Beyond Homophily: Reconstructing Structure for Graph-agnostic Clustering

Erlin Pan [1] **Zhao Kang** [1 *]

Abstract

Graph neural networks (GNNs) based methods have achieved impressive performance on node clustering task. However, they are designed on the homophilic assumption of graph and clustering on heterophilic graph is overlooked. Due to the lack of labels, it is impossible to first identify a graph as homophilic or heterophilic before a suitable GNN model can be found. Hence, clustering on real-world graph with various levels of homophily poses a new challenge to the graph research community. To fill this gap, we propose a novel graph clustering method, which contains three key components: graph reconstruction, a mixed filter, and dual graph clustering network. To be graph-agnostic, we empirically construct two graphs which are high homophily and heterophily from each data. The mixed filter based on the new graphs extracts both low-frequency and high-frequency information. To reduce the adverse coupling between node attribute and topological structure, we separately map them into two subspaces in dual graph clustering network. Extensive experiments on 11 benchmark graphs demonstrate our promising performance. In particular, our method dominates others on heterophilic graphs. The code is available at DGCN.

1. Introduction

Graphs are pervasive and have been widely used in numerous real-world scenarios, such as social networks, traffic networks, and recommendation systems (Liu et al., 2021). Graph clustering that groups nodes without the need of human annotation is a fundamental yet challenging graph analysis task (Zhang et al., 2019; Liu et al., 2022a). Based on GNN' powerful structure information exploitation capability, many clustering methods (Wang et al., 2019; Fan et al., 2020) have achieved remarkable performance on homophilic graphs.

However, there are many heterophilic graphs in which most of connected nodes belong to different classes (Pei et al., 2020; Xie et al., 2023). Traditional GNNs learn representations via message passing mechanism under the assumption of homophily (Fang et al., 2022). Facing heterophilic graphs, previous approaches mainly suffer two limitations. On the one hand, the local neighbors in a graph are nodes that are proximally located, while nodes that are semantically similar might be far apart on heterophilic graph (Zhu et al., 2020). Thus, existing techniques fail to capture long-range information from distant nodes. On the other hand, they don't distinguish similar and dissimilar neighbors, which carry different amounts of information. Learning a discriminative graph representation needs to pass diverse messages between nodes on heterophilic graphs. Consequently, the GNN-based methods performing well on homophilic graphs produce unsatisfactory performance on heterophilic graphs (Abu-El-Haija et al., 2019).

Numerous GNN methods have been proposed to deal with heterophily. Some approaches expand neighbor fields, while others refine the GNN architectures. The first class includes exploring high-order information (Wang & Derr, 2021) and mining more neighbors in other spaces (Wang et al., 2022). The latter class aggregates message adaptively, like adaptive filter (He et al., 2021; Luan et al., 2021; Wang & Zhang, 2022) and attention or weight mechanism (Bo et al., 2021; Yang et al., 2021). Moreover, several recent methods, like (Bi et al., 2022) and (Topping et al., 2022), preprocess heterophily graphs to make them fit for GNNs.

Though the aforementioned methods improve the performance of GNNs on heterophilic graphs in some downstream tasks, there exists two critical problems: 1) the training of customized network, the learning of adaptive filter, and graph rewiring (Bi et al., 2022) rely on labelled samples, which makes them not be applicable to clustering task. 2) GNNs embed the raw data into a single subspace, where the coupling of attribute and topological structure aggravates the adverse effect of heterophily. Any incorrect or incomplete in the attributes or structures would deteriorate the quality of learned representations.

For unsupervised learning, the first and foremost challenge we face is that there is no labels for us to judge

[*]Corresponding author [1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. Correspondence to: Erlin Pan <wujisixsix6@gmail.com>, Zhao Kang < zkang@uestc.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

whether a graph is homophilic or heterophilic. Therefore, it is not practical to develop individual models to handle homophilic and heterophilic graphs separately for clustering. Moreover, it is also subjective to simply classify a graph as homophilic or heterophilic since real-world graph data could have various levels of homophily. To address this open problem, we propose a holistic approach in this work to handle real graphs, which includes three key components: graph reconstruction, a mixed filter, and dual graph clustering network. We first construct new homophilic and heterophilic graphs to explore both low-frequency and high-frequency information. In particular, the structure reconstruction process is fully unsupervised and general. The mixed filter is designed to smooth graph signals, which makes our model be applicable to both homophilic and heterophilic graphs. Finally, the smoothed features are fed into dual graph clustering network to obtain the clustering result. We summarize our contributions as follows:

- We propose two unsupervised graph construction strategies to extract homophilic and heterophilic information from any type of graphs.
- We design a mixed filter that exploits both low-frequency and high-frequency components of data. This approach can also be applied to classification task.
- We reduce the adverse coupling between attribute and topological structure by mapping them into two different subspaces.
- Extensive experiments on homophilic and heterophilic graphs demonstrate the promising performance of our proposed method.

2. Related Work

2.1. Graph Clustering

Numerous attributed graph clustering methods have been proposed to exploit nodes' feature and topological structure information. These methods can be roughly classified into two categories: GNN-based methods and shallow graph embeddings based methods. GNN-based methods learn the graph representation for clustering via aggregating neighborhood information in prior graph (Kipf & Welling). To improve the clustering performance, (Cheng et al., 2021) and (Peng et al., 2021) adopt attention mechanism to adaptively integrate topology and attribute information. Inspired by the success of contrastive learning, (Xia et al., 2021) learns a consensus representation from multiview graph. Shallow methods learn graph embeddings without neural networks and perform traditional clustering methods on them. For example, (Zhang et al., 2019) obtains more discriminative representations by enlarging receptive field to explore high-order information. (Lin & Kang, 2021) and (Lin et al., 2023) learn clustering-favorable embeddings via low-pass filter. (Pan & Kang, 2021) constructs a new graph for clustering via pulling nearest neighbors close.
However, these methods only focus on homophilic graphs and are not directly applicable to heterophilic graphs. In this work, we aim to develop an omnipotent method, which is suitable for real graph with different levels of homophily.

