HFGCN: Hybrid Filter Graph Convolutional Network for Heterophilic Graphs

Zitong Bo[1,2], Chaoyi Yang[1,2], Yilin Li[1,2], Kaiyue Xu[3], Ying Qiao[1(✉)], Chang Leng[1], and Hongan Wang[1]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China
{zitong201,qiaoying}@iscas.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Agricultural Bank of China, Tianjin, China

Abstract. Graph Convolutional Networks (GCNs) are pivotal in analyzing graph data. However, as graph complexity increases, heterophily challenges the traditional GCNs that rely on homophily assumptions. These challenges have elicited various mitigation attempts, which, however, have only achieved partial success. They either inadequately harness the structural intricacies of the data or incorporate irrelevant information, thereby undermining their efficacy in complex heterophilic graphs. To address this, we introduce the Hybrid Filtering Graph Convolutional Network (HFGCN), an innovative framework integrating three specialized filtering mechanisms for spectral domain node aggregation. Inspired by the Power-law transformation, HFGCN employs low-pass filters for homophilic nodes, high-pass filters for heterophilic nodes, and a self-aggregation filter for nodes reliant on their information. An attention mechanism further refines node interaction based on attributes. Our evaluations across seven diverse datasets demonstrate HFGCN's superior adaptability and performance, surpassing state-of-the-art models in handling both homophilic and heterophilic graphs.

Keywords: Graph Convolution Network · Heterophilic Graph · Hybrid Filter

1 Introduction

Graph Convolutional Networks (GCNs) have significantly advanced graph data analysis, enabling effective feature computation by aggregating a node's features with those of its neighbors [12, 18, 22]. Pioneering models such as GAT [16], GraphSAGE [8], and GraphHeat [19], have shown enhanced performance in various tasks. These models are based on the homophily principle, where it is assumed that connected nodes are likely to belong to the same class or share similar attributes [23].

However, many real-world graphs exhibit heterophily, where connected nodes are of different classes or possess different attributes, challenging the extraction and utilization of structural information for tasks such as information retrieval. This deviation from

Z. Bo, C. Yang and Y. Qiao—These authors contributed equally to this work.

D.-S. Huang et al. (Eds.): ICIC 2024, LNAI 14875, pp. 439–450, 2024.
https://doi.org/10.1007/978-981-97-5663-6_37

homophily affects the performance of traditional GCNs designed under homophilic assumptions [14, 23]. In response, models like Geom-GCN [14], MixHop [1], H2GCN [23], GPR-GNN [5], and FAGCN [2] have been developed to improve GCN performance across heterophilic contexts.

While this approach enables the aggregation of distant, similar nodes, it may also inadvertently introduce irrelevant information. Given the prevalence of both homophilic and heterophilic nodes in complex graphs, an ideal GCN model should efficiently utilize both graph structure and node attribute information for customized node aggregation. The spectral domain offers a promising avenue for this, providing a global perspective on the differences in node connectivity relationships through the decomposition of the normalized Laplacian matrix, which can reveal implicit structural information [7]. This paper introduces the Hybrid Filter-based Graph Convolutional Network (HFGCN), a novel model that redefines neighbor identification from a spectral domain perspective. Inspired by the Power-law transformation [4], HFGCN utilizes the eigenvectors of the normalized Laplacian matrix to develop foundational filters that enrich structural information and enable adaptive node aggregation. An attention-like mechanism [15] further enhances the model by discerning the significance of neighbors based on node attributes, significantly improving adaptability and performance across different graph structures. The main contributions are as follows:

- **Hybrid Filter:** Combines low-pass and high-pass filters, uniquely catering to nodes based on their homophilic or heterophilic nature, with a self-aggregating filter for nodes reliant mainly on their information.
- **Attribute Aggregation:** Integrates an attribute weight assignment matrix C, optimizing both structure-based and attribute-based information processing, ensuring peak performance across various node homophily levels.
- **Evaluate and Visualize:** Conducts extensive testing on seven real-world networks, demonstrating the model's effectiveness and adaptability, and highlighting clear advantages over existing approaches.

The remainder of this paper is organized as follows: Sect. 2 provides a review of the literature related to our work. Section 3 introduces preliminary concepts. Section 4 details the methodology of the HFGCN model; Sect. 5 discusses the experimental setup and results; and Sect. 6 draws conclusions.

