

Article

Global Self-Attention-Driven Graph Clustering Ensemble

Lingbin Zeng , Shixin Yao *, You Huang , Liquan Xiao , Yong Cheng and Yue Qian 

National University of Defense Technology, Changsha 410073, China; zenglingbin16@nudt.edu.cn (L.Z.); huangyou@nudt.edu.cn (Y.H.); xiaoliqun@nudt.edu.cn (L.X.); ycheng@nudt.edu.cn (Y.C.); yueqian@nudt.edu.cn (Y.Q.)

* Correspondence: yaoshixin22@nudt.edu.cn

Highlights

What are the main findings?

- The proposed global self-attention-driven graph clustering ensemble (GSAGCE) effectively fuses attribute and structural information through a novel global self-attention graph autoencoder that captures long-range vertex dependencies.
- Our double-weighted graph partitioning consensus function simultaneously considers both global and local diversity within base clusterings, enhancing the overall consensus clustering performance.

What is the implication of the main finding?

- GSAGCE addresses critical limitations in existing clustering ensemble methods for graph-structured data.
- The self-supervised strategy we designed provides more reliable guidance for producing high-quality base clusterings, which can be extended to other domains requiring effective processing of complex graph-structured data.

Abstract

A clustering ensemble, which leverages multiple base clusterings to obtain a reliable consensus result, is a critical challenging task for Earth observation in remote sensing applications. With the development of multi-source remote sensing data, exploring the underlying graph-structured patterns has become increasingly important. However, existing clustering ensemble methods mostly employ shallow clustering in the base clustering generation stage, which fails to utilize the structural information. Moreover, the high dimensionality inherent in data further increases the difficulty of clustering. To address these problems, we propose a novel method termed Global Self-Attention-driven Graph Clustering Ensemble (GSAGCE). Specifically, GSAGCE firstly adopts basic autoencoders and global self-attention graph autoencoders (GSAGAEs) to extract node attribute information and structural information, respectively. GSAGAEs not only enhance structural information in the embedding but also have the capability to capture long-range vertex dependencies. Next, we employ a fusion strategy to adaptively fuse this dual information by considering the importance of nodes through an attention mechanism. Furthermore, we design a self-supervised strategy to adjust the clustering distribution, which integrates the attribute and structural embeddings as more reliable guidance to produce base clusterings. In the ensemble strategy, we devise a double-weighted graph partitioning consensus function that simultaneously considers both global and local diversity within the base clusterings to enhance the consensus performance. Extensive experiments on benchmark datasets demonstrate the superiority of GSAGCE compared to other state-of-the-art methods.

Keywords: clustering ensemble; remote sensing; structural information; global self-attention



Academic Editors: Francesco Benedetto and Fabio Tosti

Received: 29 August 2025

Revised: 3 November 2025

Accepted: 5 November 2025

Published: 10 November 2025

Citation: Zeng, L.; Yao, S.; Huang, Y.; Xiao, L.; Cheng, Y.; Qian, Y. Global Self-Attention-Driven Graph Clustering Ensemble. *Remote Sens.* **2025**, *17*, 3680. <https://doi.org/10.3390/rs17223680>

Copyright: © 2025 by the authors. 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

Benefiting from the continuous advancement of unsupervised learning, clustering tasks for complex data have significantly alleviated the challenges posed by large-scale unlabeled information [1]. Among these, the clustering ensemble has emerged as a powerful paradigm in the clustering field, aiming to elegantly combine multiple weak base clusterings to achieve a more robust and superior result [2–4]. This approach has demonstrated remarkable success across diverse domains, including computer vision [5–7], bioinformatics [8–10], and anomaly detection [11,12]. Clustering ensemble can effectively handle and fuse this data heterogeneity through different base clusterings. However, despite its widespread adoption, two critical limitations persist in current clustering ensemble research, particularly in the context of graph-structured data.

First, existing clustering ensemble methods mainly focus on consensus function design and selective ensemble strategies, while neglecting the generation of high-quality base clusterings—a fundamental yet understudied aspect of the ensemble framework. The performance of clustering ensemble heavily depends on the diversity and quality of base clusterings, low-quality base partitions can severely degrade the final ensemble result. Unfortunately, most current approaches rely on shallow clustering algorithms (e.g., k-means, spectral clustering) for base clustering generation, which suffer from limited representation capacity and fail to capture the complex, high-dimensional structures inherent in the data [13]. As big data and remote sensing applications increasingly generate graph-structured data [14–17], these shallow methods struggle to adapt, highlighting a pressing need for more advanced deep representation learning techniques in the base clustering phase.