2.2. Heterophilic Graph Learning

Heterophilic structure is prevalent in practice, from personal relationships in daily life to chemical and molecular scientific study. Developing powerful heterophilic GNN models is a hot research topic. (Lim et al., 2021b;a) provide general benchmarks for heterophilic graph learning. In addition, many methods have been proposed to revise GNNs for heterophilic graphs. (Yang et al., 2021) specifies propagation weight for each attribute to make GNNs fit heterophilic graphs and (Li et al., 2022) explores the underlying homophilic information by capturing the global correlation of nodes. (Zhu et al., 2020) enlarges receptive field via exploring high-order structure. (Chien et al., 2021) adaptively combines the representation of each layer and (Chen et al., 2020) integrates embeddings from different depths with residual operation.
Although these approaches alleviate the heterophilic problem to some extent, they rely on prior knowledge like labels for training, which are not available in unsupervised scenario. To our best knowledge, clustering on heterophilic graph has never been investigated. To fill this gap, we reconstruct homophilic and heterophilic graphs to make the proposed model handle any kinds of graphs.

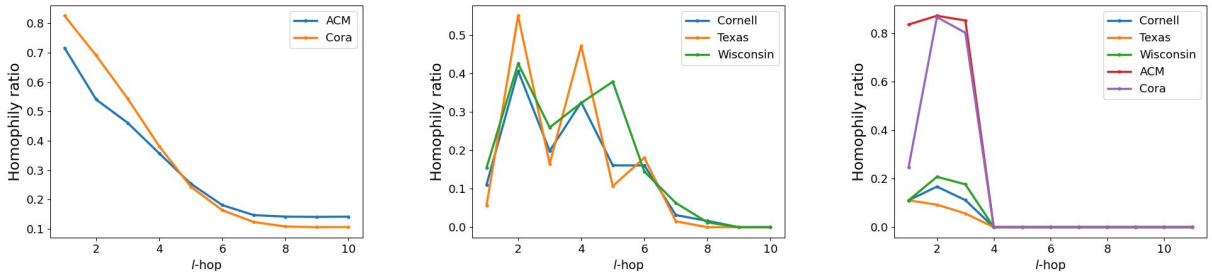
3. Methodology

3.1. Notation

Define graph data as $G = \{\mathcal{V}, \widetilde{A}, X\}$, where $\mathcal{V}$ represents the set of N nodes, $X = \{x_1, ..., x_N\}^\top$ is the feature matrix. Initial graph structure $\widetilde{A}$ characterizes the relation between nodes. D represents the degree matrix. The normalized adjacency matrix is $A = D^{-\frac{1}{2}}(\widetilde{A} + I)D^{-\frac{1}{2}}$ and the corresponding graph Laplacian is $L = I - A$. $1.$ is a matrix with all 1s.

3.2. Structure Reconstruction

In practice, we can't know whether a given graph is homophilic or not in unsupervised tasks. Hence, separately developing homophilic and heterophilic methods is unrealistic. Moreover, real graphs always contain both homophilic and heterophilic nodes. To have a holistic model, we develop a structure reconstruction approach. Specifically, we construct a heterophilic graph and a homophilic graph from the original graph.



(a) (b) (c)

Figure 1. (a) and (b) show the changes of homophily ratio in homophilic and heterophilic graphs. (c) shows the homophily ratios of 1-hop and other hops nodes sharing the same label. It's clear that homophily ratio of homophilic graphs decreases when more hops are considered. However, the change in heterophilic graphs is irregular. Particularly, the nodes in 1-hop and 2-hop sharing the same label have the highest homophily ratio.

3.2.1. Heterophilic Graph Construction

Firstly, we select the nodes which are far away from each other in both feature space and structure space as negative pairs, which prevents us from false negative pairs. Specifically, we use complementary graphs of similarity graph and topology graph to construct a heterophily graph. The procedure is formulated as follows:

$$\begin{aligned} \bar{W} &= 1. - W, \\ \bar{A} &= 1. - A, \\ H &= \bar{W} \odot \bar{A}, \end{aligned} \tag{1}$$

where the similarity matrix W is obtained through cosine similarity of node features, which characterizes the closeness among nodes in feature space. $\odot$ represents the Hadamard product, which is used to describe non-neighbor relation in both feature space and topology space. For homophilic graph, nodes of the same class tend to be adjacent topologically, and neighbors in complementary graph have a tendency to be different. The adjacent nodes in heterophilic graph are often dissimilar because of the connection between nodes of diverse types (Ning et al., 2022). Moreover, nodes with large edge weights in similarity graph more likely belong to the same class , thus adjacent nodes in corresponding complementary graph are more possibly have small edge weights, i.e., they are different. It is rational to pick adjacent nodes from both $\bar{W}$ and $\bar{A}$, so that the reconstructed graph H is heterophilic. H could be dense, thus we just keep 5 edges for each node corresponding to top 5 dissimilar nodes.

