

## Article

# Preserving Global Information for Graph Clustering with Masked Autoencoders

Rui Chen

School of Computer and Network Security, Chengdu University of Technology, Chengdu 610059, China; chen.rui2@student.zy.cdut.edu.cn

**Abstract:** Graph clustering aims to divide nodes into different clusters without labels and has attracted great attention due to the success of graph neural networks (GNNs). Traditional GNN-based clustering methods are based on the homophilic assumption, i.e., connected nodes belong to the same clusters. However, this assumption is not always true, as heterophilic graphs are also ubiquitous in the real world, which limits the application of GNNs. Furthermore, these methods overlook global positions, which can result in erroneous clustering. To solve the aforementioned problems, we propose a novel model called Preserving Global Information for Graph Clustering with Masked Autoencoders (GCMA). We first propose a low-high-pass filter to capture meaningful low- and high-frequency information. Then, we propose a graph diffusion method to obtain the global position. Specifically, a parameterized Laplacian matrix is proposed to better control the global direction. To further enhance the learning ability of the autoencoders, we design a model with a masking strategy that enhances the learning ability. Extensive experiments on both homophilic and heterophilic graphs demonstrate GCMA's advantages over state-of-the-art baselines.

**Keywords:** graph clustering; graph neural network; self-supervised learning; graph representation

**MSC:** 05C62; 62H30; 68-04



**Citation:** Chen, R. Preserving Global Information for Graph Clustering with Masked Autoencoders. *Mathematics* **2024**, *12*, 1574. <https://doi.org/10.3390/math12101574>

Academic Editor: Jianping Gou

Received: 11 April 2024

Revised: 7 May 2024

Accepted: 14 May 2024

Published: 17 May 2024



**Copyright:** © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Due to the powerful representation learning ability of graph structures, graph neural networks (GNNs) have achieved great success in numerous applications, such as node classification [1,2], graph clustering [3,4] and more. GNNs can effectively learn node representations by aggregating information from neighboring nodes. This capability enables the accurate classification of nodes into predefined categories. By leveraging the inherent connectivity patterns within graphs, GNNs can identify clusters or communities of related nodes. This is particularly useful for understanding the underlying structures of complex networks. Their ability to capture both local and global dependencies within graphs has revolutionized the field of graph-based machine learning. Among them, deep graph clustering is a basic but challenging unsupervised task that has attracted much attention recently. Clustering aims to divide data into different groups without labels and is applied in many areas such as recommendation systems and biological systems.

Most graph clustering methods are GNN-based and aim to derive a low-dimensional representation of nodes while incorporating graph structures. And self-supervised learning is a popular concept for obtaining such representations and can generally be categorized into generative learning and contrastive learning. Among them, contrastive learning methods always apply an augmentation approach to obtain positive and negative samples, ensuring that positive samples are brought closer while negative samples are pushed away. SCGC [5] and CCGC [3] are the most recent contrastive methods and have achieved dominant performance. Generative learning often involves using a graph autoencoder (GAE) [6] and is commonly paired with classical clustering techniques like

K-means. However, refs. [7,8] suggest that by following a straightforward graph reconstruction principle, GAE may excessively prioritize proximity information, which is not always advantageous for self-supervised learning.

Masking strategies for autoencoders have achieved great success in both natural language processing (NLP) and computer vision (CV) domains. Notable examples include BERT [9] and MAE [10], respectively. Interestingly, masked autoencoders should also be well-suited for graph data, as both edges and nodes can be easily masked or unmasked for self-supervised learning, as demonstrated by existing studies [11]. While the strong generalization ability of masking is conducive for learning effective graph embeddings, few studies have specifically applied it to clustering tasks.

Further, traditional clustering methods are based on the homophily assumption, i.e., connected nodes are from the same cluster, which is not always true. As a matter of fact, graphs always contain a wide range of frequencies [12], and high-frequency components are paramount in many tasks. However, existing GNN-based clustering methods only focus on low-frequency information in graphs, inevitably overlooking some discriminative information. This limitation restricts their applications in real-world scenarios. Homophily is always unknown for an unsupervised learning task. Thus, in these methods, heterophily can result in the smoothing operation generating similar representations for nodes with different labels, which can lead to poor clustering performance of GNNs. Real-world graphs are not always assortative: they are sometimes disassortative. For example, in dating networks, individuals of the opposite sex are more likely to make connections, while in protein structures, different compounds are more inclined to connect. In such scenarios, performance will be hampered if we use a low-pass filter to enforce similar representations among connected nodes. Therefore, high-frequency components are valuable for making the representations more discriminative. Focusing solely on low-frequency information can restrict practical application using real-world graphs.

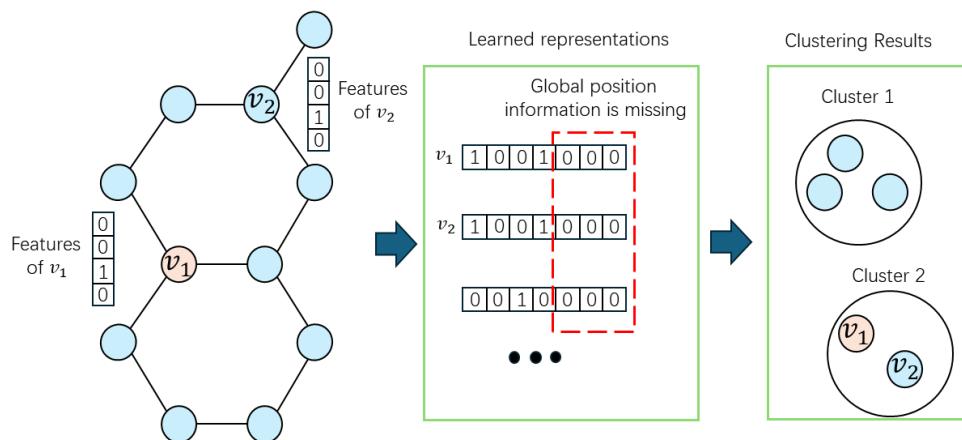
Further, traditional graph clustering methods only focus on the original features and local neighbor information by stacking 2–4 layers for the GNN, which results in a failure to capture global information [13]. A toy example illustrating the significance of global information can be seen in Figure 1. Different colors indicate various types of nodes. Although nodes  $v_1$  and  $v_2$  share similar original features and two-hop neighbors, their global information can be very different. Previous clustering methods, which ignore global information, will mistakenly classify them into one cluster. This issue arises because the methods learn the representations of two nodes using only local information, neglecting the nodes' relative positions within the graph. For instance, in social networks, where individuals form communities based on shared interests or affiliations, capturing global patterns is vital for identifying cohesive groups and understanding their interactions.

To enable deeper convolution, several studies have attempted to devise various architectures [13,14], including initial residual networks and normalization techniques. For graph clustering, autoencoders are widely applied architectures that are trained to reconstruct input data; autoencoders often have a bottleneck layer for dimensionality reduction. Variants try to make the GNN encoder layer deeper to capture more neighbors [15]. However, these approaches are still limited to local neighbors, leaving the challenge of effectively capturing global information unresolved.