Second, unlike common tabular data, graph-structured data contains more abundant information, including node attribute information and topological structure information. Due to the wealth of information inherent in graph-structured data, many real-world tasks can be effectively tackled through the clustering ensemble, relying on graphs, such as social networks [18,19] and biological interaction networks [20,21]. It is worth noting that the spatial characteristics of remote sensing data are inherently consistent with the representation of graphs. Each pixel can be regarded as a graph node, and its spatial adjacency relationship constitutes the edge structure of the graph. This graph representation can effectively preserve the most crucial spatial context information. However, existing methods often inadequately exploit this structural information. Traditional shallow clustering algorithms primarily focus on node attributes while ignoring the graph topology, which encodes critical relational patterns among samples. Even recent advances in graph neural networks (GNN) [22,23]—which excel at joint feature and topology learning [24–26]—have limitations. State-of-the-art GNN-based clustering methods (e.g., SDCN [27], DFCN [28]) rely heavily on localized graph convolutions (GCN [29,30]) or graph autoencoders (GAE [31,32]), which capture only short-range node dependencies and fail to model global structural patterns. This design restricts their ability to uncover higher-order connectivity and long-range interactions, ultimately compromising clustering accuracy. Therefore, effectively addressing the clustering ensemble of graph-structured data is a highly challenging research topic with significant practical value.

To address the aforementioned problems, we propose a novel approach termed Global Self-Attention-driven Graph Clustering Ensemble (GSAGCE), which simultaneously fuses attribute information and structural information for consensus clustering. Specifically, GSAGCE first exploits a basic autoencoder to mine node attribute information and then constructs a global self-attention graph autoencoder (GSAGAE) to sufficiently extract structural information. Unlike the standard graph autoencoder, GSAGAE utilizes a global self-attention mechanism to enhance the capability of graph convolutional network and

captures long-range vertex dependencies on a global level. Next, we employ a learning-aware feature fusion graph network, which incorporates node importance through attention mechanisms to adaptively fuse attribute and structural information. Furthermore, we design a self-supervised strategy to tune the clustering distribution, which integrates the learned representations from the autoencoder and GSAGAE as more reliable guidance to train base clusterings. Finally, in the ensemble strategy, we devise a double-weighted graph-partitioning consensus function that simultaneously considers the global diversity of base clusterings and local diversity within the base clusterings to further enhance the consensus performance.

In summary, the main contributions of our work are as follows:

- We propose a novel global self-attention-driven graph clustering ensemble method. In the process of capturing structural information, a novel global self-attention graph autoencoder is constructed to introduce extra expressive power to the graph convolutional network. It addresses the limitations of graph convolutional network in capturing global structural information. Moreover, the self-supervised strategy we designed can guide the learning of clustering distribution to achieve clearer boundaries and higher accuracy, significantly enhancing the clustering performance after training.
- In the ensemble strategy, a novel double-weighted graph-partitioning consensus function is devised to incorporate a global weighting uncertainty measure into a local weighting framework. Through this approach, it not only reflects the underlying relationships between clusters but also considers the differences between base clusterings, thereby enhancing the consensus clustering performance.
- The comprehensive experiments on seven benchmark datasets have demonstrated that our method significantly outperforms comparative state-of-the-art algorithms.

The remainder of this paper is organized as follows: In Section 2, we provide a brief related work on the clustering ensemble and deep clustering algorithms. The methodology of our proposed algorithm is elaborated in Section 3. The experimental results and analysis will be discussed in Section 4. Section 5 presents the conclusion and future work of this paper.

2. Related Work

In this section, we will introduce methods and techniques related to the proposed GSAGCE, primarily focusing on two aspects: clustering ensemble and deep clustering.

Clustering ensemble is initially inspired by the idea of ensemble learning in classification [33], which involves using multiple weak learning models to construct a more robust and superior one. Based on the construction approach of the consensus function, clustering ensemble methods can be categorized into two types: median partitioning and object co-occurrence [34]. Methods of the median partitioning type attempt to find a partition in the clustering ensemble that is most similar to all other partitions. Therefore, they typically involve solving an optimization problem to achieve a consensus partition result. Methods of the object co-occurrence type, on the other hand, obtain a consensus result by voting among the objects. Specifically, they seek cluster labels associated with each object in the consensus partition.