3.2.2. Homophilic Graph Construction

In practice, even the homophilic graph doesn't have a homophily score of 1, i.e., there exists some heterophilic nodes in homophilic graph. Thus, we could further improve the homophily level of raw graph by minimizing the distances among adjacent nodes, which is formulated as:

$$\min_{S_{i:}} \sum_{j=1}^{N} S_{ij} \left\| x_i - x_j \right\|^2 ,$$

where $S_{i:}$ represents the i-th row of S. To avoid the trivial solution $S = I$, we rewrite above equation as:

$$\min_{S_{i:}} \sum_{j=1}^{N} S_{ij} \left\| x_i - x_j \right\|^2 + S_{ij}^2.$$

It's clear that the graph S will be more homophilic when edges are defined by nodes sharing high similarity. The homophily ratio $h^{(l)}$ of nodes in different hops vary considerably according to Fig. 1, where $h^{(l)}(G) = \frac{1}{|\tilde{A}_{ij}^{(l)}|} \sum_{\tilde{A}_{ij}^{(l)}>0} \mathbb{1}(y_i = y_j)$ and l is the number of hop and exponent of $\tilde{A}$ (Ma et al., 2022). Consequently, the message propagation path in S could be incorrect when the 2-hop neighbors of a node are dissimilar to its 1-hop neighbors. Furthermore, the nodes sharing the same label with 1-hop and 2-hop neighbors have the highest ratio. Based on this observation, we design a regularization term to integrate the 1-hop and 2-hop neighbor relation, i.e., we enforce all 2-hop neighboring nodes are in the set of 1-hop neighborhood. Let $\|x_i - x_j\|^2 = K_{ij}$, then we construct a homophilic graph S by solving the following optimization problem:

$$\begin{aligned} &\min_{S_{ij}} S_{ij} K_{ij} + S_{ij}^2 + \left(S_{ij}^{(2)} - S_{ij} \right)^2 , \\ &\text{s.t. } S_{ij} > 0, \sum_{j=1}^{N} S_{ij} = 1, \end{aligned} \tag{2}$$

where $S^{(2)}$ is the 2-hop graph, i.e., $S^{(2)} = S \times S$.

Optimization Firstly, we initialize S with A. Then we reformulate problem (2) via its Lagrangian function:

$$\begin{aligned}\min_{S_{i:}} \sum_{j=1}^{N} [S_{ij}K_{ij} + (S_{ij}^{(2)} - S_{ij})^2 + S_{ij}^2] \\ - \sum_{j=1}^{N} \lambda_{1j} S_{ij} - \lambda_{2i}(\sum_{j=1}^{N} S_{ij} - 1).\end{aligned} \tag{3}$$

The derivative of Eq. (3) w.r.t. S_{ij} is

$$\begin{aligned}&K_{ij} + 2(S_{ii} + S_{jj} - 1)(S_{ij}^{(2)} - S_{ij}) + 2S_{ij} - \lambda_{1j} - \lambda_{2j} \\ &+ \sum_{f \neq j}^{N} 2S_{jf}[S_{ij}S_{jf} + (S_{if}^{(2)} - S_{ij}S_{jf}) - S_{if}].\end{aligned} \tag{4}$$

Remove the self-loop on graph, i.e., let $S_{ii} = 0$. S and $S^{(2)}$ can be regarded as constants at the last iteration. By introducing C_f,

$$C_f = \begin{cases} S_{if}^{(2)} - S_{ij}S_{jf} - S_{if}, i \neq f \\ 0, otherwise \end{cases}$$

we rewrite Eq. (4) as:

$$K_{ij} - 2(S_{ij}^{(2)} - S_{ij}) + 2S_{ij} - \lambda_{1j} - \lambda_{2i} + \sum_{f \neq j} 2S_{jf}^2 S_{ij} + 2S_{jf}C_f. \tag{5}$$

Note that the KKT condition $\lambda_{1j}S_{ij} = 0$, which yields:

$$\begin{aligned}S_{ij}(K_{ij} - 2(S_{ij}^{(2)} - S_{ij}) + 2S_{ij} - \lambda_{2j} \\ + \sum_{f \neq j} 2S_{jf}^2 S_{ij} + 2S_{jf}C_f) = 0.\end{aligned} \tag{6}$$

Afterwards, we obtain the closed-form solution of S_{ij}:

$$S_{ij} = [\frac{2S_{ij}^{(2)} + \lambda_{2j} - K_{ij} - 2\sum_{f \neq j} S_{jf}C_f}{2(2 + \sum_{f \neq j} S_{jf}^2)}]_+, \tag{7}$$

where $[\bullet]_+$ operator means $\max(\bullet, 0)$. Following the second constraint, we have

$$\sum_j [\frac{2S_{ij}^{(2)} + \lambda_{2j} - K_{ij} - 2\sum_{f \neq j} S_{jf}C_f}{2(2 + \sum_{f \neq j} S_{jf}^2)}]_+ - 1 = 0.$$

.

To obtain S_{ij}, we need to compute λ_{2j} first by solving this optimization problem:

$$\min_{\lambda_{2j}} \sum_{j=1}^{N} [\frac{2S_{ij}^{(2)} + \lambda_{2j} - K_{ij} - 2\sum_{f \neq j} S_{jf}C_f}{2(2 + \sum_{f \neq j} S_{jf}^2)}]_+ - 1. \tag{8}$$

We can solve Eq. (8) by gradient descent algorithm or treating it as a Linear programming (LP) problem.

3.3. Graph Filtering

Based on the assumption that graph signal should be smooth, i.e., the neighbor nodes tend to be similar, low-pass filter has been used to obtain smoothed representations that are clustering-friendly (Pan & Kang, 2021). One typical low-pass filter (Zhang et al., 2019) can be formulated as:

$$F = (I - \frac{1}{2}L)^k X, \tag{9}$$

where F is the filtered representation, k is the order of graph filtering.

