idé, Aurélie Lozano, and Naoki Abe. 2022. Directed Graph Auto-Encoders. In *Proceedings of the Association for the Advancement of Artificial Intelligence, AAAI*.
- [14] Angsheng Li and Yicheng Pan. 2016. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory* 62, 6 (2016), 3290–3339.
- [15] Xunkai Li, Meihao Liao, Zhengyu Wu, Daohan Su, Wentao Zhang, Rong-Hua Li, and Guoren Wang. 2024. LightDiC: A Simple Yet Effective Approach for Large-Scale Digraph Representation Learning. *Proceedings of the VLDB Endowment* 17, 7 (2024), 1542–1551.
- [16] Xunkai Li, Jingyuan Ma, Zhengyu Wu, Daohan Su, Wentao Zhang, Rong-Hua Li, and Guoren Wang. 2024. Rethinking Node-wise Propagation for Large-scale Graph Learning. In *Proceedings of the ACM Web Conference, WWW*.
- [17] Jiaxuan Liang, Jun Wang, Guoxian Yu, Wei Guo, Carlotta Domeniconi, and Maozu Guo. 2023. Directed acyclic graph learning on attributed heterogeneous network. *IEEE Transactions on Knowledge and Data Engineering* (2023).
- [18] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. 2019. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4424–4431.
- [19] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. 2022. Revisiting heterophily for graph neural networks. *Advances in Neural Information Processing Systems, NeurIPS* (2022).
- [20] Yuankai Luo, Veronika Thost, and Lei Shi. 2023. Transformers over Directed Acyclic Graphs. *Advances in Neural Information Processing Systems, NeurIPS* (2023).
- [21] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. 2021. Is homophily a necessity for graph neural networks? *International Conference on Learning Representations, ICLR* (2021).
- [22] Seiji Maekawa, Yuya Sasaki, and Makoto Onizuka. 2023. Why Using Either Aggregated Features or Adjacency Lists in Directed or Undirected Graph? Empirical Study and Simple Classification Method. *arXiv preprint arXiv:2306.08274* (2023).
- [23] Bastian Pfeifer, Anna Saranti, and Andreas Holzinger. 2022. GNN-SubNet: disease subnetwork detection with explainable graph neural networks. *Bioinformatics* 38, Supplement_2 (2022), ii120–ii126.
- [24] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at the evaluation of GNNs under heterophily: are we really making progress? *International Conference on Learning Representations, ICLR* (2023).
- [25] Yuxin Qiu. 2023. Default Risk Assessment of Internet Financial Enterprises Based on Graph Neural Network. In *IEEE Information TechnoLoGy, Networking, Electronic and Automation Control Conference*, Vol. 6. IEEE, 592–596.
- [26] Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael Bronstein. 2023. Edge Directionality Improves Learning on Heterophilic Graphs. In *Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML-PKDD Workshop* (2023).
- [27] Daohan Su, Bowen Fan, Zhi Zhang, Haoyan Fu, and Zhida Qin. 2024. DCL: Diversified Graph Recommendation With Contrastive Learning. *IEEE Transactions on Computational Social Systems* (2024).
- [28] Henan Sun, Xunkai Li, Zhengyu Wu, Daohan Su, Rong-Hua Li, and Guoren Wang. 2024. Breaking the entanglement of homophily and heterophily in semi-supervised node classification. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2379–2392.
- [29] Yifei Sun, Haoran Deng, Yang Yang, Chunping Wang, Jiarong Xu, Renhong Huang, Linfeng Cao, Yang Wang, and Lei Chen. 2022. Beyond Homophily: Structure-aware Path Aggregation Graph Neural Network. In *IJCAI*. 2233–2240.
- [30] Qiaoyu Tan, Xin Zhang, Ninghao Liu, Daochen Zha, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. 2023. Bring your own view: Graph neural networks for link prediction with personalized subgraph selection. In *Proceedings of the ACM International Conference on Web Search and Data Mining, WSDM*.
- [31] Veronika Thost and Jie Chen. 2021. Directed acyclic graph neural networks. *arXiv preprint arXiv:2101.07965* (2021).
- [32] Zekun Tong, Yuxuan Liang, Changsheng Sun, Xinke Li, David Rosenblum, and Andrew Lim. 2020. Digraph inception convolutional networks. *Advances in Neural Information Processing Systems, NeurIPS* (2020).
- [33] Zekun Tong, Yuxuan Liang, Changsheng Sun, David S Rosenblum, and Andrew Lim. 2020. Directed graph convolutional network. *arXiv preprint arXiv:2004.13970* (2020).
- [34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations, ICLR*.
- [35] Hongwei Wang and Jure Leskovec. 2020. Unifying graph convolutional neural networks and label propagation. *arXiv preprint arXiv:2002.06755* (2020).
- [36] Xiyuan Wang and Muhan Zhang. 2022. How Powerful are Spectral Graph Neural Networks. In *International Conference on Machine Learning, ICML*.
- [37] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International Conference on Machine Learning, ICML*.
- [38] Chenxuan Xie, Jiajun Zhou, Shengbo Gong, Jiacheng Wan, Jiaxu Qian, Shanqing Yu, Qi Xuan, and Xiaoni Yang. 2023. PathMLP: Smooth Path Towards High-order Homophily. *arXiv preprint arXiv:2306.13532* (2023).
- [39] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning, ICML*.
- [40] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. 2022. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462* (2022).
- [41] Liangwei Yang, Shengjie Wang, Yunzhe Tao, Jiankai Sun, Xiaolong Liu, Philip S Yu, and Taiqing Wang. 2023. DGRec: Graph Neural Network for Recommendation with Diversified Embedding Generation. In *Proceedings of the ACM International Conference on Web Search and Data Mining, WSDM*.
- [42] Yiding Yang, Xinchao Wang, Mingli Song, Junsong Yuan, and Dacheng Tao. 2021. Spagan: Shortest path graph attention network. *arXiv preprint arXiv:2101.03464* (2021).
- [43] Jie Zhang, Bo Hui, Po-Wei Harn, Min-Te Sun, and Wei-Shinn Ku. 2021. MGC: A complex-valued graph convolutional network for directed graphs. *arXiv e-prints* (2021), arXiv–2110.
- [44] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems, NeurIPS* (2018).
- [45] Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. 2021. Magnet: A neural network for directed graphs. *Advances in Neural Information Processing Systems, NeurIPS* (2021).
- [46] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. 2022. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082* (2022).
- [47] Honglu Zhou, Advith Cheguri, Samuel S Sohn, Zuohui Fu, Gerard De Melo, and Mubbasir Kapadia. 2022. D-HYPR: Harnessing Neighborhood Modeling and Asymmetry Preservation for Digraph Representation Learning. *Proceedings of the ACM International Conference on Information and Knowledge Management, CIKM* (2022).