2 Related Work

2.1 Beyond Homophily in GNN

To address the limitations of homophily, several models have been developed for heterophilic contexts, categorized by their strategies: Geom-GCN alters graph structures to connect similar nodes spatially [14]. MixHop and H2GCN mix features from multi-order neighborhoods to enrich node representations [1, 23]. GPR-GNN and FAGCN introduce negative weights for feature differentiation between node classes [2, 5]. These models enhance spatial domain insights but often struggle to fully leverage structural information or exclude irrelevant data. Our approach seeks to address these gaps from a spectral domain perspective.

2.2 Spectral-Based GNN

Spectral-based GNNs, initiated by Bruna et al. with Spectral CNN, faced challenges in large graph scalability due to computational limits [3]. ChebNet improved efficiency through Chebyshev polynomial-based convolutions [6], leading to the development of GCN, which facilitated a clearer connection to spatial interpretations [10]. Further advancements by GraphHeat and GWNN optimized spectral analysis using heat kernel and wavelet basis transformations, respectively [19]. Our work builds on these foundations, exploring new directions in spectral-based GNNs, particularly through the application of wavelet transforms to address the nuanced challenges presented by both homophilic and heterophilic graphs [21].

3 Preliminary

3.1 Graph Definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A, X, Y)$ denote an undirected graph, where $\mathcal{V} = \{v_i | i = 1, 2, \ldots, n\}$ is the set of nodes, $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V} and v_i, v_j are connected\}$. The adjacency matrix is defined as $A \in R^{n \times n} where A_{ij} = A_{ji}$, and $A_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, otherwise, $A_{ij} = 0$. $X \in R^{n \times c}$ is a set of node attributes, where c represents the number of attributes. $Y = \{y_i | i = 1, 2, \cdots, n\}$ is the node label set and $\forall y_i \in Y$, $y_i \in \{0, \cdots, k-1\}$, where k is the number of categories. Each node has only one corresponding label. Laplace matrix is defined as $L = I - D^{-1/2} A D^{-1/2}$, where I is the identity matrix.

The homophily ratio is a measure indicating the extent to which nodes of the same class are connected within $\mathcal{G}$. The graph-level homophily ratio is defined as:

$$h = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{n} \frac{\left| \{v_j | v_j \in \mathcal{N}_i \bigwedge y_i = y_j\} \right|}{\left| \mathcal{N}_i \right|} \tag{1}$$

where $\mathcal{N}_i$ denotes the set of neighbors of node i. A graph is generally considered homophilic if $h > 0.5$.

3.2 Graph Fourier Transform

The normalized Laplacian matrix L is a symmetric matrix, so its spectral decomposition can be written as $L = U \Lambda U^\top$, where $U = (u_1, u_2, \cdots, u_n)$, contains n orthogonal eigenvectors, and $\Lambda = diag(\{\lambda_i\}_{i=1}^n)$ is a diagonal matrix, where $\lambda_i \in [0, 2]$. Using the eigenvectors as a basis, the graph Fourier transform of a signal $x \in R^n$ on $\mathcal{G}$ is defined as $\hat{x} = U^\top x$, and its inverse as $x = U\hat{x}$. The spectral convolution of a signal x with a filter $g_\theta = diag(\theta)$ on the graph is:

$$g_\theta * x = U g_\theta U^\top x \tag{2}$$

where g_θ is often expressed as a function of Λ, e.g., in ChebNet, $g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k$, and in GCN, a simplified form $g_\theta(\Lambda) = \theta(I - \Lambda)$ is used.

4 The Proposed Model - HFGCN

This section outlines the motivation behind the HFGCN, details the implementation of the hybrid filter and attribute aggregation mechanism, and presents the complete architecture of the model.

4.1 Motivation

The inception of HFGCN is inspired by an analogous principle from the realm of image processing, specifically the power-law transformation, which is a cornerstone technique for contrast enhancement. In image processing, the power-law transformation is employed to dynamically adjust the brightness of an image, effectively mapping a constrained range of input intensities to a broader spectrum of output values. The basic form of power-law transformation is [4]:

$$s = c \cdot r^{\gamma} \tag{3}$$

where r and s are the input and output intensities, respectively; c and γ are positive constants. For lower intensity values, it expands the range, thereby darkening darker regions of the image. Conversely, for higher intensities, it compresses the range, thus brightening the brighter areas[11]. This selective enhancement or suppression across the intensity spectrum significantly improves the overall contrast of the image, making subtle details more discernible.