However, Eq. (9) could be ineffective resulting from its heavy dependence on raw topological graph which could be noisy and incomplete. Additionally, low-pass filtering neglects the high-frequency components in data, which leads to information loss and inferior performance. This would be more worse for heterophilic graph, where high-frequency information plays a critical role. Since it is impossible to know whether a given graph is homophilic or not in unsupervised learning, it's necessary to design a generic filter to handle different types of graphs. To this end, we design a mixed filter for graph data as follows:

$$F = \mu(\frac{1}{2}L_H)^k X + (1 - \mu)(I - \frac{1}{2}L_S)^k X, \tag{10}$$

where $\mu > 0$ is a trade-off parameter balancing low-pass and high-pass representations, L_S and L_H are the normalized Laplacian matrices of reconstructed homophilic and heterophilic graphs. Note that our mixed filter is not combining low-pass and high-pass filter simply, we apply newly constructed graphs rather than the raw graph, which often has low-quality. The final representation F is used as the input of clustering network.

3.4. Dual Graph Clustering Network

In this section, we introduce our proposed Dual Graph Clustering Network (DGCN) to address the nodes clustering task. As shown in Fig. 2, DGCN contains two unshared encoders and a decoder, and all of them are MLPs. Different from GNNs which learn representations in only one space, two encoders E_{θ_F} and E_{θ_A} are utilized to map filtered features and structural graph into attribute subspace and structure subspace respectively. This would reduce the interaction between attribute and structure for heterophilic nodes. The structure encoder is applied to preserve some original structure information. The obtained representations are:

$$\begin{aligned}Z_F &= E_{\theta_F}(F), \\ Z_A &= E_{\theta_A}(A).\end{aligned} \tag{11}$$

Moreover, to alleviate representation collapse, i.e., representations of all nodes tend to be the same, we add a correlation reduction item to prevent it (Zbontar et al., 2021).

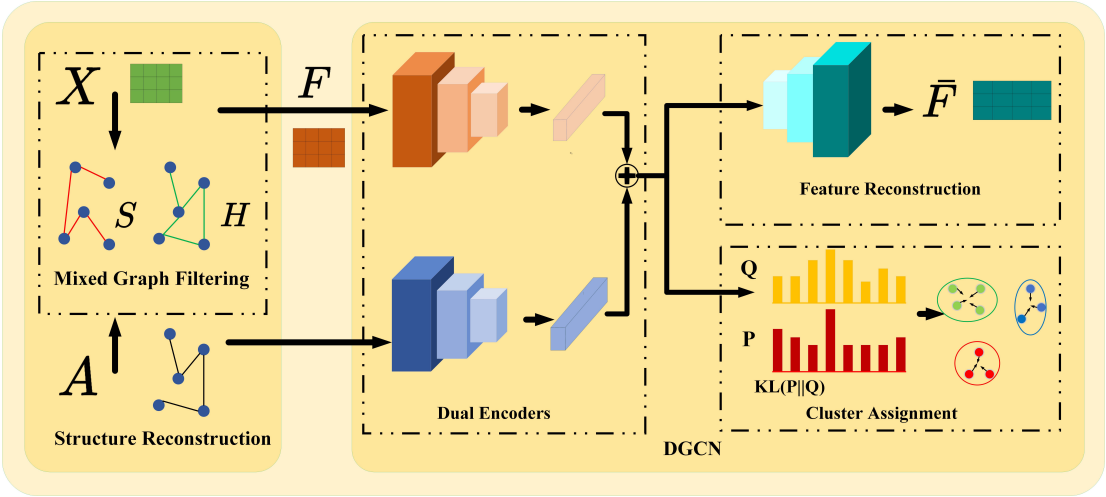


Figure 2. The framework of DGCN. There are three key components: 1) structure reconstruction is used to reconstruct homophilic and heterophilic graphs; 2) a mixed graph filter is designed to obtain smooth representations for any kind of graphs; 3) dual encoders are applied to learn embeddings in attribute and topology space.

$$\begin{aligned}\mathcal{L}_{CR} &= \frac{1}{d^2}\sum\left(\mathbf{M}-\widetilde{\mathbf{I}}\right)^2 \\ &= \frac{1}{d^2}\sum_{i=1}^{d}\left(\mathbf{M}_{ii}-1\right)^2 + \frac{1}{d^2-d}\sum_{i=1}^{d}\sum_{j\neq i}\mathbf{M}_{ij}^2,\end{aligned} \tag{12}$$

where d is the dimension of node attribute, M is the similarity of corresponding nodes in two encoders,

$$M_{ij} = \frac{Z_{Ai}^{\top} Z_{Fj}}{\|Z_{Ai}\| \cdot \|Z_{Fj}\|}.$$

Afterwards, Z_A and Z_F are concatenated as Z.