To address the aforementioned problems, we propose a method called Preserving Global Information for Graph Clustering with Masked Autoencoders (GCMA). Firstly, we propose a low–high-pass filter to capture meaningful low- and high-frequency information. After that, we propose a graph diffusion method for the global position representation of each node and preserve it in the node's features. To better control the global direction, we propose a parameterized Laplacian matrix. To enhance the model's learning ability, we design a masked autoencoder model. Capturing global patterns is crucial for understanding complex interdependencies within the graph. In this context, masked autoencoders offer a promising solution by selectively preserving global information while filtering out noise, thus improving the learning ability for the global information. This paper addresses the

necessity of global information preservation in graph clustering tasks and explores the effectiveness of masked autoencoders for this purpose. To sum up, our contributions are as follows:

- To improve the learning ability of autoencoders, we propose to use a mask technique on the original information and encoded features. To the best of our knowledge, we give the first attempt to apply masked autoencoders to the graph clustering task.
- We propose a graph diffusion method to obtain a node's global position representation and preserve it in the node's features. We also propose a parameterized Laplacian matrix to make the global direction adaptive.
- We design a low–high-pass filter that screens important low-frequency and high-frequency information conveyed by the data. A discriminative representation is obtained by graph filtering, which encodes topological structure information into features.
- Experimental results on both homophilic and heterophilic datasets suggest that the proposed method outperforms existing graph clustering methods, including recent self-supervised learning methods.



**Figure 1.** A toy example to show the importance of global information. Different colors indicate different groups. Nodes  $v_1$  and  $v_2$  share similar neighborhood patterns and original features but have totally different global information. Traditional clustering methods will group them into one cluster.

## 2. Related Work

### 2.1. Graph Clustering

Graph clustering aims to divide data into different groups in an unsupervised manner, which is a fundamental yet challenging task. These methods can be simply divided into three kinds: generative approaches, adversarial approaches and contrastive approaches. Our method belongs to the first kind.

Generative approaches are always based on autoencoders and reconstruct an adjacency matrix or features. They can efficiently incorporate both structure and content information into an attributed graph. For example, graph autoencoder (GAE) and variational GAE (VGAE) [16] use autoencoders to reconstruct the adjacency matrix, while marginalized GAE (MGAE) [17] is proposed for reconstructing node features. To include more hop neighbors, SDCN [15] transfers the encoded information and captures structural nuances at higher orders. Additionally, the aforementioned method involves two separate steps: clustering and graph embedding, which are conducted independently. There is a possibility that the learned representations may not be suitable for subsequent clustering tasks. To tackle this issue, AGC [18] proposes a goal-directed graph attentional autoencoder architecture. Specifically, an attention network is utilized to assess the importance of neighbors to a target node, and an inner product decoder is trained to reconstruct the graph structure. However, this approach still exhibits high complexity in terms of time and space. In another study, GMM-VGAE [19] introduces Gaussian mixture models to the variational graph autoencoder (VGAE) to capture the inherent complex data distributions, leading to the

development of a unified end-to-end learning model for graph clustering. Additionally, DNENC [20] proposes a neighbor-aware GAE to gather information from neighbors, and it employs an end-to-end learning strategy. Recently, several contrastive clustering methods have also been proposed. Contrastive learning aims to learn discriminative features of data by maximizing the similarity of positive pairs and minimizing the distance of negative pairs. It demonstrates significant performance, even surpassing supervised learning methods. For instance, SimCLR [21] constructs self-supervised samples using data augmentation by treating images from different views in a mini-batch as negative pairs, while BYOL [22] operates without negative samples. In the realm of graphs, several contrastive learning methods have emerged. GraphCL [23] proposes four data augmentation types to learn unsupervised representations of graph data, while MVGRL [7] compares two diffusion matrices converted from the adjacency matrix for representation learning. GCA [24] introduces a joint data augmentation scheme adaptable to both graph structure and attributes. However, many of these methods assume that the given graph is reliable, and they rely on data augmentation to identify positive pairs. Much work has focused on contrastive graph learning by maximizing mutual information (MI) between node and graph representations, yielding state-of-the-art results. For instance, deep graph infomax (DGI) [25] enhances node representations to capture more global information, while contrastive clustering [26] designs a dual contrastive learning paradigm operating at instance and cluster levels. SCAGC [27] employs a clustering module to generate clustering labels by comparing the representations of distinct clusters, and DCRN [28] proposes a dual network to reduce information correlation and applies K-means on embeddings. Most recently, CGC [12] proposes a graph-level contrastive regularizer to refine the graph structure. CCGC [3] proposes to improve the quality of negative samples. SCGC [5] proposes a simple augmentation method with only MLP.

However, these clustering methods only focus on feature and local structure information, which ignores the global position information.

## 2.2. Graph Masked Autoencoder

Masking is often used for data augmentation during training. By randomly masking certain parts of input data (such as words in sentences or pixels in images), models learn to be more robust and generalize better. The learning objective of a masked autoencoder is to conceal segments of the input data and predict the obscured content, constituting a form of self-supervised learning. Although the masked autoencoder technique has achieved great success in the CV and NLP areas, it has received comparatively less attention within the domain of GNN. In CV [10], masking is often used for image inpainting tasks, where certain regions of an image are intentionally obscured or “masked” out. This masking process involves replacing the pixel values in the masked regions with special placeholders or zeros, effectively removing the information from those areas. The goal is to train a model, such as a convolutional neural network or a generative adversarial network, to predict or “fill in” the missing information in the masked regions based on the surrounding context. This process helps the model learn to understand and reconstruct visual content, enabling tasks such as image completion and restoration. In NLP, masking is commonly used in language model pretraining tasks such as BERT [9]. During pretraining, a certain percentage of tokens in the input text are randomly selected and replaced with a special masked token. The model is then trained to predict the original tokens based on the context provided by the surrounding tokens. This masked language model pretraining allows the model to learn contextual representations of words or subwords, capturing syntactic and semantic relationships within the text. MGAE [29] attempts to apply the masking approach to the edges of the graph structure, using it as a form of self-supervised learning. MaskMAE [30] introduces novel one-sided and path mask strategies, accompanied by theoretical analysis of the side mask strategy. GiGaMAE [31] also employs masking techniques on both nodes and edges. However, these methods rely on labels to train the model and lack improvement in the graph clustering task.

### 2.3. Graph Filtering

Many research studies have indicated that smooth graph signals contain more low-frequency signals than high-frequency ones [32]. The pioneering works of [33,34] employ filters in the Fourier space for graph structures. In recent times, there has been widespread adoption of low-pass filters to derive smooth representations [32,35]. However, these methods often overlook meaningful high-frequency information, treating it as noise. Some studies have emphasized the significance of high-frequency components, suggesting that they contain valuable information [36]. AutoGCN [37] utilizes three graph convolutional filters to capture meaningful information at different frequencies while imposing a constraint on filter functions to have linear and quadratic complexity. Similarly, refs. [37,38] emphasize the significance of the high-frequency band of graph signals and design flexible GNN models that excel on both homophilic and heterophilic graphs for node classification. However, these approaches rely on neural networks. In contrast, we propose a simple and shallow method to capture both meaningful low- and high-frequency information.

## 3. Methodology

### 3.1. Notation

To begin, we define a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{X})$  with node set  $\mathcal{V}$ , feature matrix  $\mathcal{X}$  and adjacency matrix with  $N$  nodes  $\mathcal{A} \in \mathbb{R}^{N \times N}$ . If node  $i$  and node  $j$  are connected, then  $A_{ij} = 1$ ; otherwise,  $A_{ij} = 0$ . Its symmetric normalized adjacency matrix with a self-loop is  $\tilde{\mathcal{A}} = D^{-\frac{1}{2}}(\mathcal{A} + I)D^{-\frac{1}{2}}$ . The normalized graph Laplacian is given by  $L = I - \tilde{\mathcal{A}}$ . Let  $X_i$  represent the  $i$ -th row of  $\mathcal{X}$ .