Drawing a parallel from this technique to the domain of graph networks, we observe a similar need for nuanced differentiation among nodes based on their connectivity and attributes. Homophily, where connected nodes exhibit similar traits or classifications, benefits from aggregation techniques that enhance these similarities. Conversely, heterophily involves connections between dissimilar nodes, requiring strategies that accentuate these differences to enhance the graph's overall informational contrast. To navigate this dichotomy, our GCN model employs a hybrid filtering approach, analogous to using multiple gamma transformations tailored to specific image regions, which includes:

- A **low-pass filter** that smooths connected nodes' features to enhance homophilic interactions, similar to how darkening lower intensities in images increases uniformity.
- A **high-pass filter** that accentuates the differences in heterophilic nodes, analogous to brightening higher intensities in images to highlight disparities.
- A **self-aggregating filter** designed to preserve the intrinsic characteristics of nodes that rely minimally on external data, maintaining their original representations.

The architecture of HFGCN is visualized in Fig. 1, highlighting the strategic integration of the three filters: low-pass, high-pass, and self-aggregating. To make the model better adaptable, our filter contains two parts, the structure weight matrix and the attribute weight matrix, respectively. Each filter comprises a structural weight assignment matrix S, derived from the spectral domain to reflect graph topology, and an attribute weight assignment matrix C, learned from node attributes. This dual-matrix setup ensures a comprehensive consideration of both structural and attribute data, enhancing the model's ability to aggregate information effectively.

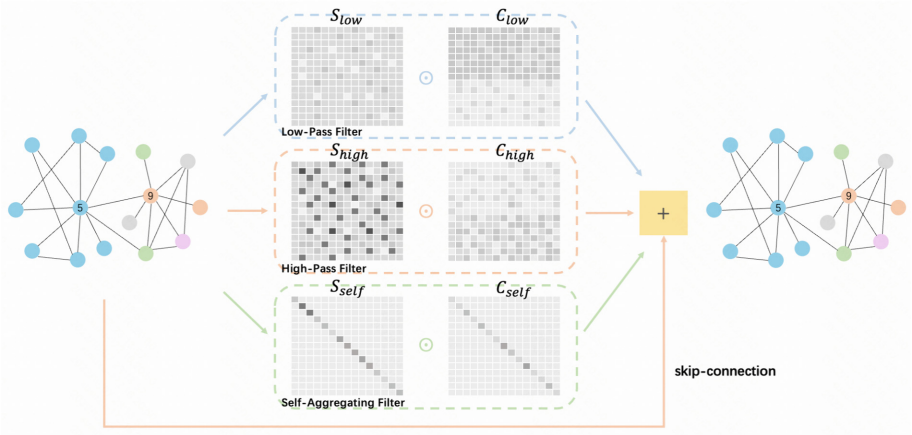


Fig. 1. The architecture of HFGCN.

4.2 Hybrid Filter

In the graph Fourier transform, given that U is orthogonal and g_θ typically a diagonal matrix, the spectral convolution in (2) can be expanded as:

$$g_\theta * x = U g_\theta U^\top x = \sum_{i=1}^{n} \alpha_i u_i u_i^\top x \tag{4}$$

Here, $u_i u_i^\top$ acts as a set of basic filters, and α_i is the corresponding coefficient. This representation allows us to interpret the spectral domain's eigenvalues as indicators of graph signal smoothness, akin to frequencies, with lower eigenvalues indicating smoother changes (low frequencies) and higher eigenvalues denoting more significant differences (high frequencies) across adjacent nodes. Given the relationship:

$$u_i^\top L u_i = \lambda_i = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} \left(u_i - u_j\right)^2 \tag{5}$$

we deduce that basic filters associated with smaller eigenvalues function as low-pass filters, enhancing homophily nodes' similarity, whereas those linked to larger eigenvalues serve as high-pass filters, suitable for emphasizing heterophily nodes' dissimilarity.

By leveraging the normalized Laplacian matrix, $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^\top$, and redefining $D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U(I - \Lambda) U^\top$. Let $\lambda_A = 1 - \lambda$, we introduce λ_A. The frequency response functions for λ_A are then defined to obtain the structural weight distribution matrix S for both filters, inspired by the power-law transformation.