In recent years, many classic clustering ensemble methods have been primarily focused on research around the co-occurrence type. Tao et al. [35] proposed a robust spectral ensemble clustering method to address the noise issue in the co-association matrix by learning a robust representation of the co-association matrix through low-rank constraint. Liu et al. [36] devised a spectral ensemble clustering method for large-scale data, aiming to alleviate the high time and space complexities of consensus clustering through more efficient utilization of the co-association matrix. Zhou et al. [37] developed a dense representation-based ensemble clustering algorithm by weakening the influence of outliers

and theoretically verified that the final solution is the global optimal solution. Taking into account the impact of the reliability of base clusterings on the ensemble, Huang et al. [38] proposed a clustering ensemble algorithm based on ensemble-driven cluster uncertainty estimation and local weighting strategy. This algorithm is one of the most classic clustering ensemble methods in recent years. Zhou et al. [39] optimized the clustering ensemble from the perspective of improving the quality of base clustering and proposed an adaptive consensus multiple k-means method. Of course, there are many other excellent types of clustering ensemble methods used to address various constraints [40,41]. However, most of these clustering ensemble methods obtain base clusterings using shallow clustering models, with research focusing on the design of consensus functions while overlooking the importance of the mechanism for base clustering generation.

Deep clustering combines unsupervised deep representation networks into the clustering process, enabling clustering through the learning of deep embedded representations within complex data [42]. At present, researchers have explored various deep clustering methods, which are typically categorized based on the types of unsupervised deep representation networks used, such as autoencoder-based deep clustering, variational autoencoder-based deep clustering [43–46], and graph neural network-based deep clustering. The former two mainly perform clustering based on attribute features, while the latter mainly performs clustering based on graph structure features. Here, two main types of deep clustering are introduced, including autoencoder-based (AE-based) deep clustering and graph neural network-based (GNN-based) deep clustering.

Deep embedded clustering (DEC) [47], as a representative of the AE-based type, utilizes autoencoders for representation learning and enhances cluster cohesion through a KL divergence loss function. Guo et al. [48] improved DEC by incorporating a decoder network to preserve local structures in the data. They also proposed a convolutional neural network-based version specifically tailored for image data [49]. Alqahtani et al. [50] proposed a deep convolutional autoencoder embedded clustering method that is highly effective in image processing by simultaneously learning feature representations and cluster assignments. While these AE-based methods are able to capture deep intrinsic features in complex data compared to traditional shallow clustering models, they often fail to fully leverage the structural characteristics present in graph-structured data. Bo et al. [27] proposed a structural deep clustering network to address this limitation, where they integrated structural information into deep clustering by transferring representations learned by autoencoders to graph convolutional networks. This method is the first to explicitly apply structural information to deep clustering and is also the first GNN-based deep clustering method. Tu et al. [28] proposed a deep fusion clustering network by dynamically combining autoencoders and graph neural networks to leverage structural information for enhancing clustering performance. Liu et al. [51] proposed a self-supervised deep graph clustering method that addresses the issue of representation collapse during graph node encoding. They reduced information correlation in a dual manner to enhance the discriminative capability of features. These GNN-based methods, compared to AE-based methods, have the capability to incorporate local structural information into deep clustering. However, the existing graph attention mechanism learns representations through local neighbor aggregation. Although it is computationally efficient, its attention range is limited, and it is difficult to capture long-range dependencies in the graph. This is particularly crucial in clustering-ensemble tasks, as the quality of the base clusters is highly dependent on the accurate modeling of global node similarities. Therefore, we designed a global self-attention mechanism to calculate the attention weights for all node pairs, which can directly establish associations between distant nodes and avoid the information attenuation caused by multiple layers of message passing.

3. Global Self-Attention-Driven Graph Clustering Ensemble

In this section, we present the proposed model for global self-attention-driven graph clustering ensemble (GSAGCE). The overall framework of GSAGCE is depicted in Figure 1. GSAGCE can be divided into two major modules, namely the global self-attention-driven graph clustering (GSAGC) module and the double-weighted clustering ensemble (DWCE) module.