### 3.2. Graph Filtering

Many works [1,39] consider the feature matrix as graph signals and propose smoothing it with graph filters. Motivated by their success, we propose a low-high-pass filter to perform neighbor information aggregation as an independent learning step before training. This approach efficiently captures both low- and high-frequency information.

Let  $L = U\Lambda U^\top$  be the eigendecomposition, where  $\Lambda = \text{diag}[\mu_1, \mu_2, \dots, \mu_N]$  is the diagonal matrix of eigenvalues, and  $\mu_i \in [0, 2]$ . A spectral filter on graph signal  $\mathcal{X}$  can be expressed as follows:

$$\begin{aligned} g(L)\mathcal{X} &= Ug(\Lambda)U^\top\mathcal{X} \\ &= U\text{diag}[g(\mu_1), g(\mu_2), \dots, g(\mu_N)]U^\top\mathcal{X}, \end{aligned} \tag{1}$$

where  $g(\mu)$  represents the filter function. The choice of  $g(\mu)$  is crucial for graph filtering. Most graph clustering methods set  $g(\mu) = 1 - \frac{\mu}{2}$  to smooth the signals of neighboring nodes, which is considered to be a low-pass filter. Let  $H$  represent the smoothed signals. It is easy to have:

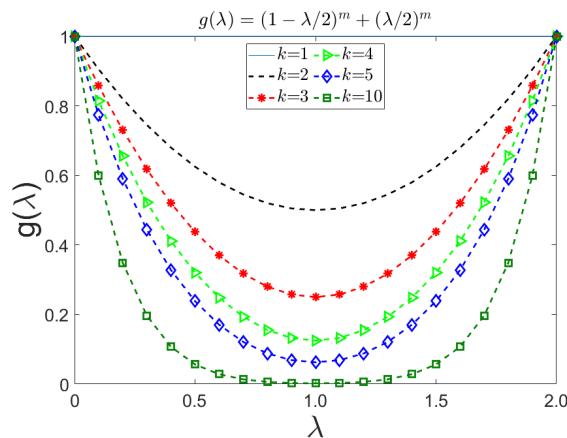
$$\begin{aligned} H &= U\text{diag}\left[(1 - \frac{\mu_1}{2})^m, (1 - \frac{\mu_2}{2})^m, \dots, (1 - \frac{\mu_N}{2})^m\right]U^\top\mathcal{X} \\ &= (I - \frac{L}{2})^m\mathcal{X}, \end{aligned} \tag{2}$$

where  $m$  is the filter order. Through the proposed low-pass filter, high-frequency components in node features will be filtered out.

However, filtering with only a low-pass filter treats all high frequencies as noise, which fails to capture meaningful high-frequency information. To have a low-high-pass filter, i.e., the low frequency and high frequency dominate the middle frequency, we combine a high-pass filter and a low-pass filter:

$$H = [(I - \frac{L}{2})^m + (\frac{L}{2})^m]\mathcal{X}. \tag{3}$$

This filter is flexible and can adapt to both homophilic and heterophilic graphs. In Figure 2, we plot  $g(\lambda)$  with different values of  $m$ , which leads to different types of filters. A large  $m$  represents more variation, where distant neighbors could be helpful. The high-pass filter pushes a node's feature vector away from those of its neighbors, while the low-pass filter does the opposite. It can be seen that the proposed filter can capture both low- and high-frequency information. In this way, homophilic neighbors tend to be similar, while heterophilic neighbors tend to be different in  $H$  [12].



**Figure 2.** Some examples of our proposed low–high-pass filter.

### 3.3. Global Position Encoding

To achieve finer control over directional information, we define a new class of Laplacian matrices, i.e., the parameterized normalized Laplacian matrix:

$$L(\beta) = D^{-\beta} L D^{\beta-1}, \quad (4)$$

and we define the corresponding parameterized normalized adjacency matrix:

$$A(\beta) = I - L(\beta), \quad (5)$$

where the parameter  $\beta \in [0, 1]$ .

Graph diffusions refer to classic models underlying the centrality, influence and similarity of nodes. Graph diffusion propagates along reachability searches. Traditional graph diffusions can be seen as spacial cases of Equation (5). For example, when  $\beta = \frac{1}{2}$ ,  $A$  is symmetric normalized, i.e.,  $A = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ ; when  $\beta = 1$ ,  $A$  is random-walk normalized, i.e.,  $A = D^{-1} A$ . Inspired by [40], we consider these graph diffusions as PageRank, and global position encoding is defined as  $k$ -steps of graph diffusions:

$$B_i = [A(\beta)_{ii}, A^2(\beta)_{ii}, \dots, A^k(\beta)_{ii}] \in \mathbb{R}^k. \quad (6)$$

The size of the neighborhood influencing each node can be adjusted through the teleport probability  $\beta$ . The flexibility in selecting  $\beta$  allows us to customize the model for different types of networks. Diverse graph structures require the consideration of varying neighborhood sizes. Different from [40], which utilizes the full matrix of  $A$  for all pairs of nodes, we employ a low-complexity approach to the parameterized normalized adjacency matrix by only considering the probability of a node  $i$  landing on itself, i.e.,  $A(\beta)_{ii}$ . The global direction can be controlled by  $\beta$ . This approach offers a distinctive node representation by assuming that each node has a unique  $k$ -hop topological neighborhood for a sufficiently large  $k$ . However, this assumption can be debated. For example, when analyzing strongly regular graphs, each node in a graph has the same positional encoding for any given  $k$  value, as they are isomorphic by construction. Despite having identical position encodings for all nodes within a graph, these positional encodings are unique

for each class of isomorphic graphs, leading to accurate classification. Based on the above analysis,  $B_i$  can provide a global view of node  $i$ 's positions.

### 3.4. Graph Masked Autoencoder

#### 3.4.1. Graph Mask Encoder

Define the masked graph as  $\mathcal{G}' = \text{Mask}(\mathcal{G})$ . We then input  $\mathcal{G}'$  into the encoder and learn a representation in a low-dimensional space. Finally, the decoder is applied to reconstruct the graph information. Similar to GiGaMAE, we use GCN [1] and GAT [41] as encoders and MLP as the decoder.

At the masking stage, only masking and reconstructing either  $H$  or  $A$  limits the model's ability to learn from the other, thus hindering the acquisition of comprehensive information. However, simultaneously masking both can detrimentally affect model learning, as it may result in insufficient information for graph reconstruction. Thus, we propose to use two models to learn the information. Further, GNNs with  $H$  and  $A$  learn the representations of nodes with only local information, overlooking their relative positions within the graph. Thus, we preserve the global position  $B$  in the model. Masking in autoencoders selectively preserves global information while filtering out noise, thus improving the learning ability for the global information. To begin with, we represent the random masking process as  $P$ .  $P = \{P_e, P_f, P_p\}$ , where  $P_e \in \{0,1\}^{N \times N}$ ,  $P_f \in \{0,1\}^d$  and  $P_p \in \{0,1\}^k$  are edge, feature and position mask matrices, respectively. Then, we have  $P(\mathcal{G} = (\mathcal{V}, A, H)) = \{\mathcal{V}, A * P_e, H \text{diag}(P_f) \oplus B \text{diag}(P_p)\} = \{\mathcal{V}, A', H'\} = \mathcal{G}'$ , where  $A'$ ,  $H'$ ,  $*$  and  $\oplus$  indicate the masked feature, masked adjacency matrix, Hadamard product and concatenation. We input them into the encoder with two models and obtain node representations  $S_1$  and  $S_2$ , i.e.,

$$\begin{aligned} S_1 &= \text{GCN}(\mathcal{G}) \\ S_2 &= \text{GAT}(\mathcal{G}). \end{aligned} \quad (7)$$

#### 3.4.2. Graph Mask Decoder

Various methods concentrate on distinct facets of graph data. For instance, GAE emphasizes learning the graph structures, whereas AE with  $X$  primarily focuses on node attributes. Hence, we contemplate embedding reconstruction methods that aim to amalgamate multiple models. The loss function is formulated based on mutual information (MI).  $S_n \in \mathbb{R}^{N \times d_n}$  indicates the  $n$ -th reconstruction target. Information from different modalities is stored in the chi-square continuous embedding space.