Low-Pass Filter. The design of the low-pass filter is rooted in the principle of promoting signal smoothness across the graph, particularly beneficial for homophilic regions where nodes of similar classes or characteristics are connected. The frequency response function for the low-pass filter is carefully chosen to enhance the weights of signals associated with lower eigenvalues (λ_A), thereby encouraging the propagation of similar features

among closely connected nodes. This approach is encapsulated in the function:

$$f(\lambda_A) = \begin{cases} \lambda_A^2 \lambda_A \geq 0 \\ -\gamma\sqrt{-\lambda_A}\lambda_A < 0 \end{cases} \tag{4}$$

where γ acts as a hyperparameter.

High-pass Filter. Conversely, the high-pass filter is engineered to accentuate the differences in signals between nodes, a property that is particularly useful for delineating heterophilic relationships within the graph. By emphasizing the higher eigenvalues (λ_A), this filter aims to highlight the discrepancies in node features, thereby facilitating the identification and preservation of boundaries between dissimilar node classes. The frequency response function is given by:

$$f(\lambda_A) = \begin{cases} \gamma\sqrt{\lambda_A}\lambda_A \geq 0 \\ -\lambda_A^2 \lambda_A < 0 \end{cases} \tag{5}$$

Self-aggregating Filter. To accommodate nodes requiring minimal neighbor information aggregation, a self-aggregating filter is introduced. This filter aims to preserve the previous layer's node representation. The structural weight assignment matrix for this filter is a diagonal matrix, derived from the normalized adjacency matrix $\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$, where $\tilde{A}$ is the adjacency matrix with the self-loop added, and $\tilde{D}$ is the corresponding degree matrix. Therefore, the structure weight distribution matrix can be written as:

$$S = U\,\mathrm{diag}\big(\{f\left(\lambda_{A_i}\right)\}_{i=1}^{n}\big)U^{\top} \tag{6}$$

4.3 Attribute Aggregation

In the HFGCN model, the attribute weight distribution matrix C is expertly designed to complement the structural weight matrix S, enhancing the model's capability to integrate and analyze information based on both node connectivity and attributes. The construction of C employs an attention-like mechanism[15], which quantifies the relative significance between nodes as follows:

$$C_{ij} = sigmoid < \boldsymbol{a}, X_i||X_j > \tag{7}$$

where $\boldsymbol{a}$ is a learnable parameter vector. And $X_i||X_j$ denotes concatenation of the attribute vectors X_i and X_j of nodes i and j, respectively. The sigmoid function ensures that the attention coefficients are bounded between 0 and 1, representing the strength of influence between nodes.

To reduce noise from less relevant or distant nodes, the model confines the neighborhood scope to the third order, thus enhancing computational effciency and ensuring precise interpretation. Differing from standard attention models, the coefficients $C_{i,j}$ are not normalized across a node's neighbors. This design choice is deliberate, allowing the raw sigmoid values to reflect the absolute importance of each connection, free from the constraints of a normalized distribution. The fusion of S and C to form each filter F

is achieved through an element-wise multiplication $\odot$, symbolizing the intersection of structural connectivity and attribute affinity:

$$F = S \odot C \tag{8}$$

This mechanism ensures that node representations are profoundly influenced by both graph structure and node-specific attributes, with variations in C across different nodes and filters illustrating the variable significance of attributes. Nodes are able to adaptively prioritize different filters based on their unique characteristics. Additionally, a skip-connection mechanism [20] is incorporated to promote model stability and smooth learning, which not only carries forward output from the previous layer but also the initial node representations, enhancing information continuity and gradient flow across network layers.

4.4 Forward Propagation

The forward propagation mechanism of the HFGCN is formulated as follows:

$$\begin{aligned} X^{(0)} &= \Phi(W_1 X), \\ X^{(l)} &= \epsilon X^{(0)} + \left(F_{\text{low}} + F_{\text{high}} + F_{\text{self}}\right) X^{(l-1)}, \\ X^{(\text{out})} &= \text{softmax}\left(\Phi\left(W_2 X^{(L)}\right)\right), \end{aligned} \tag{9}$$

where:

- $X^{(0)}$ represents the initial node attributes transformed into a lower-dimensional space through a Multi-Layer Perceptron (MLP) layer $\Phi(W_1 X)$, capturing essential features for node classification.
- $X^{(l)}$ denotes the node representation at each hidden layer (l), updated through a hybrid filtering approach that incorporates low-pass (F_{low}), high-pass (F_{high}), and self-aggregating (F_{self}) filters to enrich the representation by blending structural and attribute information.
- $X^{(\text{out})}$ is the final node representation projected onto a k-dimensional space corresponding to the number of node categories, with the softmax function outputting the probability distribution over categories.

