

Leveraging Personalized PageRank and Higher-Order Topological Structures for Heterophily Mitigation in Graph Neural Networks

Yumeng Wang¹, Zengyi Wo^{1,2}, Wenjun Wang^{1,2,3}, Xingcheng Fu⁴, Minglai Shao¹

¹School of New Media and Communication, Tianjin University, Tianjin, China

²College of Intelligence and Computing, Tianjin University, Tianjin, China

³Yazhou Bay Innovation Institute, Hainan Tropical Ocean University, Sanya, Hainan, China

⁴Key Lab of Education Blockchain and Intelligent Technology, Guangxi Normal University, China
{ymwang, wozengyi1999, wjwang, shaoml}@tju.edu.cn, fuxc@gxnu.edu.cn

Abstract

Graph Neural Networks (GNNs) excel in node classification tasks but often assume homophily, where connected nodes share similar labels. This assumption does not hold in many real-world heterophilic graphs. Existing models for heterophilic graphs primarily rely on pairwise relationships, overlooking multi-scale information from higher-order structures. This leads to suboptimal performance, particularly under noise from conflicting class information across nodes. To address these challenges, we propose **HPGNN**, a novel model integrating **Higher-order Personalized PageRank** with **Graph Neural Networks**. HPGNN introduces an efficient high-order approximation of Personalized PageRank (PPR) to capture long-range and multi-scale node interactions. This approach reduces computational complexity and mitigates noise from surrounding information. By embedding higher-order structural information into convolutional networks, HPGNN effectively models key interactions across diverse graph dimensions. Extensive experiments on benchmark datasets demonstrate HPGNN’s effectiveness. The model achieves better performance than five out of seven state-of-the-art methods on heterophilic graphs in downstream tasks while maintaining competitive performance on homophilic graphs. HPGNN’s ability to balance multi-scale information and robustness to noise makes it a versatile solution for real-world graph learning challenges. Codes are available at <https://github.com/streetcorner/HPGNN>.

1 Introduction

Graph Representation Learning (GRL) tackles the challenge of capturing complex relational information in non-Euclidean structured data, enabling advanced machine learning applications on graphs [Hamilton *et al.*, 2017; Hamilton, 2020]. Graph Neural Networks (GNNs), a cornerstone of GRL, were originally developed under the homophily assumption, where connected nodes share similar properties. Early GNN models focused on homophilic graphs [Veličković *et al.*, 2017;

Xu *et al.*, 2018], yet many real-world networks exhibit heterophily, with connections spanning dissimilar nodes. The performance of GNNs often depends on the graph’s homophily level [Pei *et al.*, 2020], driving recent efforts to address heterophily [Jin *et al.*, 2021; Zhu *et al.*, 2022; Wu *et al.*, 2023b; Qiu *et al.*, 2024; Wo *et al.*, 2024].

Despite these advances, most prior work centers on pairwise node relationships [Majhi *et al.*, 2022], which fail to capture the multi-scale, complex interactions prevalent in real-world scenarios while higher-order structures better reflect real-world complex systems [Benson *et al.*, 2016]. To overcome this limitation, higher-order network methods, such as hypergraphs and simplicial complexes (SCs), have emerged. Hypergraphs generalize graphs by allowing edges to connect multiple nodes but often neglect intra-hyperedge relationships, limiting their robustness to noise compared to SCs [Antelmi *et al.*, 2023]. Grounded in algebraic topology [Spanier and Spanier, 1989], SCs model multidimensional relationships, offering richer node context [Hajij *et al.*, 2021] and proving effective in domains like propagation [Iacopini *et al.*, 2019], sensor networks [Schaub *et al.*, 2021], and social systems [Fu *et al.*, 2024; Benson *et al.*, 2016].

A central challenge in higher-order neural networks lies in constructing node relationships using SCs. Recent approaches leverage the Hodge Laplacian from SC theory [Ebli *et al.*, 2020; Roddenberry *et al.*, 2021; Yang *et al.*, 2022], but its computation can reach $\mathcal{O}(n^3)$ complexity in dense graphs, posing scalability issues. Meanwhile, **Personalized PageRank (PPR)**, a random walk variant with teleportation [Page, 1999], efficiently captures local node information. Recent studies integrate PPR with GNNs to enhance representations while preserving efficiency [Gasteiger *et al.*, 2018; Chien *et al.*, 2020; Jack *et al.*, 2023], though they often overlook multi-level, higher-order interactions.

In this paper, we propose **HPGNN**, a novel framework that combines higher-order topological structures with Personalized PageRank to advance graph representation learning, particularly for heterophilic graphs. Unlike methods limited to node-to-node interactions, HPGNN incorporates relationships across multiple simplicial complexes. Our approach comprises two key components: 1) **Higher-order Personalized PageRank (HiPPR)**, which efficiently computes long-range interactions while reducing noise, and 2) **Higher-order**

Adaptive Spectral Convolution (HiASC), which extends spectral convolution to capture higher-dimensional relationships via SCs. Our contributions are threefold:

- HPGNN pioneers the integration of higher-order information into PPR approximation, bridging simplicial complex theory with personalized random walks.
- We introduce an adaptive PPR matrix operator that encodes interactions among higher-order graph structures.
- Extensive experiments on real-world datasets demonstrate that HPGNN achieves better performance than five out of seven state-of-the-art models.

2 Related Work

In this section, we provide a brief discussion of related work on learning from graphs with heterophily and personalized pageRank.

2.1 GNNs with Heterophily

Recent GNNs for heterophilous graphs break traditional homophily assumptions, APPNP [Gasteiger *et al.*, 2018] decouples feature transformation and propagation, while GPRGNN [Chien *et al.*, 2020] adaptively combines graph filters for task-specific aggregation. They enable long-range information propagation and reduce over-smoothing, preserving node-specific features. BernNet expands by leveraging Bernstein polynomial filters to effectively balance low and high frequency signals to handle label inconsistency in heterophily [Chen *et al.*, 2022]. Most recently, POLYGCL [Chen *et al.*, 2024] improves graph contrastive learning via structure-aware sampling and polynomial propagation, preserving structural roles and handling heterophily. Nonetheless, these methods generally lack a high-order perspective to stably suppress boundary noise caused by conflicting class information across nodes.