In graph clustering, the decoder reconstructs relatively fewer informative multi-dimensional node features. Traditional GAEs employ either no neural decoders or a simple MLP for decoding with less expressiveness, causing the latent code to be nearly identical to the input features. In order to incentivize the encoder to acquire compressed representations even further, GgraphMAE [11] proposes a re-masking technique to handle the latent representations during decoding. Note that the global information is encoded in latent space; thus, re-masking can further improve the learning ability for global information during reconstruction. Denote the masking process as  $P_d \in \{0,1\}^d$ . Then  $S_n$  becomes  $Z_n$ . Each  $Z_n$  is then fed into a multilayer perceptron (MLP). So we have  $\tilde{Z}_n = \text{MLP}(Z_1 + Z_2) \in \mathbb{R}^{N \times (d+k)}$ .

#### 3.4.3. Loss Function

During target reconstruction, the  $l_2$ -norm loss considers the characteristics of each target individually, disregarding their interrelationships. Conversely, cross-entropy is more suitable for discrete variables. Given that we are calculating the reconstruction loss involving multiple targets, it is crucial to capture as much pertinent information as possible from the targets. Therefore, we opt to use MI as the basis for calculating the loss. Our reconstruction loss is defined as:

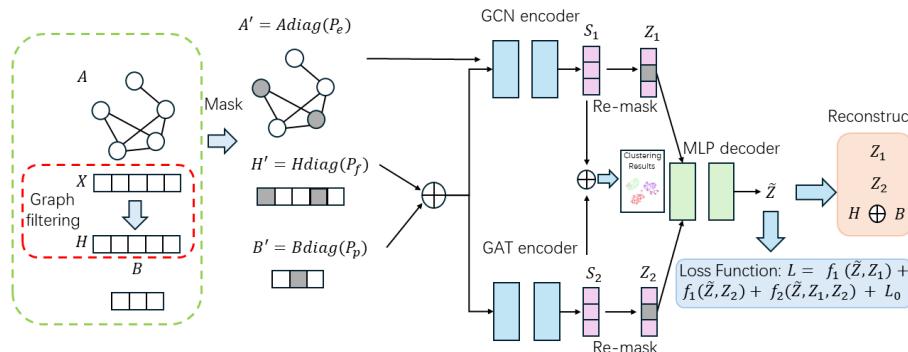
$$f_1(x, y) = \exp\left(\frac{x \times y}{\|x\| \times \|y\|}\right) \quad (8)$$

$$f_2(x, y, z) = \exp\left(\frac{x \times (y \oplus z)}{\|x\| \times \|y \oplus z\|}\right)$$

To reconstruct the original input features, we use the loss  $L_0 = \sum_{i=1}^n \|\tilde{Z}_i - H_i \oplus B_i\|$ . Finally, our total loss is defined as:

$$L = f_1(\tilde{Z}, Z_1) + f_1(\tilde{Z}, Z_2) + f_2(\tilde{Z}, Z_1, Z_2) + L_0 \quad (9)$$

Our result is from performing K-means on  $S_1 + S_2$ . The overall model can be seen in Figure 3.



**Figure 3.** Overall model.

## 4. Experiments

### 4.1. Datasets

To verify our proposed method, we test it on five widely used benchmark datasets, which are Cora, CiteSeer, PubMed [1], Wiki [42] and Large Cora [43]. Cora, CiteSeer, PubMed and Large Cora are citation networks, while Wiki is a webpage network. To further test our model in real-world applications, we evaluate the proposed model for clustering tasks on heterophilic graph datasets, including webgraphs from WebKB datasets (<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/>, (accessed on 14 May 2024)): Wisconsin, Cornell, Texas, Chameleon, and Wikipedia networks Squirrel [44].

The homophily of a graph is computed as:

$$\text{Homophily} = \frac{1}{N} \sum_{v \in \mathcal{V}} \alpha_v \text{ and } \alpha_v = \frac{|\{u \in \mathcal{N}_v | \ell(u) = \ell(v)\}|}{|\mathcal{N}_v|}, \quad (10)$$

where  $\ell(u)$  indicates the label of node  $u$ . A large value  $\text{Homophily}$  indicates more connected nodes from the same class. The details of these datasets are shown in Table 1.

**Table 1.** The statistics of all datasets.

Dataset	Nodes	Edges	Features	Classes	Homophily
Cora	2708	5429	1433	7	0.83
CiteSeer	3327	4732	3703	6	0.71
PubMed	191,717	44,338	500	3	0.79
Wiki	2405	17,981	4973	17	0.46
Large Cora	11,881	64,898	3780	10	0.73
Squirrel	5201	217,073	2089	5	0.22
Chameleon	2277	2325	31,371	5	0.23
Wisconsin	251	515	1703	5	0.16
Cornell	183	298	1703	5	0.11
Texas	183	325	1703	5	0.06

### 4.2. Baselines

We compare GCMA with recent popular clustering methods. SCI [42] integrates network topology with semantic information to detect communities and infer their meanings

simultaneously. ARGE and ARVGE [45] propose an adversarial graph embedding framework that encodes both the topological structure and node content in a graph into a compact representation. GAE, VGAE [16] and MGAE [17] are traditional graph autoencoder methods. AGC [32] leverages high-order graph convolutions to capture global cluster structures and adaptively select the appropriate order for different graphs. DAEGC [18] combines attention networks to capture both the topological structure and node content. SDCN [15] proposes an autoencoder to capture more structural information. ARGA\_AX and ARVGA\_AX [46] are adversarial methods that focus on latent code distributions. DGI [25] maximizes the mutual information between patch representations and high-level summaries. VGAE with Gaussian mixture models (GMM-VGAE) [19] combines an autoencoder with a semi-supervised module. DNENC-Att (with a graph attentional autoencoder) and DNENC-Con (with a graph convolutional autoencoder) [20] are the most recent autoencoder methods. FGC [35] and CGC [12] are the most recent shallow methods and apply high-order structure and contrastive learning ideas, respectively, as regularizers. CCGC [3] and SCGC [5] are the most recent contrastive methods and propose a new augmentation strategy.