Decoder is employed to obtain reconstructed features $\bar{F}$. The features of some "easy samples" change little during reconstruction, which indicates that these nodes contribute less information for training our model. We adopt the Scaled Cosine Error (SCE) as the objective of reconstruction (Hou et al., 2022), which can down-weight easy samples' contribution by controlling sharpening parameter β in training:

$$\mathcal{L}_{\mathrm{SCE}} = \sum_{i=1}^{N}\left(1-\frac{F_i^{\top}\bar{F}_i}{\|F_i\|\cdot\|\bar{F}_i\|}\right)^{\beta}, \beta \geq 1. \tag{13}$$

Finally, we pull soft cluster assignment probabilities distribution and target distribution for cluster enhancement. Specifically, the soft assignment distribution Q is computed as:

$$q_{iu} = \frac{\left(1+\|z_i-\sigma_u\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{u'}\left(1+\|z_i-\sigma_{u'}\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}}, \tag{14}$$

where cluster centres σ are initialized by k-means on embeddings and α is the Student's t-distribution's degree of freedom. Then target distribution P is formulated as:

$$p_{iu} = \frac{q_{iu}^2/\sum_i q_{iu}}{\sum_{k'}\left(q_{ik'}^2/\sum_i q_{ik'}\right)}. \tag{15}$$

We minimize the KL divergence loss between Q and P distributions to make the data representation closer to cluster centres and improve cluster cohesion:

$$\mathcal{L}_{CLU} = KL(P\|Q) = \sum_i\sum_u p_{iu}\log\frac{p_{iu}}{q_{iu}} \tag{16}$$

In summary, the objective of DGCN can be computed by:

$$\mathcal{L} = \mathcal{L}_{CR} + \mathcal{L}_{SCE} + \mathcal{L}_{CLU}. \tag{17}$$

We minimize this objective function to train our model and the result of clustering for node i is:

$$Y_i = \operatorname*{argmax}_c q_{ic}. \tag{18}$$

4. Experiments

4.1. Datasets

To evaluate the effectiveness of the proposed method, we conduct extensive experiments on 11 benchmarks, including homophilic graph datasets, like Cora, Citeseer (Kipf & Welling), ACM (Fan et al., 2020), AMAP (Liu et al., 2022b), EAT (Mrabah et al., 2022); heterophilic graph datasets, like Texas, Cornell, Wisconsin, Washington (Pei et al., 2020), Twitch (Lim et al., 2021b), and Squirel (Rozemberczki et al., 2021). The statistical information of them is summarized in Table 1.

Table 1. Statistics information of datasets.

Graph datasets		Nodes	Dims.	Edges	Clusters	Homophily Ratio
Heterophilic Graphs	Texas	183	1703	325	5	0.0614
	Cornell	183	1703	298	5	0.1220
	Wisconsin	251	1703	515	5	0.1703
	Washington	230	1703	786	5	0.1434
	Twitch	1912	2545	31299	2	0.5660
	Squirrel	5201	2089	217073	5	0.2234
Homophilic Graphs	Cora	2708	1433	5429	7	0.8137
	Citeseer	3327	3703	4732	6	0.7392
	ACM	3025	1870	29281	3	0.8207
	AMAP	7650	745	119081	8	0.8272
	EAT	399	203	5994	4	0.4046

Table 2. Results on heterophilic graphs. The best results are **bolded** with red and the second-best performance is also **bolded**. '-' means that the source code can't produce any results.

Methods	Texas		Cornell		Wisconsin		Washington		Twitch		Squirrel	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
DAEGC	45.99	11.25	42.56	12.37	39.62	12.02	40.43	-	56.59	-	25.55	2.36
MSGA	57.22	12.13	50.77	14.05	54.72	16.28	-	-	-	-	27.42	4.31
FGC	53.48	5.16	44.10	8.6	50.19	12.92	51.30	-	**70.71**	-	25.11	1.32
GMM	58.29	13.06	58.86	-	52.08	8.89	60.86	20.56	-	-	-	-
RWR	57.22	13.87	58.29	-	53.96	16.02	63.91	23.13	-	-	-	-
ARVGA	59.89	16.37	56.23	-	56.23	13.73	60.87	16.19	-	-	-	-
DGCN$_{\beta=2}$	**72.68**	**33.67**	**62.29**	**29.93**	**71.71**	**41.29**	**69.13**	**28.22**	70.34	**39.84**	**31.34**	**7.24**
DGCN$_{\beta=1}$	**74.57**	**39.93**	**61.74**	**21.76**	**72.90**	**43.52**	**69.56**	**28.43**	**71.00**	**41.81**	**31.39**	**8.54**

4.2. Comparison Methods

To demonstrate the superiority of our method, we adopt 13 baselines for performance comparison. Specifically, there are four kinds of methods: 1) typical GNN-based methods, like DAEGC (Wang et al., 2019), MSGA (Wang et al., 2021), SSGC (Zhu & Koniusz, 2021), GMM (Zhu et al., 2022), RWR (Huang et al., 2019), ARVGA (Pan et al., 2019); 2) contrastive learning-based methods, like MVGRL (Hassani & Khasahmadi, 2020), SDCN (Bo et al., 2020), DFCN (Tu et al., 2021), and DCRN (Liu et al., 2022b), which employ MLP and GNNs jointly to learn an aligned representation from augmented views; 3) state-of-the-art MLP-based clustering method AGE (Cui et al., 2020), which obtains a clustering-favorable representation via Laplacian smoothing filter and adaptive encoder; 4) shallow methods which utilize a low-pass filter to smooth the raw features and remove the noises, like MCGC (Pan & Kang, 2021) and FGC (Kang et al., 2022). We implement DGCN with both $\beta = 1$ and $\beta = 2$. In fact, SCE loss with $\beta = 1$ is equivalent to traditional Frobenius norm.