To ensure efficiency, the structure weight assignment matrix is precomputed to minimize computational overhead associated with decomposing L. With the requirement for adjacency matrix decomposition, HFGCN exhibits a time complexity of $O\left(n^2\right)$ and a space complexity of $O\left(n^2\right)$, where n is the number of nodes. This computational profile is on par with state-of-the-art models such as H2GCN [23], offering competitive performance while introducing advanced filtering mechanisms tailored to the complex dynamics of graph structures.

5 Experiments

5.1 Datasets

We selected seven commonly used datasets to evaluate our approach, with their statistics presented in Table 1.

Table 1. Datasets statistics

Dataset	Nodes	Edges	Features	Classes	Homophily
Cora	2708	5429	1433	7	0.81
Citeseer	3327	4732	3703	6	0.74
Texas	183	309	1703	5	0.11
Wisconsin	251	499	1703	5	0.21
Cornell	183	295	1703	5	0.30
Chameleon	2277	36101	2325	5	0.23
Squirrel	5201	217073	2089	5	0.22

Citation network datasets such as Cora and Citeseer, represent typical **homophilic** graphs where nodes are papers and edges represent citations. We adopt a division of 48% training, 32% validation, and 20% testing. Datasets from WebKB (e.g., Texas, Wisconsin, Cornell) and Wikipedia (e.g., Chameleon, Squirrel) were selected to represent **heterophilic** graphs. We divide these datasets in a 60%, 20%, and 20% ratio for training, validation, and testing, respectively.

5.2 Experimental Settings

Baseline. To evaluate the effectiveness of our method, we compare HFGCN with nine baseline models: MLP, GCN [100], GAT [16], GraphHeat [19], Geom-GCN [14], GPR-GNN [5], FAGCN [2], H2GCN [23], and HOG-GCN [17]. The latter five models are tailored for heterophilic graph, whereas GCN, GAT, and GraphHeat are designed under the homophily assumption. All models, except MLP, are configured according to their original settings for optimal performance.

Hyperparameters. For all experiments, we run 1500 epochs with an early stopping patience set at 100. The learning rate is fixed at 0.01, and the hidden unit size is set to 64. For citation datasets, we use 2 layers, a dropout rate of 0.6, weight decay of 1e−3, $\epsilon = 0.4$, and $\gamma = 0.4$ for Cora, and $\gamma = 0.3$ for Citeseer. For WebKB datasets, we adjust the parameters specifically for each dataset: Texas, Wisconsin, and Cornell have their configurations set to optimize performance. Given the challenging nature of heterophilic graphs, these datasets employ a division ratio of 60% training, 20% validation, and 20% testing. For Wikipedia datasets, which exhibit poor performance with existing methods, we preprocess features using t-SNE [13], choosing $D^{-1/2}AD^{-1/2}X$ as model input for better representation. HFGCN configurations are adapted accordingly, with specific layer, dropout, γ, and ϵ values set for Chameleon and Squirrel. All models are optimized using the Adam optimizer [11], selecting the best-performing model on the validation set for testing. Results are averaged over 10 runs, reporting both mean and standard deviation.

5.3 Results

We evaluate the effectiveness of HFGCN under the node semi-supervised classification task. Similar to previous methods, we use classification accuracy as the evaluation criterion. Experimental results of all methods under different datasets are shown in Table 2. Our method performs well on both homophilic and heterophilic graphs with good robustness. HFGCN has advantages over baselines on five real networks, and the performance is second only to H2GCN on Wisconsin and Chameleon datasets. Although the accuracy rate in Wisconsin is 2% lower than that of H2GCN, since the network has only 251 nodes, there are only 50 nodes in the test set, the difference of 2% is only one node.