#### 4.3. Setup

The same as our baselines, we use grid search to find the best performance. The learning rate is fixed to  $1 \times 10^{-2}$ . The maximum number of epochs is set to 200, and training stops when the model converges. GCMA has four hyperparameters:  $m$ ,  $\beta$ ,  $k$  and the mask ratio. The variable  $m$  is searched in  $[1, 2, 3, 4, 5]$ . We set  $m = 5$  on Cora and CiteSeer,  $m = 3$  on Large Cora and PubMed, and  $m = 2$  on the other datasets. The variable  $\beta$  is searched in  $[0.1, 0.5, 0.7]$ . We set  $\beta = 0.1$  on CiteSeer and Chameleon,  $\beta = 0.7$  on Large Cora and Wiki, and  $\beta = 0.1$  on other datasets. It can be seen that datasets with large neighborhood sizes prefer a relatively large  $\beta$ . The variable  $k$  is fixed at 15 since most homophilic neighbors can be included in 15 hops according to [47]. The mask ratio is fixed at 30% for  $P_f$ ,  $P_p$  and  $P_e$ . We use three metrics that are popular for clustering studies, including clustering accuracy (ACC), normalized mutual information (NMI) and adjusted Rand index (ARI). Their definitions are:

$$\text{ACC} = \frac{\sum_{i=1}^n \delta(\text{map}(l_i) = y_i)}{n}, \quad (11)$$

$$\text{NMI}(Y, L) = \frac{2I(Y; L)}{H(Y) + H(L)}, \quad (12)$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (13)$$

where  $\delta$  represents an indicator function,  $l_i$  signifies the assigned label for node  $i$ ,  $y_i$  denotes the ground truth label of node  $i$ ,  $\text{map}$  is a transformation function that maps  $l_i$  to its group label based on the Kuhn–Munkres algorithm,  $H(Y)$  represents the entropy of  $Y$ , and  $I(Y; L)$  denotes the mutual information between two discrete variables. These metrics provide quantitative measures of clustering quality, allowing for comparison between different clustering algorithms and parameter settings.

#### 4.4. Results Analysis

The clustering results for homophilic graphs are presented in Table 2. It can be seen that our method demonstrates superior performance in most cases. For the methods based on autoencoders, such as DNENC-Att, DNENC-Con and GMM-VGAE, GCMA can surpass them by a large margin. This demonstrates that the masking strategy can enhance the learning ability of the autoencoders. For the methods that utilize the contrastive learning idea, such as CGC, CCGC and SCGC, they generally perform better than other autoencoder methods. This indicates that contrastive methods continue to dominate in self-supervised methods. However, GCMA can surpass them in most cases, confirming the validity of our method. GCMA combines the strengths of graph convolutional networks with masked autoencoders, effectively preserving global information while filtering noise. By selectively

propagating relevant global features, GCMA produces more informative representations, leading to improved clustering performance. Note that GCMA fails to achieve the best performance only on the NMI metric for PubMed and the F1 metric for Large Cora. This may be because these two datasets are too sparse, which results in less information being learned from masked edges. In general, the experimental results successfully verify the effectiveness of GCMA compared to recent clustering methods on homophilic graphs.

**Table 2.** Clustering performance on homophilic graphs. The best performance is marked in **bold**.

Methods	Cora			CiteSeer			PubMed			Wiki			Large Cora		
	ACC%	NMI%	F1%												
SCI [42]	41.21	21.57	11.82	33.45	9.77	18.01	44.89	5.99	35.73	32.72	26.38	19.03	26.78	11.31	7.68
ARGE [45]	64.00	44.90	61.90	57.30	35.00	54.60	59.12	23.17	58.41	41.40	39.50	38.27	-	-	-
ARVGE [45]	63.80	45.00	62.70	54.40	26.10	52.90	58.22	20.62	23.04	41.55	40.01	37.80	-	-	-
GAE [16]	53.25	40.69	41.97	41.26	18.34	29.13	64.08	22.97	49.26	17.33	11.93	15.35	-	-	-
VGAE [16]	55.95	38.45	41.50	44.38	22.71	31.88	65.48	25.09	50.95	28.67	30.28	20.49	-	-	-
MGAE [17]	63.43	45.57	38.01	63.56	39.75	39.49	43.88	8.16	41.98	50.14	47.97	39.20	38.04	32.43	29.02
AGC [32]	68.92	53.68	65.61	67.00	41.13	62.48	69.78	31.59	68.72	47.65	45.28	40.36	40.54	32.46	31.84
DAEGC [18]	70.40	52.80	68.20	67.20	39.70	63.60	67.10	26.60	65.90	38.25	37.63	23.64	39.87	32.81	19.05
SDCN [15]	60.24	50.04	61.84	65.96	38.71	63.62	65.78	29.47	65.16	-	-	-	-	-	-
ARGA_AX [46]	59.70	45.50	57.90	54.70	26.30	52.70	63.70	24.50	63.90	-	-	-	-	-	-
ARVGA_AX [46]	71.10	52.60	69.30	58.10	33.80	52.50	64.00	23.90	64.40	-	-	-	-	-	-
DGI [25]	71.81	54.09	69.88	68.60	43.75	64.64	-	-	-	44.37	42.20	40.16	-	-	-
GMM-VGAE [19]	71.50	54.43	67.76	67.44	42.30	63.22	71.03	30.28	69.74	-	-	-	-	-	-
EVGC [48]	72.95	55.76	71.01	67.02	41.89	62.89	70.80	35.63	70.32	51.46	49.37	45.14	-	-	-
DNENC-Att [20]	70.40	52.80	68.20	67.20	39.70	63.60	67.10	26.60	65.90	-	-	-	-	-	-
DNENC-Con [20]	68.30	51.20	65.90	69.20	42.60	63.90	67.70	27.50	67.50	-	-	-	-	-	-
FGC [35]	72.90	56.12	63.27	69.01	44.02	64.43	70.01	31.56	69.10	51.10	44.12	34.79	48.25	35.24	35.52
CGC [12]	75.15	56.90	66.22	69.31	43.61	64.74	67.43	33.07	67.14	59.04	53.20	45.43	50.18	34.10	43.79
CCGC [3]	73.88	55.56	70.98	69.84	44.33	62.71	68.32	31.08	68.84	60.21	54.54	44.32	51.21	34.55	44.89
SCGC [5]	73.88	56.10	70.81	70.81	45.25	64.80	67.76	33.82	68.23	60.43	53.76	44.56	51.09	34.02	44.56
GCMA	<b>76.12</b>	<b>57.21</b>	<b>71.43</b>	<b>71.95</b>	<b>45.98</b>	<b>65.21</b>	<b>72.04</b>	<b>33.45</b>	<b>71.04</b>	<b>61.32</b>	<b>55.43</b>	<b>45.80</b>	<b>51.85</b>	<b>36.01</b>	<b>44.23</b>

The clustering results on heterophilic graphs are presented in Table 3. It can be seen that GCMA and CGC have clear advantages over other methods. Similar to GCMA, CGC also considers high-frequency information, making it significant for heterophilic graphs. This conclusion is similar to semi-supervised learning. However, GCMA can outperform CGC by a significant margin thanks to the global positional information we utilize. Note that heterophilic graphs are also noisy. Therefore, the results show that the proposed filter and masked autoencoders in GCMA provide robustness to noisy and variable graph data by learning compact representations while filtering out irrelevant features. This robustness ensures more reliable and interpretable clustering results, even in the presence of noise. In general, GCMA can successfully be applied to both homophilic and heterophilic graphs, which is more practical than methods that only focus on one type of graph structure.