2.2 Higher-order Simplicial Neural Networks

Simplicial Neural Networks focus on efficiently modeling higher-order structures and embedding them into neural networks. BScNets use block operators to capture dependencies across multi-order structures. [Chen *et al.*, 2022]. Another approach extends the block matrix concept, defining an inter-order adjacency matrix for transformations between simplices [Zhou *et al.*, 2024]. SCN introduces random walks on simplices, enabling comprehensive simplex learning [Wu *et al.*, 2023a]. HiGCN extends SCN with two-step random walks, capturing distant higher-order relationships [Huang *et al.*, 2024]. Despite these advancements, traditional random walk methods face issues like rapid information diffusion and lack of personalized control, hindering effective node importance differentiation.

3 Preliminary

This subsection provides the foundational concepts and problem setting for our work.

3.1 Background

Our work builds on foundational concepts of graphs, graph heterophily, simplicial complexes, and clique complex lifting.

Graph. Consider an undirected, unweighted graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_N\}$ is a finite set of $N = |V|$ nodes, and $E \subseteq V \times V$ is the set of $M = |E|$ edges. Let $X \in \mathbb{R}^{N \times F}$ denote the node feature matrix, where F is the feature dimension, and $A \in \{0, 1\}^{N \times N}$ represent the adjacency matrix, with $A_{ij} = 1$ if $(v_i, v_j) \in E$ and 0 otherwise.

Graph Heterophily. Graph homophily reflects the tendency of connected nodes to share similar attributes, while heterophily indicates connections between dissimilar nodes [McPherson *et al.*, 2001]. We quantify homophily using the node homophily ratio:

$$H_{\text{node}} = \frac{1}{N} \sum_{v \in V} \frac{|\{(u, v) \in E : y_u = y_v\}|}{d_v}, \quad (1)$$

where y_v is the label of node v , and d_v is its degree. The ratio $H_{\text{node}} \in [0, 1]$ indicates the level of homophily, with lower values signifying stronger heterophily. This metric is critical for understanding network properties in heterophilic settings [Pei *et al.*, 2020].

Simplicial Complexes. A simplicial complex \mathcal{K} is a finite collection of simplices satisfying: 1) a k -simplex $\sigma^k = \{v_0, v_1, \dots, v_k\}$ is the convex hull of $k + 1$ vertices; 2) if $\sigma^k \in \mathcal{K}$, all its subsets (faces) are in \mathcal{K} ; and 3) \mathcal{K} is finite [Battiston *et al.*, 2020]. Simplicial complexes model higher-order interactions, where a 0-simplex is a vertex, a 1-simplex an edge, a 2-simplex a triangle, and so forth, enabling the study of topological relationships.

3.2 Problem Setting

In semi-supervised node representation learning, the goal is to learn an encoder $f(\cdot) : \mathbb{R}^{N \times F} \times \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times D}$ that maps the node feature matrix X and adjacency matrix A to a representation matrix $Z = f(X, A) = \{z_1, z_2, \dots, z_N\}$, where $z_i \in \mathbb{R}^D$ is the D -dimensional representation of node v_i . These representations are optimized for downstream tasks, such as node classification or clustering, particularly in graphs with varying levels of homophily or heterophily.

4 Methodology

Overview. We propose the **Higher-order Personalized PageRank Graph Neural Network (HPGNN)** to address heterophily and model complex node interactions in graph representation learning. As shown in Figure 1, HPGNN integrates three components: (1) Clique complex lifting [Schaub *et al.*, 2020] to transform the original pairwise graphs into simplicial complexes, providing more comprehensive views of the graph structure; (2) Higher-order Personalized PageRank (HiPPR) to model long-range dependencies while mitigating noise in heterophilic graphs; and (3) Higher-order Adaptive Spectral Convolution (HiASC) to model higher-dimensional relationships via spectral methods. By combining HiPPR and HiASC, HPGNN produces robust node representations that excel in heterophilic settings and remain competitive in homophilic ones, enabling superior performance in downstream tasks.

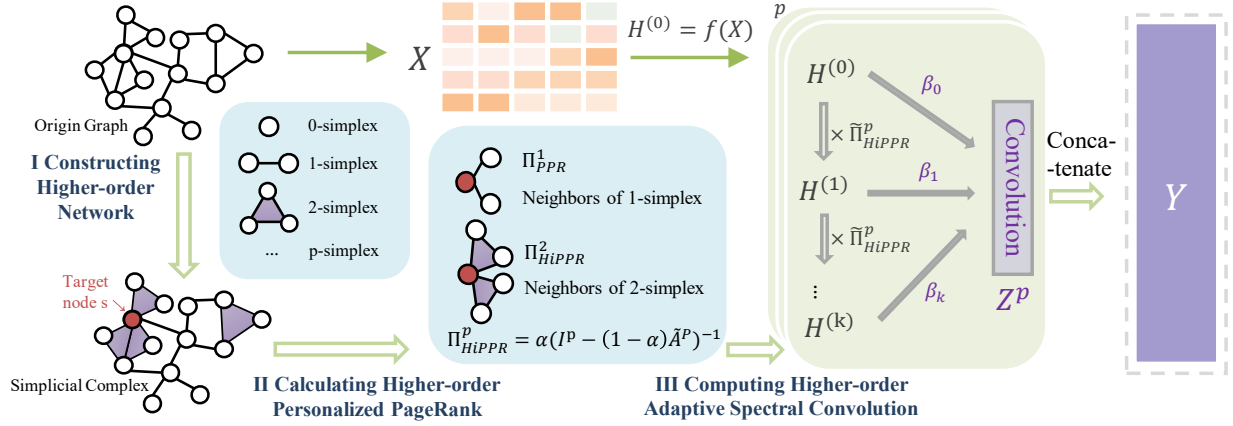


Figure 1: HPGNN ARCHITECTURE. A modular framework comprising: (1) clique complex lifting to construct higher-order topological structures, (2) Higher-order Personalized PageRank (HiPPR) for robust long-range dependency modeling and (3) Higher-order Adaptive Spectral Convolution (HiASC) to capture multi-dimensional interactions.