4.3. Experimental Setting

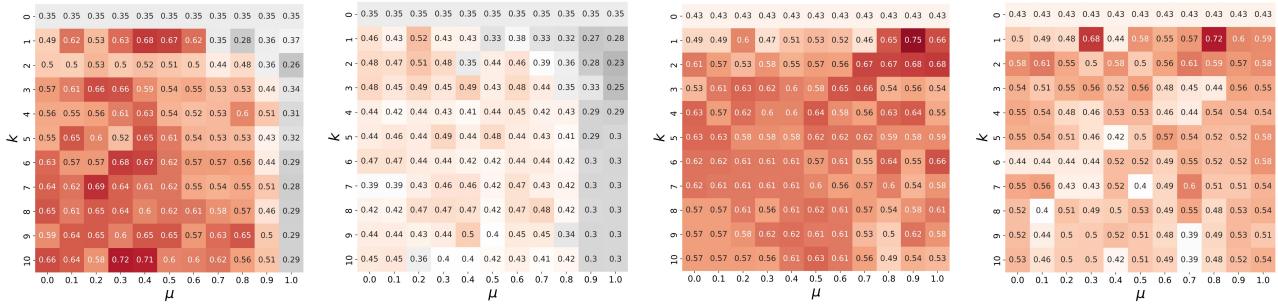
For fairness, all compared methods are implemented with the same setting in original papers. Our network is trained with Adam optimizer for 500 epochs until convergence. The learning rate of optimizer is set to 1e-2. We tune filter order k in [1, 2, 3, 4, 5, 10]. We adopt ACCuracy (ACC) and Normalized Mutual Information (NMI) as cluster metrics in all experiments.

4.4. Results

Table 2 reports the results on heterophilic datasets. We can see that DGCN obtains much better performance than existing clustering methods. With respect to the second-best performance, our method improves the accuracy up to 10% on Texas and Wisconsin, 5% on Washington. One reason is that the comparison methods are not deliberately designed for heterophilic graphs. To our best knowledge, there is no clustering techniques considering graph heterophily in the literature. In addition, GNN-based methods don't perform well on heterophilic graphs, which is consistent with the conclusion on classification task. By contrast, our method

Table 3. Results on homophilic graphs. AvgRank represents the average ranking of methods on five graphs.

Methods	Cora		Citeseer		ACM		AMAP		EAT		AvgRank
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	
DFCN	36.33	19.36	69.50	43.9	90.90	69.40	**76.88**	**69.21**	49.37	**32.90**	5.4
DCRN	48.93	-	70.86	**45.86**	91.93	71.56	**79.94**	**73.70**	51.33	-	3.6
SSGC	69.60	54.71	69.11	42.87	89.09	64.71	60.23	60.37	32.41	4.65	7.6
MVGRL	70.47	55.57	68.66	43.66	86.73	60.87	45.19	36.89	32.88	11.72	8.2
SDCN	60.24	50.04	65.96	38.71	90.45	68.31	53.44	44.85	39.07	8.83	7.8
AGE	**73.50**	**57.58**	70.39	44.92	90.91	69.42	75.98	-	47.26	-	4
MCGC	42.85	24.11	64.76	39.11	91.64	70.71	71.64	61.54	32.58	7.04	7.6
FGC	**72.90**	56.12	69.01	44.02	88.13	62.77	71.04	-	41.11	-	6.2
DGCN$_{\beta=2}$	72.19	56.04	**71.27**	44.13	**92.03**	**71.58**	76.07	66.13	**53.13**	22.92	**2.6**
DGCN$_{\beta=1}$	72.89	**56.82**	**71.60**	**44.83**	**92.60**	**71.85**	76.06	65.46	**53.52**	**24.81**	**2**



(a) Cora with mixed filter (b) Cora with raw graph A (c) Texas with mixed filter (d) Texas with raw graph A

Figure 3. Accuracy on Cora and Texas with raw graphs and reconstructed graphs.

takes care of both homophilic and heterophilic nodes, thus it improves the performance significantly. In addition, we can see that β can impact the performance a little bit.

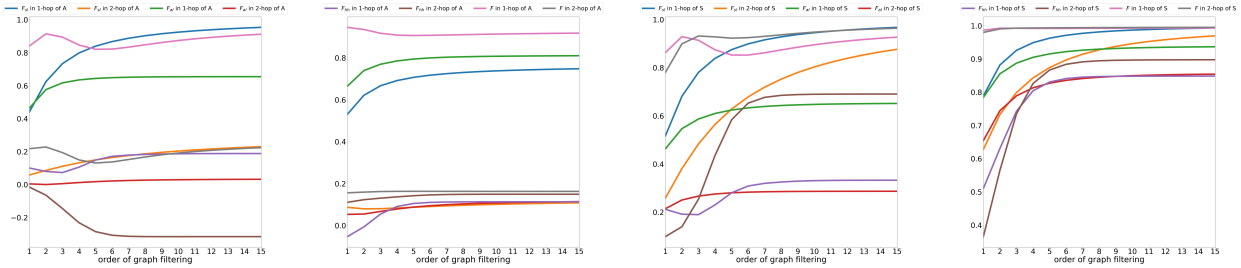
Table 3 shows the clustering results on homophilic graphs. Our method achieves competitive performance and shows the highest rank. It can be observed that the SOTA contrastive learning methods produce unstable results. This is because their performance highly depends on the graph augmentation strategy, which requires domain knowledge and is not flexible to any data. For shallow methods FGC and MCGC, their performance also changes a lot on different datasets. This is attributed to their usage of low-pass filter, which is not suitable to real graphs with different levels of homophily. DGCN is better than AGE in most cases since our mixed filter explores both low-frequency and high-frequency components of original data and reduces information loss, while AGE neglects the high-frequency signal.

To sum up, DGCN obtains stable and promising results on both homophilic and heterophilic graphs. This is mainly because it extracts homophilic and heterophilic information from raw graph. Consequently, DGCN is applicable to cluster real graphs, where we have no idea of homophily ratios.