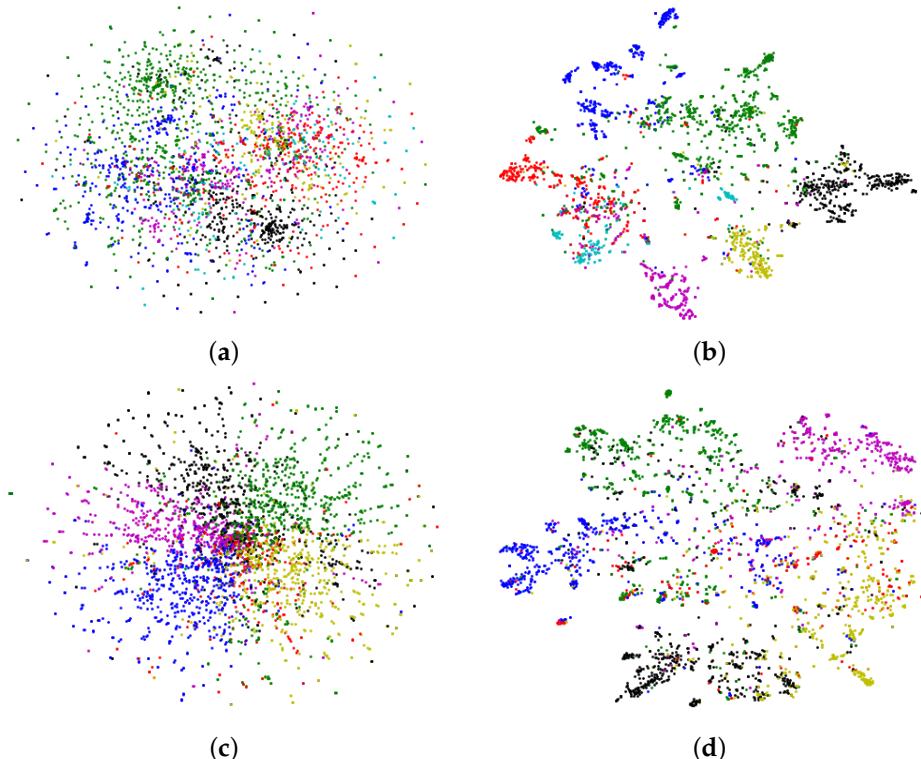
**Table 3.** Clustering performance on heterophilic graphs. The best performance is marked in **bold**.

Methods	Squirrel			Chameleon			Wisconsin			Cornell			Texas		
	ACC%	NMI%	F1%	ACC%	NMI%	F1%	ACC%	NMI%	F1%	ACC%	NMI%	F1%	ACC%	NMI%	F1%
DAEGC [18]	25.55	2.36	24.07	31.08	7.89	9.17	39.62	12.02	6.22	42.56	12.37	30.20	45.99	11.25	18.09
ARVGA-Col-M [49]	-	-	-	-	-	-	54.34	11.41	-	-	-	-	59.89	16.37	-
RWR-Col-M [49]	-	-	-	-	-	-	53.58	16.25	-	-	-	-	57.22	13.82	-
FGC [35]	25.11	1.32	22.13	34.21	11.31	9.05	50.19	12.92	25.93	44.10	8.60	<b>32.68</b>	53.48	5.16	17.04
CGC [12]	27.23	2.98	20.57	36.31	11.21	12.97	55.85	23.03	27.29	44.62	14.11	21.91	61.50	21.48	27.20
CCGC [3]	26.03	1.78	20.21	33.21	10.98	10.22	53.31	22.67	25.81	45.54	13.62	21.66	60.98	19.34	22.73
SCGC [5]	26.76	1.98	20.99	34.11	11.37	10.67	54.34	21.57	25.81	45.76	13.53	22.09	61.22	19.77	21.94
GCMA	<b>30.15</b>	<b>4.21</b>	<b>24.83</b>	<b>39.97</b>	<b>15.32</b>	<b>14.33</b>	<b>56.95</b>	<b>24.12</b>	<b>29.11</b>	<b>46.30</b>	<b>15.76</b>	<b>22.78</b>	<b>63.20</b>	<b>22.01</b>	<b>29.22</b>

#### 4.5. Ablation Study

To further verify our proposed model, we test the model by deleting key components. Firstly, we test our method without a low-high-pass filter and name it GCMA *w/o f*. The results are illustrated in Table 4. It can be seen that the performance deteriorates in all cases; therefore, low-pass filtering is beneficial for training. Furthermore, we test our method with

only one model. We refer to the one without GAT as GCMA *w/o GAT* and the one without GCN as GCMA *w/o GCN*. The results in Table 4 show that the performance degrades dramatically. This result is consistent with our previous conclusion that masking both  $A$  and  $X$  with only a single model results in a lack of information. Further, we test our method without the global position and name it GCMA *w/o G*. The results are in Table 4. It can be seen that the global position is very helpful in most cases. Finally, we visualize the Cora and CiteSeer representations with GCMA using the t-SNE technique in Figure 4. Compared with the original features, it can be seen that GCMA has a clear advantage in identifying clusters.



**Figure 4.** Visualization of Cora and CiteSeer with GCMA. (a) Cora's original features. (b) Cora's representations with GCMA. (c) CiteSeer's original features. (d) CiteSeer's representations with GCMA.

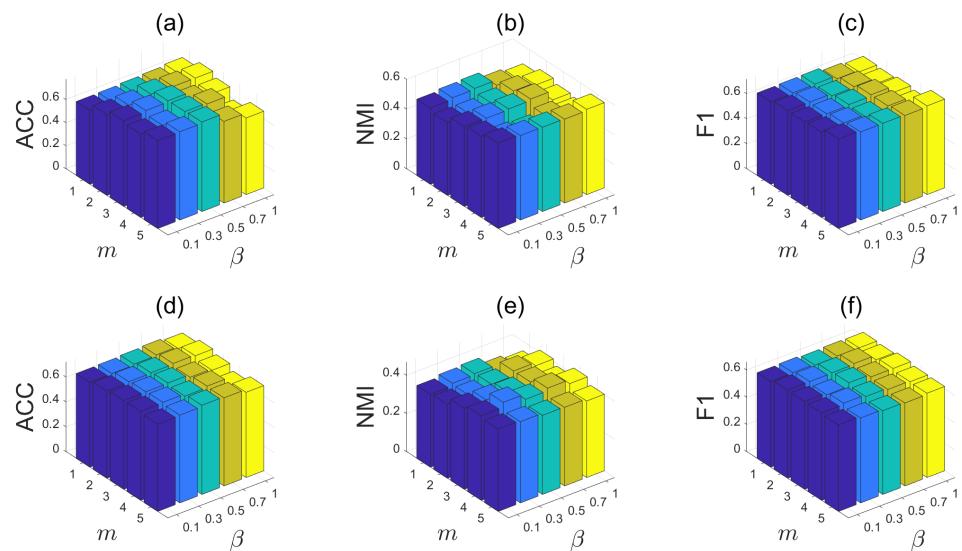
**Table 4.** Results of ablation study. The best performance is marked in **bold**.

Methods	Cora			CiteSeer			PubMed			Wiki			Large Cora		
	ACC%	NMI%	F1%												
GCMA <i>w/o f</i>	72.74	50.83	64.04	68.88	42.97	62.93	68.55	31.86	67.77	58.14	53.97	<b>46.23</b>	44.81	28.36	32.69
GCMA <i>w/o GAT</i>	74.17	54.25	67.26	60.18	38.90	56.76	67.75	33.37	67.72	57.01	52.02	39.25	47.36	33.73	43.61
GCMA <i>w/o GCN</i>	75.43	55.75	68.20	66.94	41.63	61.59	68.91	32.43	68.87	59.22	55.13	43.52	50.92	21.40	27.20
GCMA <i>w/o G</i>	74.54	54.45	68.77	68.78	42.37	62.94	68.76	31.50	68.52	59.96	54.95	43.42	47.76	33.88	43.81
GCMA	<b>76.12</b>	<b>57.21</b>	<b>71.43</b>	<b>71.95</b>	<b>45.98</b>	<b>65.21</b>	<b>72.04</b>	<b>33.45</b>	<b>71.04</b>	<b>61.32</b>	<b>55.43</b>	<b>45.80</b>	<b>51.85</b>	<b>36.01</b>	<b>44.23</b>