4.1 Constructing Higher-order Network

To capture higher-order interactions, we transform a graph $G = (V, E)$ into simplicial complexes (SCs) via clique complex lifting [Schaub *et al.*, 2020]. Let $C(G)$ denote the set of all cliques in G , where a clique $C_i \subseteq V$ is a complete subgraph. Each p -clique in $C(G)$ is mapped to a $(p-1)$ -simplex, where 2-simplices are the 3-cliques in the graph, forming the SC $\mathcal{K} = \bigcup_{C_i \in C(G)} \sigma_i$. This process encodes multi-node interactions beyond pairwise edges.

4.2 Higher-order Personalized PageRank

PPR is a variant based on random walks, it modified the original PageRank algorithm to incorporate a “teleport” mechanism that allows us to personalize the random walk starting from a specific root node [Page, 1999]. This adaptation ensures the influence of the root node is reflected, preserving the local structure around it even in the long run.

On pairwise graphs, The root node s is defined via the teleport vector i_s , which is a one-hot indicator vector. The PPR of node s can be computed as: $\pi_{\text{ppr}}(i_s) = \alpha i_s + (1 - \alpha) \tilde{A} \pi_{\text{ppr}}(i_s)$, with the teleport (or restart) probability $\alpha \in (0, 1]$. By solving the equation and substituting the indicator vector i_s with the unit matrix I_s , we obtain the PPR matrix:

$$\Pi_{\text{PPR}} = \alpha \left(I_n - (1 - \alpha) \tilde{A} \right)^{-1}, \quad (2)$$

where \tilde{A} is the normalized adjacency matrix, defined as $\tilde{A} = AD^{-1}$, A is the adjacency matrix of the graph and D is the degree matrix. I_n is the identity matrix of size $n \times n$, and n is the number of nodes in the graph. The matrix Π_{ppr} thus represents the personalized influence of node s on all other nodes in the graph.

Higher-order Information Integration in PPR

After incorporating the higher-order information, we design to treat the node as a simplex to obtain higher-order PPR. The HiPPR vector Π_{ppr}^p for a given simplex σ can be derived

from the concept of a random walk with restarts. Specifically, for the p -order simplicial complex, the corresponding Higher-order Personalized PageRank (HiPPR) Matrix can be calculated as:

$$\Pi_{\text{HiPPR}}^p = \alpha \left(I^p - (1 - \alpha) \tilde{A}^p \right)^{-1}, \quad (3)$$

where I^p is the identity matrix corresponding to the number of simplices of order p and \tilde{A}^p is the normalized adjacency matrix incorporating the higher-order simplicial structure, representing the adjacency matrix between 0-simplices (nodes) and p -simplices (where $p > 0$). This innovative formula extends the basic idea of PPR to account for higher-order relationships within the simplicial complex.

In practical computations, approximate algorithms balance speed and accuracy. Recent key approximate approaches [Yang *et al.*, 2024] include Monte Carlo, Power Iteration, and Forward Push. We chose to improve the PwrPush-SOR method [Chen *et al.*, 2023], which combines Power Iteration and Forward Push, achieving better error and runtime on undirected graphs. It computes PPR vectors for each node v with $O(n \cdot d(v)_{\text{avg}})$ time and $O(n)$ space complexity. The innovative algorithm HiPwrPushSOR used to approximate the higher-order PPR vector is presented in Algorithm 1. The input is a single root node s , and the output is the PPR vector we aim to compute. For SCs with order p , we compute p PPR vector for the root node.

The input includes the SC \mathcal{K} , the source node s and parameters $\alpha, \epsilon, \lambda$ and ω . Where α controls the transition probability, which affects the range of information propagation in the PPR algorithm; ϵ determines the node activation criteria for queue insertion; λ regulates the stopping condition to guarantee computational precision; and $\omega \in (0, 2)$ is the Successive Over-Relaxation (SOR) parameter to control the intensity of the update at each iteration. In this study, we adopt the value recommended for undirected graphs [Chen *et al.*, 2023].

In the new algorithm, we iterate all nodes (0 to $n-1$), treating each as the source node s to calculate its PPR vector, which reflects its relationships with all other nodes. By com-

Algorithm 1 HiPwrPushSOR: Higher-order Personalized PageRank Approximation

Input: Simplicial complex \mathcal{K} , source node $s \in \mathcal{K}$, number of nodes n , number of simplices m

Parameter: Precision ϵ , teleport probability $\alpha \in (0, 1)$, error threshold $\lambda > 0$, relaxation parameter $\omega > 0$