5. Ablation Study

Our mixed filter is based on the newly constructed homophilic and heterophilic graphs in which neighbor nodes tend to be similar and dissimilar, thus they contain low-frequency and high-frequency information, respectively. To show the effectiveness of our filter, we examine the clustering accuracy of DGCN with different filter orders k and balance parameters μ. Particularly, we don't apply graph filtering when $k = 0$ and employ low-pass (high-pass) filter only when $\mu = 0$ ($\mu = 1$). As shown in Fig. 3, graph filtering considerably boosts model performance on both homophilic and heterophilic graphs with A or reconstructed graphs. Furthermore, the best performance is always achieved when $\mu \neq 0$, which validates the importance of incorporating high-frequency information.

With respect to raw graph, the constructed graphs further boost the performance. However, there are two differences: 1) heterophilic graph is sensitive to high-frequency components while homophilic graph prefers to low-frequency



(a) A of Cora (b) A of Texas (c) S of Cora (d) S of Texas

Figure 4. The variation of average similarity of 1-hop and 2-hop neighbor nodes under different filter orders on Cora and Texas. F_{sl} and F_{al} represent the filtered features with low-pass filter on homophilic graph S and raw graph A, F_{hh} is obtained by high-pass filter on heterophilic graph H.

parts; 2) DGCN achieves the best performance on heterophilic graph with a small k, while it also works well with a large k on homophilic graph.

Table 4. The homophily score of constructed graphs.

	A	S	H
Cora	0.8137	0.8110	0.1699
Citeseer	0.7392	0.8018↑	0.1780
ACM	0.8207	0.9006↑	0.3236
AMAP	0.8272	0.9449↑	0.1259
EAT	0.4046	0.6039↑	0.2691
Texas	0.0614	0.4654↑	0.1767
Cornell	0.1220	0.4583↑	0.1734
Wisconsin	0.1703	0.4301↑	0.2274
Washington	0.1434	0.4522↑	0.1391↓
Twitch	0.5660	0.7103↑	0.2401↓
Squirrel	0.2234	0.4781↑	0.1999↓

We report the homophily score of constructed graphs in Table. 4. Homophilic graph S indeed has higher homophily score on most datasets, which proves that the designed regularizer pulls 1-hop and 2-hop neighbors close. The homophily score of heterophilic graph is improved by 16% on average. There is no clear improvement on Cora because few nodes are near each other in terms of a 1-hop and 2-hop manner. All Hs have a relative small value. However, the homophily score of H is increased on Texas, Cornell, and Wisconsin. The reason is that these graphs have few edges, then reconstructed heterophilic graphs become dense, where some edges connect the nodes of the same class.

To further understand the influence of filters, we plot the variation of cosine similarity by changing the filter order k in Fig. 4. As observed, on Cora, filtered feature similarity of 1-hop and 2-hop nodes obtained from high-order low-pass filter gets big while features from high-order high-pass filter tend to be dissimilar. By contrast, on heterophilic graph Texas, features from high-order low-pass filter or high-pass filter get more similar. These verify that graph filtering does preserve different kinds of topological information, i.e., low-pass and high-pass filter capture homophilic and heterophilic structures, respectively. Consequently, our mixed filter can handle two types of graphs simultaneously.

Moreover, similarity variation patterns with the mixed filter of S and A are similar. This means that S does not loss too much topological information from A. Importantly, the similarity with low-pass or high-pass filter converges to larger values, which indicates that filters with S and H get the similar nodes close.

6. Conclusion

In this paper, we make the first attempt to address the challenge of node clustering without any prior knowledge of graph homophily. We design a mixed filter to jointly explore low-frequency and high-frequency components by reconstructing two graphs from the raw graph, which can be generalized to other tasks. Moreover, to reduce the potential coupling between attribute and topology structure of data, we project the smoothed attribute and raw structure into two subspaces via two individual MLPs. Comprehensive experimental results on 11 graph benchmarks demonstrate the proposed method's impressive performance. Therefore, our method generalizes well to real data. One potential limitation is that there could be information loss resulting from feature reconstruction without considering structure. In future work, we aim to design a new architecture to handle this issue.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62276053).

References

Abu-El-Haija, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., Ver Steeg, G., and Galstyan, A. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pp. 21–29. PMLR, 2019.

Bi, W., Du, L., Fu, Q., Wang, Y., Han, S., and Zhang, D. Make heterophily graphs better fit gnn: A graph rewiring approach. *arXiv preprint arXiv:2209.08264*, 2022.

Bo, D., Wang, X., Shi, C., Zhu, M., Lu, E., and Cui, P. Structural deep clustering network. In *Proceedings of The Web Conference 2020*, pp. 1400–1410, 2020.

Bo, D., Wang, X., Shi, C., and Shen, H. Beyond low-frequency information in graph convolutional networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pp. 3950–3957. AAAI Press, 2021.

Chen, M., Wei, Z., Huang, Z., Ding, B., and Li, Y. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pp. 1725–1735. PMLR, 2020.

Cheng, J., Wang, Q., Tao, Z., Xie, D., and Gao, Q. Multi-view attribute graph convolution networks for clustering. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 2973–2979, 2021.

Chien, E., Peng, J., Li, P., and Milenkovic, O. Adaptive universal generalized pagerank graph neural network. In *9th International Conference on Learning Representations, ICLR 2021,*, 2021.

Cui, G., Zhou, J., Yang, C., and Liu, Z. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 976–985, 2020.

Fan, S., Wang, X., Shi, C., Lu, E., Lin, K., and Wang, B. One2multi graph autoencoder for multi-view graph clustering. In *Proceedings of The Web Conference 2020*, pp. 3070–3076, 2020.