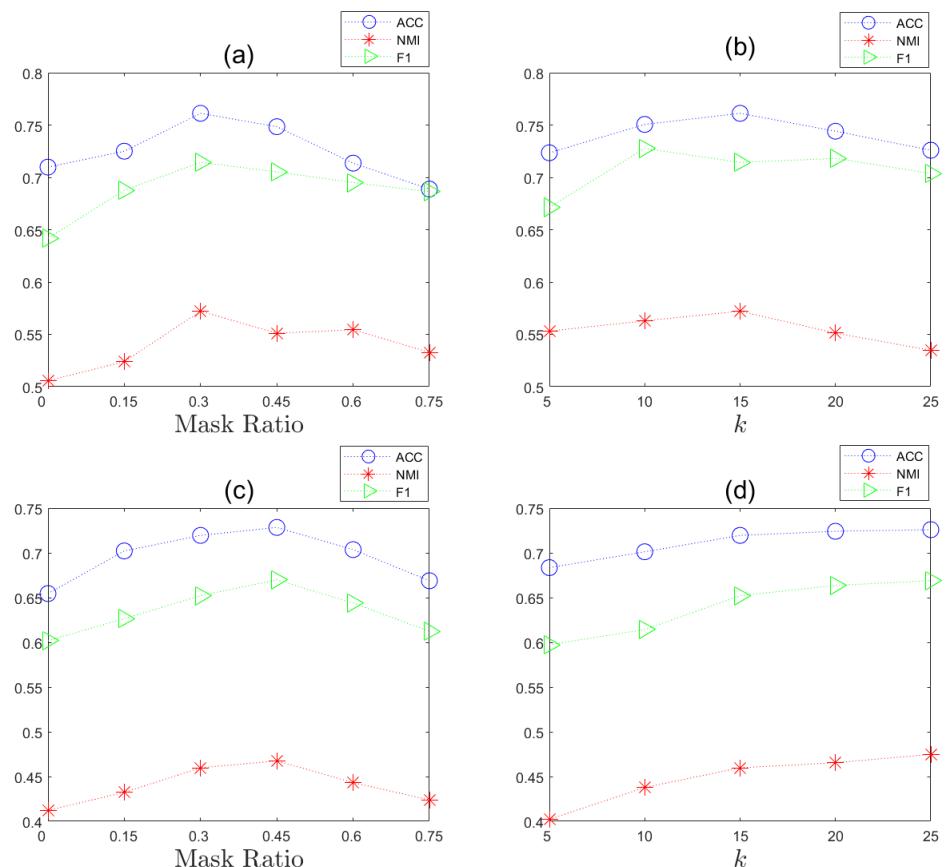
#### 4.6. Parameter Analysis

Our model has four hyperparameters: filter order  $m$ ,  $\beta$ , which controls the directional information, the hop of global neighbors  $k$  and the mask ratio. We test on the Cora and CiteSeer datasets. For  $m$  and  $\beta$ , we set  $m = [1, 2, 3, 4, 5]$  and  $\beta = [0.1, 0.3, 0.5, 0.7, 1.0]$ . The results are shown in Figure 5. It can be seen that our method is insensitive to them; thus, GCMA can work well for a large scale of hyperparameters on both datasets. For the mask ratio, we set it to  $[5\%, 10\%, 15\%, 20\%, 25\%]$ . The results are shown in Figure 6. Both Cora and CiteSeer show a similar trend. When the mask ratio is 0, i.e., without masking, GCMA

achieves poor results. It achieves the best performance when the masking ratio is around 30–45%, which further verifies that masking can help improve learning ability. When the ratio increases, the performance decreases as a result of insufficient information. We set  $k = [5, 10, 15, 20, 25]$ . The results are also shown in Figure 6. When  $k = 5$ , it achieves the worst results, while the other values of  $k$  exhibit comparable performance. Thus, global information has clear advantages for enhancing performance.



**Figure 5.** Results of parameter analysis for  $m$  and  $\beta$ : (a–c) are on Cora, while (d–f) are on CiteSeer.



**Figure 6.** Results of parameter analysis on mask ratio and  $k$ : (a,b) are on Cora, while (c,d) are on CiteSeer.

## 5. Future Directions

- Exploring different masking strategies: In this work, we used the traditional augmentation strategy of masking. Future work could explore different masking strategies to see if they can further improve the model's learning ability. This includes investigating different types of masks, such as one-sided and path masks, and examining their impact on the model's performance. The new masking strategies could be verified through experiments with the ACC metric.
- Adapting the masking strategy to different graph structures: While the GCMA method demonstrates remarkable performance on attributed graphs, which are characterized by their rich node features and interconnections, there remains a compelling need to apply it to diverse graph structures, such as hypergraphs and bipartite graphs.
- Application to other tasks: While our focus has been on graph clustering, the mask techniques we have developed could potentially be applied to other tasks. Future work could explore these possibilities.
- Integration of graph attention mechanisms: Incorporating graph attention mechanisms into masked autoencoders could improve their ability to capture important global structures in graphs. Graph attention mechanisms enable nodes to dynamically attend to their neighbors based on learned attention weights, allowing the model to focus on relevant global information while filtering out noise and irrelevant features. Extra constraints can be considered for attention weights.

## 6. Conclusions

This study introduces a novel method called Preserving Global Information for Graph Clustering with Masked Autoencoders (GCMA), which was meticulously crafted to tackle the inherent complexities of deep graph clustering tasks. It first introduces a low-high-pass filter to capture meaningful low- and high-frequency information. By integrating masked autoencoders into the domain of graph clustering, we leverage their formidable capability for representation learning, thereby enhancing the effectiveness of clustering efforts. Notably, GCMA pioneers an innovative graph diffusion technique, which is synergistically paired with a parameterized Laplacian matrix for precise global direction control. This innovative fusion significantly enhances the model's ability to capture the crucial global insights necessary for achieving nuanced clustering outcomes. Through exhaustive experimental evaluations conducted across a spectrum of benchmark datasets, GCMA emerges as a beacon of superiority, outshining established methodologies with resounding clarity. These empirical validations resolutely underscore GCMA's prowess in reshaping the landscape of graph clustering applications, reaffirming its status as a transformative force poised to push the boundaries of research in this domain. Overall, masked autoencoders play a crucial role in learning both structure and feature information for effective graph clustering. By selectively retaining relevant global information while filtering out noise, masked autoencoders generate more accurate and comprehensive representations of graph data without labels, resulting in enhanced clustering performance. Further, preserving global information is crucial for effective graph clustering, as it enables researchers to uncover hidden structures, gain insights into complex interdependencies, and produce better clustering solutions.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the author on request.