Output: Estimated PPR vector $\hat{\pi}_s \in \mathbb{R}^n$

```

1: Initialize: epochNum  $\leftarrow 8$ , scanThreshold  $\leftarrow \lfloor n/4 \rfloor$ 
2: Set  $\hat{\pi}(s, \sigma) \leftarrow 0$ ,  $r(s, \sigma) \leftarrow 0$  for all  $\sigma \in \mathcal{K}$ ;  $r(s, s) \leftarrow 1$ 
3: Initialize FIFO queue  $Q \leftarrow \emptyset$ ;  $Q.append(s)$ 
4: Set  $r_{\max} \leftarrow \lambda/m$ ;  $r_{\text{sum}} = 1$ 
5: while  $Q \neq \emptyset$  and  $|Q| \leq \text{scanThreshold}$  and  $r_{\text{sum}} > \lambda$  do
6:    $\sigma \leftarrow Q.pop()$ 
7:    $\hat{\pi}(s, \sigma) \leftarrow \hat{\pi}(s, \sigma) + \omega\alpha r(s, \sigma)$ 
8:    $r_{\text{sum}} \leftarrow r_{\text{sum}} - \omega\alpha r(s, \sigma)$ 
9:   for each  $\tau \in N_{\text{out}}(\sigma)$  do
10:     $r(s, \tau) \leftarrow r(s, \tau) + \frac{\omega(1-\alpha)r(s, \sigma)}{d_\sigma}$ 
11:    if  $\tau \notin Q$  then
12:       $Q.append(\tau)$ 
13: if  $r_{\text{sum}} > \lambda$  then
14:   /* Switch to sequential scan */
15:   for  $i = 1$  to epochNum do
16:     $r'_{\max} \leftarrow \lambda^i / \text{epochNum} / m$ 
17:    while  $r_{\text{sum}} > m \cdot r'_{\max}$  do
18:      for each  $\sigma \in \mathcal{K}$  do
19:        if  $|r(s, \sigma)| \geq \epsilon d_\sigma$  then
20:           $\hat{\pi}(s, \sigma) \leftarrow \hat{\pi}(s, \sigma) + \omega\alpha r(s, \sigma)$ 
21:          for each  $\tau \in N_{\text{out}}(\sigma)$  do
22:             $r(s, \tau) \leftarrow r(s, \tau) + \frac{\omega(1-\alpha)r(s, \sigma)}{d_\sigma}$ 
23: return  $\hat{\pi}_s = [\hat{\pi}(s, \sigma)]_{\sigma \in \mathcal{K}}$ 
    
```

putting PPR vectors for every node, we obtain the PPR matrix. After computing the PPR matrix for each order p , Π_{HiPPR}^p is obtained. Moreover, excessive propagation of higher-order information in node classification can cause redundancy and increased computational cost. To address this, we introduce a constraint in HiPPR, modeling nodes propagating to higher-order neighbors and back, thus efficiently controlling higher-order information flow.

4.3 Higher-order Adaptive Spectral Convolution

Spectral graph convolution

Served as the foundation for many spectral GNNs [Bianchi et al., 2021], spectral graph convolution is typically unified under the expression $Z = g(L) \cdot X$, where Z represents the output feature matrix and $g(L)$ is the graph filter applied to the graph Laplacian matrix L . The core of spectral graph convolution lies in the design of the graph filter g , which governs how graph signals are manipulated in the spectral domain.

For a given graph Laplacian matrix $L = U\Lambda U^\top$, the eigenvalue matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$ captures the frequency components of the graph signal, while $U = [u_1, u_2, \dots, u_m]$ represents the eigenvectors corresponding to these frequencies. The graph Fourier transform for a graph signal x is given by $\hat{x} = U^\top x$. The goal of spectral graph convolution is to manipulate the frequency response of the graph signal by

designing an appropriate graph filter $g(L)$. The graph convolution in the spectral domain can then be expressed as:

$$z = g(L)x = Ug(\Lambda)U^\top x. \quad (4)$$

However, computing the spectral decomposition of large-scale networks is computationally prohibitive. To address this challenge, a practical approach is to approximate the graph filter using polynomial expansions, which avoids direct spectral decomposition. Specifically, the graph filter $g(\lambda)$ is often approximated as a truncated polynomial: $g(\lambda) = \sum_{k=0}^K \alpha_k \lambda^k$, where α_k are learnable coefficients and K denotes the highest polynomial order considered. This polynomial approximation allows for efficient spectral convolution, bypassing expensive eigen-decompositions.

Incorporating Higher-order Structures

Expanding upon the standard spectral graph convolution, we introduce higher-order spectral convolutions by integrating SCs. In this context, the graph laplacian is generalized to higher-order laplacians, which model interactions within simplices—structures that generalize edges to higher-dimensional relationships like triangles and tetrahedra. For each order of the SCs, we compute a PPR laplacian matrix, which captures the connectivity relationships at different orders. The convolution operation for each Higher-order Laplacian is defined as:

$$g_p(L_p) = \sum_{k=0}^K \beta_{p,k} L_p^k, \quad (5)$$

where $g_p(L_p)$ represents the graph filter applied to the p -th order Laplacian, p corresponds to the order of the SCs. $\beta_{p,k}$ are learnable coefficients that capture the influence of neighbors at different hops in the higher-order SCs, and k determines the polynomial degree for each PPR laplacian matrix.

The convolution across higher-order graphs allows us to encode richer information from neighbors at varying distances within the graph. For a signal $X \in \mathbb{R}^{n \times d}$ with d characteristics per node, we first extract hidden state features for each node and then uses HiPPR to propagate them:

$$H^0 = f(X), H^k = \tilde{\Pi}_{\text{HiPPR}}^p H^{(k-1)}, \quad (6)$$

$$Z^p = \sum_{k=0}^K \beta_{p,k} H^{p,k}, Y = \rho(\|_{p=1}^P Z^p), \quad (7)$$

where Z^p represents the output combining contributions from the k -th hop, β_k are k learnable coefficients, $\tilde{\Pi}_{\text{HiPPR}}^p$ captures neighborhood information at order p , $\tilde{\Pi}_{\text{HiPPR}}^p$ is the normalised version of Π_{HiPPR}^p , formally defined as $\tilde{\Pi}_{\text{HiPPR}}^p = D^{-1/2} \Pi_{\text{HiPPR}}^p D^{-1/2}$, where D is the diagonal degree matrix. X is the input feature matrix, ρ is the simplified linear functions suggested by [Wang and Zhang, 2022], and $\|$ denotes concatenation.