Fang, R., Wen, L., Kang, Z., and Liu, J. Structure-preserving graph representation learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2022.

Hassani, K. and Khasahmadi, A. H. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pp. 4116–4126. PMLR, 2020.

He, M., Wei, Z., Xu, H., et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34: 14239–14251, 2021.

Hou, Z., Liu, X., Dong, Y., Wang, C., Tang, J., et al. Graphmae: Self-supervised masked graph autoencoders. *SIGKDD*, 2022.

Huang, P.-Y., Frederking, R., et al. Rwr-gae: Random walk regularization for graph auto encoders. *arXiv preprint arXiv:1908.04003*, 2019.

Kang, Z., Liu, Z., Pan, S., and Tian, L. Fine-grained attributed graph clustering. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pp. 370–378. SIAM, 2022.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *Bayesian Deep Learning Workshop (NIPS 2016)*.

Li, X., Zhu, R., Cheng, Y., Shan, C., Luo, S., Li, D., and Qian, W. Finding global homophily in graph neural networks when meeting heterophily. In *International Conference on Machine Learning, ICML 2022*, volume 162, pp. 13242–13256. PMLR, 2022.

Lim, D., Hohne, F., Li, X., Huang, S. L., Gupta, V., Bhalerao, O., and Lim, S. N. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021a.

Lim, D., Li, X., Hohne, F., and Lim, S.-N. New benchmarks for learning on non-homophilous graphs. *International World Wide Web Conferences*, 2021b.

Lin, Z. and Kang, Z. Graph filter-based multi-view attributed graph clustering. In *IJCAI*, pp. 2723–2729, 2021.

Lin, Z., Kang, Z., Zhang, L., and Tian, L. Multi-view attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1872–1880, 2023.

Liu, C., Wen, L., Kang, Z., Luo, G., and Tian, L. Self-supervised consensus representation learning for attributed graph. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2654–2662, 2021.

Liu, L., Kang, Z., Ruan, J., and He, X. Multilayer graph contrastive clustering network. *Information Sciences*, 613:256–267, 2022a.

Liu, Y., Tu, W., Zhou, S., Liu, X., Song, L., Yang, X., and Zhu, E. Deep graph clustering via dual correlation reduction. In *Proc. of AAAI*, 2022b.

Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., Chang, X.-W., and Precup, D. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

Ma, Y., Liu, X., Shah, N., and Tang, J. Is homophily a necessity for graph neural networks? In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

Mrabah, N., Bouguessa, M., Touati, M. F., and Ksantini, R. Rethinking graph auto-encoder models for attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Ning, Z., Wang, P., Wang, P., Qiao, Z., Fan, W., Zhang, D., Du, Y., and Zhou, Y. Graph soft-contrastive learning via neighborhood ranking. *arXiv preprint arXiv:2209.13964*, 2022.

Pan, E. and Kang, Z. Multi-view contrastive graph clustering. *Advances in neural information processing systems*, 34:2148–2159, 2021.

Pan, S., Hu, R., Fung, S.-f., Long, G., Jiang, J., and Zhang, C. Learning graph embedding with adversarial training methods. *IEEE transactions on cybernetics*, 50(6):2475–2487, 2019.

Pei, H., Wei, B., Chang, K. C., Lei, Y., and Yang, B. Geom-gcn: Geometric graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020,*, 2020.

Peng, Z., Liu, H., Jia, Y., and Hou, J. Attention-driven graph clustering network. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 935–943, 2021.

Rozemberczki, B., Allen, C., and Sarkar, R. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2), 2021.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022.

Tu, W., Zhou, S., Liu, X., Guo, X., Cai, Z., Zhu, E., and Cheng, J. Deep fusion clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9978–9987, 2021.

Wang, C., Pan, S., Hu, R., Long, G., Jiang, J., and Zhang, C. Attributed graph clustering: A deep attentional embedding approach. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 3670–3676, 2019.

Wang, T., Wu, J., Zhang, Z., Zhou, W., Chen, G., and Liu, S. Multi-scale graph attention subspace clustering network. *Neurocomputing*, 459:302–314, 2021.

Wang, T., Jin, D., Wang, R., He, D., and Huang, Y. Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4210–4218, 2022.

Wang, X. and Zhang, M. How powerful are spectral graph neural networks. In *International Conference on Machine Learning, ICML 2022,*, volume 162, pp. 23341–23362. PMLR, 2022.

Wang, Y. and Derr, T. Tree decomposed graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2040–2049, 2021.

Xia, W., Gao, Q., Yang, M., and Gao, X. Self-supervised contrastive attributed graph clustering. *arXiv preprint arXiv:2110.08264*, 2021.

Xie, X., Chen, W., Kang, Z., and Peng, C. Contrastive graph clustering with adaptive filter. *Expert Systems with Applications*, 219:119645, 2023.

Yang, L., Li, M., Liu, L., Niu, B., Wang, C., Cao, X., and Guo, Y. Diverse message passing for attribute with heterophily. In *Advances in Neural Information Processing Systems 34, NeurIPS 2021,*, pp. 4751–4763, 2021.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Zhang, X., Liu, H., Li, Q., and Wu, X.-M. Attributed graph clustering via adaptive graph convolution. pp. 4327–4333, 2019.

Zhu, H. and Koniusz, P. Simple spectral graph convolution. In *9th International Conference on Learning Representations, ICLR 2021,*, 2021.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

Zhu, P., Li, J., Wang, Y., Xiao, B., Zhao, S., and Hu, Q. Collaborative decision-reinforced self-supervision for attributed graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.