**Conflicts of Interest:** The author declares no conflicts of interest.

## References

1. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
2. Duan, Z.; Wang, C.; Zhong, W. SSGCL: Simple Social Recommendation with Graph Contrastive Learning. *Mathematics* **2024**, *12*, 1107. [[CrossRef](#)]
3. Yang, X.; Liu, Y.; Zhou, S.; Wang, S.; Tu, W.; Zheng, Q.; Liu, X.; Fang, L.; Zhu, E. Cluster-guided contrastive graph clustering network. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023.
4. Gan, J.; Liang, Y.; Du, L. Local-Sample-Weighted Clustering Ensemble with High-Order Graph Diffusion. *Mathematics* **2023**, *11*, 1340. [[CrossRef](#)]
5. Liu, Y.; Yang, X.; Zhou, S.; Liu, X.; Wang, S.; Liang, K.; Tu, W.; Li, L. Simple contrastive graph clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**. [[CrossRef](#)] [[PubMed](#)]
6. Tian, F.; Gao, B.; Cui, Q.; Chen, E.; Liu, T.Y. Learning deep representations for graph clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
7. Hassani, K.; Khasahmadi, A.H. Contrastive multi-view representation learning on graphs. In Proceedings of the International Conference on Machine Learning, PMLR 2020, Virtual Event, 13–18 July 2020; pp. 4116–4126.
8. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Deep graph contrastive representation learning. *arXiv* **2020**, arXiv:2006.04131.
9. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
10. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009.
11. Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; Tang, J. Graphmae: Self-supervised masked graph autoencoders. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 594–604.
12. Xie, X.; Chen, W.; Kang, Z.; Peng, C. Contrastive graph clustering with adaptive filter. *Expert Syst. Appl.* **2023**, *219*, 119645. [[CrossRef](#)]
13. Chien, E.; Peng, J.; Li, P.; Milenkovic, O. Adaptive Universal Generalized PageRank Graph Neural Network. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
14. Li, G.; Müller, M.; Ghanem, B.; Koltun, V. Training graph neural networks with 1000 layers. In Proceedings of the International Conference on Machine Learning, PMLR 2021, Virtual Event, 18–24 July 2021; pp. 6437–6449.
15. Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; Cui, P. Structural deep clustering network. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 1400–1410.
16. Kipf, T.N.; Welling, M. Variational graph auto-encoders. In Proceedings of the NIPS Bayesian Deep Learning Workshop, Barcelona, Spain, 10 December 2016.
17. Wang, C.; Pan, S.; Long, G.; Zhu, X.; Jiang, J. Mgae: Marginalized graph autoencoder for graph clustering. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 889–898.
18. Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; Zhang, C. Attributed Graph Clustering: A Deep Attentional Embedding Approach. In Proceedings of the IJCAI-19, Macao, China, 10–16 August 2019.
19. Hui, B.; Zhu, P.; Hu, Q. Collaborative graph convolutional networks: Unsupervised learning meets semi-supervised learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 4215–4222.
20. Wang, C.; Pan, S.; Celina, P.Y.; Hu, R.; Long, G.; Zhang, C. Deep neighbor-aware embedding for node clustering in attributed graphs. *Pattern Recognit.* **2022**, *122*, 108230. [[CrossRef](#)]
21. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR 2020, Virtual Event, 13–18 July 2020; pp. 1597–1607.
22. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.A.; Guo, Z.D.; Azar, M.G.; et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv* **2020**, arXiv:2006.07733.
23. You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph contrastive learning with augmentations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5812–5823.
24. Zhu, J.; Rossi, R.A.; Rao, A.; Mai, T.; Lipka, N.; Ahmed, N.K.; Koutra, D. Graph Neural Networks with Heterophily. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 11168–11176.
25. Veličković, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep Graph Infomax. In Proceedings of the ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
26. Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J.T.; Peng, X. Contrastive Clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 8547–8555.
27. Xia, W.; Gao, Q.; Yang, M.; Gao, X. Self-supervised Contrastive Attributed Graph Clustering. *arXiv* **2021**, arXiv:2110.08264.
28. Liu, Y.; Tu, W.; Zhou, S.; Liu, X.; Song, L.; Yang, X.; Zhu, E. Deep Graph Clustering via Dual Correlation Reduction. In Proceedings of the AAAI, Virtual Event, 22 February–1 March 2022.

29. Tan, Q.; Liu, N.; Huang, X.; Choi, S.H.; Li, L.; Chen, R.; Hu, X. S2GAE: Self-supervised graph autoencoders are generalizable learners with graph masking. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 787–795.
30. Li, J.; Wu, R.; Sun, W.; Chen, L.; Tian, S.; Zhu, L.; Meng, C.; Zheng, Z.; Wang, W. What’s Behind the Mask: Understanding Masked Graph Modeling for Graph Autoencoders. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 1268–1279.
31. Shi, Y.; Dong, Y.; Tan, Q.; Li, J.; Liu, N. Gigamae: Generalizable graph masked autoencoder via collaborative latent space reconstruction. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023; pp. 2259–2269.
32. Zhang, X.; Liu, H.; Li, Q.; Wu, X.M. Attributed Graph Clustering via Adaptive Graph Convolution. In Proceedings of the IJCAI-19, Macao, China, 10–16 August 2019.
33. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and deep locally connected networks on graphs. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014.
34. Henaff, M.; Bruna, J.; LeCun, Y. Deep convolutional networks on graph-structured data. *arXiv* **2015**, arXiv:1506.05163.
35. Kang, Z.; Liu, Z.; Pan, S.; Tian, L. Fine-grained Attributed Graph Clustering. In Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), SIAM 2022, Alexandria, VA, USA, 28–30 April 2022; pp. 370–378.
36. Chang, H.; Rong, Y.; Xu, T.; Huang, W.; Sojoudi, S.; Huang, J.; Zhu, W. Spectral graph attention network with fast eigen-approximation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, New York, NY, USA, 1–5 November 2021; pp. 2905–2909.
37. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Beyond Low-pass Filtering: Graph Convolutional Networks with Automatic Filtering. *arXiv* **2021**, arXiv:2107.04755.
38. Li, S.; Kim, D.; Wang, Q. Beyond low-pass filters: Adaptive feature propagation on graphs. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Bilbao, Spain, 13–17 September 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 450–465.
39. Zhang, X.; Xie, X.; Kang, Z. Graph Learning for Attributed Graph Clustering. *Mathematics* **2022**, *10*, 4834. [[CrossRef](#)]
40. Li, P.; Wang, Y.; Wang, H.; Leskovec, J. Distance Encoding: Design Provably More Powerful Neural Networks for Graph Representation Learning. *arXiv* **2020**, arXiv:2009.00142.
41. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
42. Wang, X.; Jin, D.; Cao, X.; Yang, L.; Zhang, W. Semantic community identification in large attribute networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
43. Li, Q.; Wu, X.M.; Liu, H.; Zhang, X.; Guan, Z. Label efficient semi-supervised learning via graph filtering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9582–9591.
44. Rozemberczki, B.; Allen, C.; Sarkar, R. Multi-scale attributed node embedding. *J. Complex Netw.* **2021**, *9*, cnab014. [[CrossRef](#)]
45. Pan, S.; Hu, R.; Long, G.; Jiang, J.; Yao, L.; Zhang, C. Adversarially regularized graph autoencoder for graph embedding. In Proceedings of the IJCAI 2018, Stockholm, Sweden, 13–19 July 2018.
46. Pan, S.; Hu, R.; Fung, S.f.; Long, G.; Jiang, J.; Zhang, C. Learning graph embedding with adversarial training methods. *IEEE Trans. Cybern.* **2019**, *50*, 2475–2487. [[CrossRef](#)] [[PubMed](#)]
47. Li, X.; Zhu, R.; Cheng, Y.; Shan, C.; Luo, S.; Li, D.; Qian, W. Finding global homophily in graph neural networks when meeting heterophily. In Proceedings of the International Conference on Machine Learning. PMLR 2022, Baltimore, MN, USA, 17–23 July 2022; pp. 13242–13256.
48. Guo, L.; Dai, Q. End-to-end variational graph clustering with local structural preservation. *Neural Comput. Appl.* **2021**, *34*, 3767–3782. [[CrossRef](#)]
49. Zhu, P.; Li, J.; Xiao, B.; Zhao, S.; Hu, Q. Collaborative Decision-Reinforced Self-Supervision for Attributed Graph Clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 10851–10863. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.