Higher-order Spectral Convolution Formulation

To implement the process, we refer to the work of [Wang and Zhang, 2022], who simplified the functions ρ into linear forms. Building on this, we integrate our novel design of the PPR operator into Higher-order spectral convolutions, leading to the following formulation for the model output:

$$Y = \left\|_{p=1}^P \left(\sum_{k=0}^K \beta_{p,k} \tilde{\Pi}_{HiPPR}^p X \Theta_p \right) W, \quad (8)$$

where $\beta_{p,k}$, Θ_p and W are learnable parameters, $\|$ denotes concatenation, and P refers to the highest order of the simplicial complex under consideration.

By incorporating higher-order interactions through our proposed PPR-based filter, this framework extends the capabilities of spectral graph convolution, enabling the model to capture complex dependencies and multiscale information from the graph structure. This adaptive spectral convolution method not only enhances the representation of graph signals but also improves the scalability and efficiency of the model.

5 Experiments

We first give the experimental setup, and then compare the performance of our method with baselines on node classification. After that, we analyse the performance of different components of HPGNN and give the parameter analysis. Finally, we provide a visualization task with t-SNE.

5.1 Experimental Setup

Datasets

We adopt seven real world datasets to evaluate the performance of our proposed HPGNN, including two categories:

(1) Homophilic Graphs: Citation graphs Cora and Citeseer [Sen *et al.*, 2008]; Social network Photo [McAuley *et al.*, 2015].

(2) Heterophilic Graphs: Cornell, Texas, and Wisconsin webpage graphs [Pei *et al.*, 2020]; Actor collaboration network Actor [Pei *et al.*, 2020].

We tabulated the basic statistics of the datasets in Table 1. The number of triangles, represents the count of 2-simplex in the graph, where three nodes are mutually connected.

Datasets	#Node	#Edge	#Feature	#Class	#Triang.	#Homo.
Cora	2708	10556	1433	6	1630	0.810
Citeseer	3327	9104	3703	7	1167	0.736
Photo	7650	119081	745	8	717400	0.827
Cornell	183	298	1703	5	59	0.127
Actor	7600	29926	931	5	7121	0.219
Texas	183	309	1703	5	67	0.087
Wisconsin	251	499	1703	5	118	0.192

Table 1: Benchmark datasets properties and statistics.

Baselines

We compare HPGNN with three categories of baselines:

(1) Homophily-aware Models: Classical graph convolutional networks based on homophily assumption: ChebNet [Defferrard *et al.*, 2016], GCN [Xu *et al.*, 2018], GAT [Veličković *et al.*, 2017].

(2) Heterophily-aware Models: APPNP [Gasteiger *et al.*, 2018] mitigates the limitations of local aggregation by employing a fixed personalized pagerank-based propagation scheme, GPRGNN [Chien *et al.*, 2020] generalizes by introducing learnable propagation weights, enabling the model to flexibly capture multi-hop dependencies through adaptive PageRank coefficients, BernNet [He *et al.*, 2021] adjusts node influences using a probabilistic model to handle heterophily, HiGCN [Huang *et al.*, 2024] uses flower-petals Laplacian operators to mitigate heterophily challenges, and PolyGCL [Chen *et al.*, 2024] integrates polynomial filters into GCL to effectively capture both local and global structural patterns, leading to improved performance on graphs with varying degrees of homophily.

(3) Higher-order Models: S2V [Billings *et al.*, 2019] and SNN [Ebli *et al.*, 2020] embeds simplicial complexes to capture higher-order relationships, SGAT and SGAT-EF [Lee *et al.*, 2022] adapts aggregation weights using attention mechanisms for heterophilic graphs.

Implementation Details

We implement our proposed HPGNN with Pytorch framework on four NVIDIA RTX 3090 GPUs. Following [He *et al.*, 2021], For all datasets, we randomly split nodes into 60%, 20%, and 20% for training, validation, and testing, all methods share the same 10 random splits. We train the proposed HiGCN model with the learning rate $lr \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$ and the weight decay $wd \in \{0.0, 0.0001, 0.001, 0.005, 0.1\}$. For all methods, we run 100 times with the same partition and report the average and standard deviation of accuracy values.

5.2 Main Experiment Results

Tables 2 shows the comparison of our proposed HPGNN with SOTAs for node classification task on seven real-world network datasets. It can be concluded that: (1) On heterophilic graphs Cornell, Actor, Texas and Wisconsin, HPGNN shows substantial improvement, particularly in datasets with more complex relationships. This suggests its effectiveness in capturing multiscale node dependencies, which is crucial for heterophilic graph structures. (2) HPGNN maintaining competitive performance on homophilic graphs Cora, Citeseer and Photo, showcasing its versatility and robustness across different types of graph structures, a point further supported by the average standard deviation in the table. (3) Compared to higher-order simplicial complex neural networks, our proposed algorithm shows an overwhelming advantage, mainly because it combines multi-scale features and adaptive weights to avoid over-smoothing. (4) Though HPGNN performs competitively on Actor and Texas, exceeding the third-best method by at least 1.49 and 1.65 percentage points, respectively. (5) Notably, the top two results across dataset,