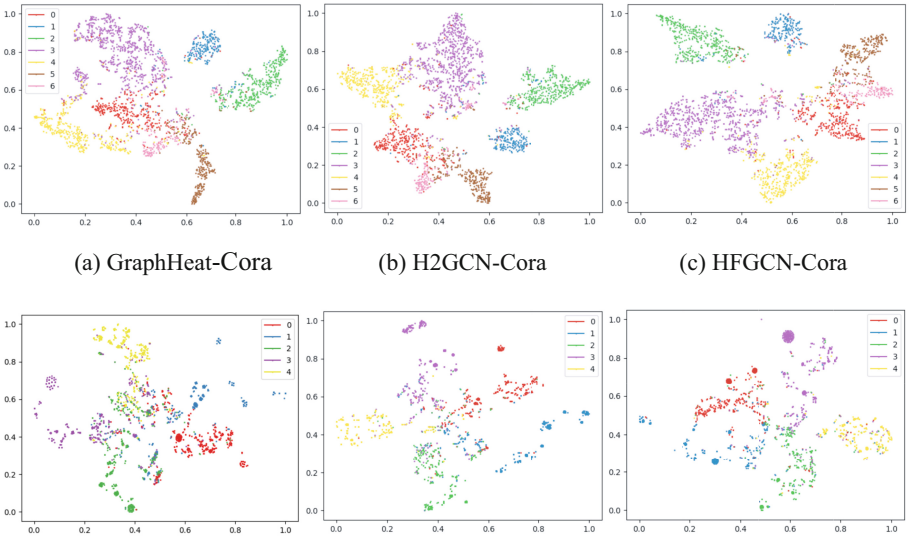
Table 2. Node classification accuracy on seven datasets. The best results are in bold and the second best results are underlined.

Method	Cora	Citeseer	Texas	Wisconsin
MLP	74.37 ± 1.99	71.96 ± 1.15	84.61 ± 5.19	84.72 ± 7.97
GCN	85.41 ± 2.04	75.74 ± 1.87	54.69 ± 5.26	51.94 ± 5.11
GAT	87.73 ± 1.19	76.20 ± 1.32	55.17 ± 5.14	50.67 ± 10.86
GraphHeat	87.91 ± 1.80	76.67 ± 1.49	80.87 ± 7.61	80.09 ± 8.78
Geom-GCN	84.49 ± 1.17	74.49 ± 1.59	60.02 ± 4.53	68.60 ± 6.07
GPR-GNN	86.20 ± 1.55	74.44 ± 1.55	86.49 ± 2.83	85.07 ± 4.59
FAGCN	83.80 ± 1.89	75.33 ± 1.92	81.45 ± 5.45	84.05 ± 7.10
H2GCN	88.01 ± 1.19	76.60 ± 1.83	85.01 ± 3.77	**88.45 ± 5.75**
HOG-GCN	86.40 ± 1.54	75.40 ± 2.27	85.25 ± 4.51	83.08 ± 5.40
HFGCN	**88.55 ± 1.42**	**77.23 ± 1.58**	**88.26 ± 3.12**	86.65 ± 5.30

Method	Cornell	Chameleon	Squirrel
MLP	82.30 ± 4.43	79.07 ± 2.07	73.65 ± 1.12
GCN	59.65 ± 7.69	55.98 ± 0.84	40.36 ± 1.44
GAT	60.77 ± 11.18	62.63 ± 2.94	44.48 ± 1.67
GraphHeat	50.08 ± 13.63	74.63 ± 2.07	62.23 ± 1.55
Geom-GCN	59.12 ± 6.01	62.40 ± 3.54	45.05 ± 3.00
GPR-GNN	83.86 ± 4.45	60.51 ± 4.76	45.59 ± 3.50
FAGCN	78.78 ± 6.14	76.72 ± 2.45	68.84 ± 1.62
H2GCN	78.82 ± 5.17	**79.93 ± 2.02**	73.51 ± 0.97
HOG-GCN	81.52 ± 6.60	71.06 ± 3.58	51.96 ± 2.94
HFGCN	**84.47 ± 4.80**	79.63 ± 1.71	**73.70 ± 1.42**

In addition, in the citation network, compared with GraphHeat, which uses a lower-pass way to model the smoothness of homophilic graphs, HFGCN can also better capture the features of a small number of heterophilic nodes in homophilic graphs.

5.4 Visualization



(a) GraphHeat-Cora (b) H2GCN-Cora (c) HFGCN-Cora

(d) GraphHeat- Chameleon (e) H2GCN- Chameleon (f) HFGCN-Chameleon

Fig. 2. Visualization results.