Method	Cora	Citeseer	Photo	Cornell	Actor	Texas	Wisconsin
ChebNet	87.04 ± 0.10	76.53 ± 0.10	90.75 ± 0.14	83.40 ± 0.47	39.10 ± 0.87	85.57 ± 0.36	92.0 ± 0.30
GCN	88.29 ± 0.14	78.8 ± 0.12	81.92 ± 0.25	63.62 ± 0.81	34.2 ± 0.10	80.49 ± 0.54	70.38 ± 0.62
GAT	88.10 ± 0.13	78.44 ± 0.13	74.61 ± 0.90	68.72 ± 0.37	30.50 ± 0.45	80.33 ± 0.32	71.38 ± 0.40
APNP	87.27 ± 0.17	78.42 ± 0.19	80.44 ± 0.28	77.66 ± 0.51	37.46 ± 0.93	82.25 ± 0.44	82.38 ± 0.39
GPRGNN	88.88 ± 0.15	78.43 ± 0.19	92.36 ± 0.11	87.79 ± 0.43	38.75 ± 0.74	93.11 ± 0.26	94.13 ± 0.32
BernNet	87.95 ± 0.16	78.58 ± 0.15	93.54 ± 0.04	85.53 ± 0.39	38.11 ± 0.10	89.83 ± 0.38	<u>94.50 ± 0.28</u>
HiGCN	89.24 ± 0.91	78.70 ± 0.78	94.13 ± 0.88	84.25 ± 0.44	39.41 ± 0.12	89.18 ± 0.46	<u>93.13 ± 0.26</u>
POLYGCL	<u>87.57 ± 0.62</u>	<u>79.81 ± 0.85</u>	<u>93.12 ± 0.23</u>	82.62 ± 0.31	41.15 ± 0.88	88.03 ± 1.80	85.50 ± 1.88
S2V	80.15 ± 0.88	78.21 ± 0.34	84.33 ± 0.19	77.63 ± 0.21	39.22 ± 0.50	82.12 ± 0.23	83.48 ± 0.89
SNN	87.13 ± 1.02	79.87 ± 0.68	88.27 ± 0.74	70.57 ± 0.78	30.59 ± 0.23	75.16 ± 0.96	61.93 ± 0.83
SGAT	77.49 ± 0.79	78.93 ± 0.63	N/A	85.46 ± 0.36	36.71 ± 0.49	89.83 ± 0.66	81.47 ± 0.64
SGAT-EF	78.12 ± 0.85	79.16 ± 0.72	N/A	85.21 ± 0.62	37.33 ± 0.58	89.67 ± 0.74	81.59 ± 0.81
HPGNN(Ours)	89.96 ± 0.13	79.89 ± 0.12	95.19 ± 0.14	88.30 ± 0.42	<u>40.89 ± 0.10</u>	<u>91.48 ± 0.27</u>	94.75 ± 0.33

Table 2: Node classification results on empirical benchmark networks: mean accuracy (%) ± 95% confidence interval. The best results are in **bold**, while the second-best ones are underlined.

aside from ours, come from four different algorithms, demonstrating the stability and consistency of our method, highlighting the impact of higher order techniques in achieving this stability.

5.3 Ablation Study

As shown in Figure 2, ablation studies were performed on four datasets: two homophilic graphs (Cora and Citeseer) and two heterophilic graphs (Actor and Wisconsin). To validate the effectiveness of each component in our proposed model, we conducted several ablation experiments comparing our model with different variants: 1) HPGNN(w/o High): This variant excludes the higher-order structure, using only pairwise PPR weights and convolution for signal filtering between nodes. For this variant, we set the model hyperparameter order to 1 in different datasets. It shows that removing higher-order structures degrades performance, especially on Wisconsin and Actor, highlighting their role in complex heterophilic graph interactions. 2) HPGNN(w/o High and PPR): This variant removes the PPR mechanism from the pairwise graph, replacing the PPR matrix Π with $\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ as suggested in [Wu *et al.*, 2019]. It can be analyzed that excluding PPR causes a notable drop in performance, particularly on Wisconsin and Actor, underlining its importance in noise filtering and long-range dependency.

In summary, HPGNN outperforms both variants, demonstrating the effectiveness of the proposed modules. In heterophilic graphs, performance degradation is more significant when higher-order and PPR modules are removed, emphasizing their importance in such environments. Combining both higher-order structures and PPR yields the best results, demonstrating their complementary roles in improving performance.

5.4 Parameter Analysis

We estimate the sensitivity of three important hyperparameters, that is, the order p , the teleportation parameter α and the convolution layer k .

Analysis of Order p . The order p represents the order in higher-order networks SCs. Table 3) shows the results of

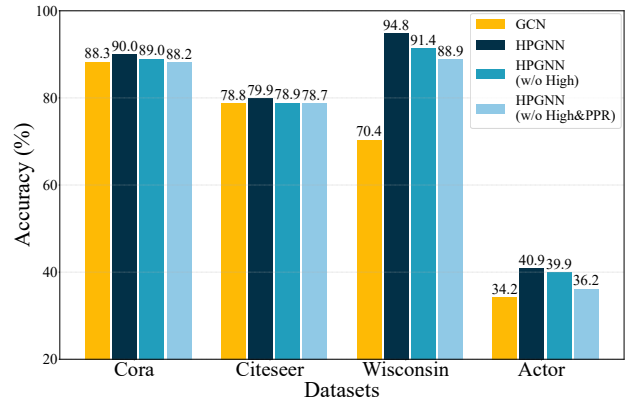


Figure 2: The impact of Higher-order and PPR Modules

order	p=1	p=2	p=3	p=4	p=5
Cora	88.96	89.96	89.75	89.45	N/A
Citeseer	78.90	79.89	79.93	79.84	79.84
Actor	39.90	40.90	40.56	40.57	40.49
Cornell	87.40	88.30	88.05	N/A	N/A

Table 3: Node classification results of different orders

order p on node classification across Cora, Citeseer, Actor and Cornell four datasets. As observed in the table and also supported by findings in the previous literature [Huang *et al.*, 2024], second order generally yields better performance. Hence, we consistently adopt second-order information in all our experiments to ensure fair and consistent evaluation.

Analysis of Teleportation Parameter α . The teleportation parameter α represents the restart probability in HiPPR. All lines show relatively stable accuracy across the tested range, with minor fluctuations. The results on Wisconsin and Cora exhibit slight peaks near $\alpha = 0.15$, while Citeseer remains consistently flat. In reference to common practice in most papers, we selected $\alpha = 0.15$ as the optimal value.

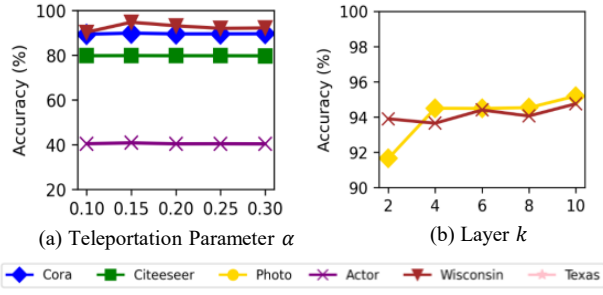


Figure 3: Visualization of sensitivity analysis.

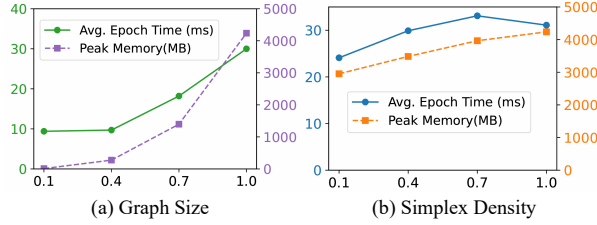


Figure 4: Training Time vs. Peak Memory Usage.

Analysis of Layer k . The layer k represents the layer in higher-order adaptive spectral graph convolutions. We investigated the impact of k in HPGNN, evaluating its influence on model performance across Photo and Actor datasets. As shows in Table 3(b), both lines show a general upward trend as k increases, with minor fluctuations. we selected $k = 10$ as the optimal value, as it yields the highest accuracy while maintaining stability across the tested range.

5.5 Computationally Efficient

Table 4 illustrates that our model significantly outperforms Hodge-based SNN in terms of speed while maintaining performance comparable to pairwise methods. The runtime and memory usage curves for the Actor dataset, shown in Figure 4 highlight distinct trends. Specifically, it demonstrates that as the graph size increases, both the training time per epoch and peak memory usage exhibit exponential growth. In contrast, with an increase in simplex density, the growth in training time per epoch and peak memory usage follows a more linear trend.

5.6 Visualization

To evaluate the clustering behavior and effectiveness of our proposed model, HPGNN, we present t-SNE visualizations for the Cora and Wisconsin datasets in Figure 5. These

Dataset	HPGNN	SNN	HIGCN	BernNet	GPRGNN
Cora	13.7/4.92	32.2/10.03	14.4/3.13	24.4/8.25	10.1/3.16
Citeeseer	9.6/3.5	27.7/9.27	12.2/2.75	25.6/8.68	9.3/3.23
Cornell	8.6/2.02	28.1/7.33	10/2.18	23.7/7.52	9.7/2.61
Wisconsin	9.3/2.1	24.4/6.65	8.7/2	22.5/7.32	10.3/2.73

Table 4: Efficiency on node classification experiments: Average running time per epoch(ms)/ average total running time(s).

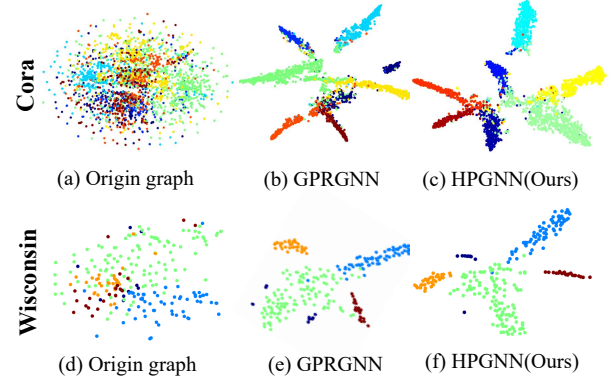


Figure 5: Structure visualization

plots show how well the models separate classes in a low-dimensional space, with edges omitted for clarity.

As shows Figure 5(c), for the Cora dataset, HPGNN clusters nodes more clearly than GPRGNN, indicating that its integration of high-order information and PPR mechanisms improves the separation of classes in homophilic graphs. Figure 5(e) shows that in the Wisconsin dataset, HPGNN also outperforms other models in clustering heterophilic graphs, with better class separation, showing its robustness in handling complex structures.

6 Conclusion

In this paper, we propose HPGNN, a novel graph neural network model that combines higher-order node relationships with the personalized PageRank (PPR) mechanism to improve node classification, especially in complex graph structures. Unlike previous methods focusing on pairwise relationships, our approach captures multilevel interactions, crucial for heterophilic environments.

Experiments on seven real-world datasets show that HPGNN outperforms state-of-the-art methods in 5 out of 7 cases, especially on heterophilic graphs. Its ability to handle noisy and complex relationships sets it apart, offering a more robust solution for graph representation learning. Further optimization is needed for simpler or sparse higher-order graphs to improve performance across all scenarios.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62472305 and No.62272338), the Key R&D Project in Hainan Province, China (No.ZDYF2024SHFZ051), and the Hainan Tropical Ocean Institute Yazhou Bay Innovation Research Institute Major Science and Technology Plan Project (No.2023CXYZD001).

Contribution Statement

Yumeng Wang and Zengyi Wo contributed equally to this work. Minglai Shao is the corresponding author.