In order to visually demonstrate the learning effect of our method, we use t-SNE to map the representation of the nodes in the network's last layer to a two-dimensional space and visualize it. Figure 2 is the visualization results of the Cora and Chameleon datasets under a specific division, respectively, and all nodes are colored with ground-truth labels. For comparison, we selected two baseline models with better performance: GraphHeat and H2GCN. The ideal node representation should be compact within the class and have clear boundaries between classes in the visualization results. For the Cora dataset, the node representations learned by HFGCN and H2GCN have clear advantages over GraphHeat. The low-pass mechanism of GraphHeat makes the representation of nodes of different classes more concentrated. For the Chameleon dataset, compared with the three baseline methods, the node representations learned by HFGCN are more clearly demarcated between different classes. The visualization results can illustrate the effectiveness of our hybrid filtering idea on graphs of different homophily rates to a certain extent.

6 Conclusion

We introduced HFGCN, an innovative approach designed for the adaptive aggregation of node information in graph-based data. Leveraging spectral domain insights, HFGCN employs low-pass, high-pass, and self-aggregating filters to address the diverse connectivity patterns. This methodology, supplemented by an attention mechanism for attribute-based neighbor importance, enables nuanced and efficient content analysis and retrieval

across complex graph structures. Our comprehensive experiments across seven datasets underscore HFGCN's effectiveness for both homophilic and heterophilic graphs. However, the scalability of HFGCN, particularly in handling large-scale graphs, presents a challenge due to increased computational demands.

Acknowledgments. This work was supported by the Youth Innovation Promotion Association CAS (Grant No. 2021103) and the CAS Project for Young Scientists in Basic Research (Grant No. YSBR-040).

References

1. Abu-El-Haija, S., Perozzi, B., Kapoor, A., et al.: Mixhop: higher-order graph convolutional architectures via sparsified neighborhood mixing. In: International Conference on Machine Learning, pp. 21–29. PMLR (2019)
2. Bo, D., Wang, X., Shi, C., Shen, H.: Beyond low-frequency information in graph convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 3950–3957 (2021)
3. Bruna, J., Zaremba, W., Szlam, A., et al.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
4. Castleman, K.R.: Digital Image Processing. Prentice Hall Press (1996)
5. Chien, E., Peng, J., Li, P., et al.: Adaptive universal generalized pagerank graph neural network. arXiv preprint arXiv:2006.07988 (2020)
6. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
7. Gao, Y., Wang, X., He, X., Liu, Z., Feng, H., Zhang, Y.: Addressing heterophily in graph anomaly detection: a perspective of graph spectrum. In: Proceedings of the ACM Web Conference 2023, pp. 1528–1538 (2023)
8. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
9. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
11. Lee, J.S.: Digital image enhancement and noise filtering by use of local statistics. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-2**, 165–168 (1980)
12. Liu, K., Xue, F., Guo, D., et al.: Multimodal graph contrastive learning for multimedia-based recommendation. IEEE Trans. Multimedia (2023)
13. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. **9**(11) (2008)
14. Pei, H., Wei, B., Chang, K.C.C., et al.: Geom-GCN: geometric graph convolutional networks. arXiv preprint arXiv:2002.05287 (2020)
15. Vaswani, A., Shazeer, N.M., Parmar, N., et al.: Attention is all you need. In: Neural Information Processing Systems (2017)
16. Veličković, P., Cucurull, G., Casanova, et al.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
17. Wang, T., Jin, D., Wang, R., et al.: Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 4210– 4218 (2022)

18. Xu, B., Xu, C., Su, B.: Cross-modal graph attention network for entity alignment. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 3715–3723 (2023)
19. Xu, B., Shen, H., Cao, Q., et al.: Graph convolutional networks using heat kernel for semi-supervised learning. arXiv preprint arXiv:2007.16002 (2020)
20. Xu, B., Shen, H., Cao, Q., et al.: Graph wavelet neural network. arXiv preprint arXiv:1904.07785 (2019)
21. Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.i., Jegelka, S.: Representation learning on graphs with jumping knowledge networks. In: International Conference on Machine Learning, pp. 5453–5462. PMLR (2018)
22. Yu, P., Tan, Z., Lu, G., Bao, B.K.: Multi-view graph convolutional network for multimedia recommendation. In: Proceedings of the 31st ACM International Conference on Multimedia. MM'23, Association for Computing Machinery, NY, USA, pp. 6576–6585 (2023)
23. Zhu, J., Yan, Y., Zhao, L., et al.: Beyond homophily in graph neural networks: Current limitations and effective designs. In: Advances in Neural Information Processing Systems, vol. 33, pp. 7793–7804 (2020)