References

- [Antelmi *et al.*, 2023] Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. A survey on hypergraph representation learning. *ACM Computing Surveys*, 56(1):1–38, 2023.
- [Battiston *et al.*, 2020] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Pataña, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics reports*, 874:1–92, 2020.
- [Benson *et al.*, 2016] Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.
- [Bianchi *et al.*, 2021] Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3496–3507, 2021.
- [Billings *et al.*, 2019] Jacob Charles Wright Billings, Mirko Hu, Giulia Lerda, Alexey N Medvedev, Francesco Mottes, Adrian Onicas, Andrea Santoro, and Giovanni Petri. Simplex2vec embeddings for community detection in simplicial complexes. *arXiv preprint arXiv:1906.09068*, 2019.
- [Chen *et al.*, 2022] Yuzhou Chen, Yulia R Gel, and H Vincent Poor. Bscnets: Block simplicial complex neural networks. In *Proceedings of the aaai conference on artificial intelligence*, volume 36, pages 6333–6341, 2022.
- [Chen *et al.*, 2023] Zhen Chen, Xingzhi Guo, Baojian Zhou, Deqing Yang, and Steven Skiena. Accelerating personalized pagerank vector computation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 262–273, 2023.
- [Chen *et al.*, 2024] Jingyu Chen, Runlin Lei, and Zhewei Wei. In *The Twelfth International Conference on Learning Representations*, 2024.
- [Chien *et al.*, 2020] Eli Chien, Jianhao Peng, Pan Li, and Olga Milenkovic. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*, 2020.
- [Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.
- [Ebli *et al.*, 2020] Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. *arXiv preprint arXiv:2010.03633*, 2020.
- [Fu *et al.*, 2024] Yuanyuan Fu, Decui Liang, and Zeshui Xu. Higher-order interaction of stability simplicial complex driven group consensus reaching in social network. *Information Fusion*, 103:102095, 2024.
- [Gasteiger *et al.*, 2018] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.
- [Hajij *et al.*, 2021] Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Vasileios Maroulas, and Xuanning Cai. Simplicial complex representation learning. *arXiv preprint arXiv:2103.04046*, 2021.
- [Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.
- [Hamilton, 2020] William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.
- [He *et al.*, 2021] Mingguo He, Zhewei Wei, Hongteng Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34:14239–14251, 2021.
- [Huang *et al.*, 2024] Yiming Huang, Yujie Zeng, Qiang Wu, and Linyuan Lü. Higher-order graph convolutional network with flower-petals laplacians on simplicial complexes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12653–12661, 2024.
- [Iacopini *et al.*, 2019] Iacopo Iacopini, Giovanni Petri, Alain Barrat, and Vito Latora. Simplicial models of social contagion. *Nature communications*, 10(1):2485, 2019.
- [Jack *et al.*, 2023] Dominic Jack, Sarah Erfani, Jeffrey Chan, Sutharshan Rajasegarar, and Christopher Leckie. It’s pagerank all the way down: Simplifying deep graph networks. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 172–180. SIAM, 2023.
- [Jin *et al.*, 2021] Di Jin, Zhizhi Yu, Cuiying Huo, Rui Wang, Xiao Wang, Dongxiao He, and Jiawei Han. Universal graph convolutional networks. *Advances in Neural Information Processing Systems*, 34:10654–10664, 2021.
- [Lee *et al.*, 2022] See Hian Lee, Feng Ji, and Wee Peng Tay. Sgat: Simplicial graph attention network. *arXiv preprint arXiv:2207.11761*, 2022.
- [Majhi *et al.*, 2022] Soumen Majhi, Matjaž Perc, and Dibakar Ghosh. Dynamics on higher-order networks: A review. *Journal of the Royal Society Interface*, 19(188):20220043, 2022.
- [McAuley *et al.*, 2015] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.
- [McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.
- [Page, 1999] Lawrence Page. The pagerank citation ranking: Bringing order to the web. Technical report, Technical Report, 1999.
- [Pei *et al.*, 2020] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Ge-

- ometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.
- [Qiu et al., 2024] Chenyang Qiu, Guoshun Nan, Tianyu Xiong, Wendi Deng, Di Wang, Zhiyang Teng, Lijuan Sun, Qimei Cui, and Xiaofeng Tao. Refining latent homophilic structures over heterophilic graphs for robust graph convolution networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8930–8938, 2024.
- [Roddenberry et al., 2021] T Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pages 9020–9029. PMLR, 2021.
- [Schaub et al., 2020] Michael T Schaub, Austin R Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62(2):353–391, 2020.
- [Schaub et al., 2021] Michael T Schaub, Yu Zhu, Jean-Baptiste Seby, T Mitchell Roddenberry, and Santiago Segarra. Signal processing on higher-order networks: Livin’ on the edge... and beyond. *Signal Processing*, 187:108149, 2021.
- [Sen et al., 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.
- [Spanier and Spanier, 1989] Edwin H Spanier and Edwin Henry Spanier. *Algebraic topology*. Springer Science & Business Media, 1989.
- [Veličković et al., 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [Wang and Zhang, 2022] Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International conference on machine learning*, pages 23341–23362. PMLR, 2022.
- [Wo et al., 2024] Zengyi Wo, Minglai Shao, Wenjun Wang, Xuan Guo, and Lu Lin. Graph contrastive learning via interventional view generation. In *Proceedings of the ACM Web Conference 2024*, pages 1024–1034, 2024.
- [Wu et al., 2019] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.
- [Wu et al., 2023a] Hanrui Wu, Andy Yip, Jinyi Long, Jia Zhang, and Michael K Ng. Simplicial complex neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [Wu et al., 2023b] Lirong Wu, Cheng Tan, Zihan Liu, Zhangyang Gao, Haitao Lin, and Stan Z Li. Learning to augment graph structure for both homophily and heterophily graphs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–18. Springer, 2023.
- [Xu et al., 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- [Yang et al., 2022] Maosheng Yang, Elvin Isufi, and Geert Leus. Simplicial convolutional neural networks. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8847–8851. IEEE, 2022.
- [Yang et al., 2024] Mingji Yang, Hanzhi Wang, Zhewei Wei, Sibow Wang, and Ji-Rong Wen. Efficient algorithms for personalized pagerank computation: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [Zhou et al., 2024] Cai Zhou, Xiyuan Wang, and Muhan Zhang. Facilitating graph neural networks with random walk on simplicial complexes. *Advances in Neural Information Processing Systems*, 36, 2024.
- [Zhu et al., 2022] Jiong Zhu, Junchen Jin, Donald Loveland, Michael T Schaub, and Danaï Koutra. How does heterophily impact the robustness of graph neural networks? theoretical connections and practical implications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2637–2647, 2022.