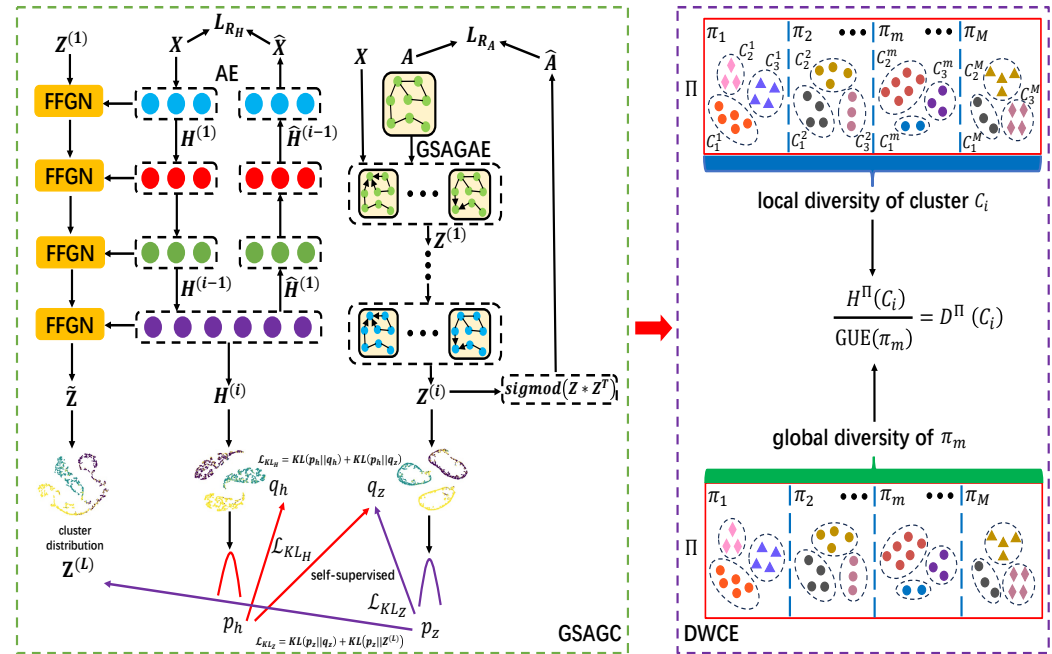


Figure 1. Illustration of the GSAGCE framework. GSAGCE is divided into two components, including the global self-attention-driven graph clustering (GSAGC) module and the double-weighted clustering ensemble (DWCE) module. Among them, GSAGC generates base clustering results by fully utilizing and integrating node attribute information and graph structure information, while DWCE enhances the ensemble performance by considering global and local diversity in the base clusterings.

The GSAGC can be further divided into an autoencoder (AE) module, a global self-attention graph autoencoder (GSAGAE) module, a feature fusion graph network (FFGN) module, and a self-supervised strategy module. Formally, given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with K categories of nodes, $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The graph $\mathcal{G}$ is characterized by its attribute feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and an original adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where n is the number of samples, and d is the feature dimension. The AE module captures the node attributed feature information, and the GSAGAE module extracts the topological graph structure information. The FFGN exploits a learning-aware fusion strategy, which adaptively fuses the attribute information and the structure information by considering the importance of nodes through an attention mechanism. As for the self-supervised strategy module, to generate more reliable guidance for clustering network training, we integrate the information from AE and GSAGAE to learn a more suitable clustering distribution for target distribution generation.

In the double-weighted clustering-ensemble module, we simultaneously consider the global diversity of base clusterings and the local cluster-wise diversity inside the same base clustering. Then, we design a novel double-weighted graph-partitioning consensus strategy that incorporates a global weighting uncertainty measure into a local weighting framework. After obtaining the double-weighted cluster uncertainty, the hybrid ensemble-driven cluster estimation (HECE) is proposed to further measure the reliability of the clusters. In the following subsections, we will detail the main components of our approach. Table 1 presents the summary of the main notations used in this paper.

Table 1. The summary of the main notations and descriptions.

Notations	Descriptions
$\mathbf{X} \in \mathbb{R}^{n \times d}$	The attribute feature data matrix
$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$	The graph structure
K	The number of clusters
$\mathcal{V}, \mathcal{E}$	The node set and the edge set
$\mathbf{A} \in \mathbb{R}^{n \times n}$	The original adjacency matrix
$\mathbf{H}^{(i)} \in \mathbb{R}^{n \times d_i}$	The i -th layer encoder part of autoencoder
d_i	The dimension of the i -th layer network
$\hat{\mathbf{H}}^{(i)}$	The i -th layer decoder part of autoencoder
$\mathbf{O}^{(l)}$	The global self-attention representation
$\mathbf{Z}^{(l)}$	The representation of GSAGAE
$\hat{\mathbf{A}}$	The reconstruction of graph adjacency structure
$\mathbf{M}^{(i)}$	The attention coefficient matrix of FFGN
$\tilde{\mathbf{Z}}$	The fused representation of AE and GSAGAE
$\mathbf{Z}^{(L)}$	The clustering probability distribution
$q_{h_{ij}}$	The similarity between $\mathbf{h}_i$ and cluster center μ_j
$p_{h_{ij}}$	The auxiliary distribution of $\mathbf{H}$
$q_{z_{ij}}$	The similarity between $\mathbf{z}_i$ and cluster center v_j
$p_{z_{ij}}$	The auxiliary distribution of $\mathbf{Z}$
$\Pi = \{\pi^i\}_{i=1}^M$	The clustering ensemble set
M	The number of base clusterings
π^m	The m -th base clustering
n^m	The number of clusters for π^m
$\text{Sim}(\pi^m, \pi^n)$	The similarity between π^m and π^n
$\text{GUE}(\pi^m)$	The global uncertainty estimation of π^m
$H^m(C_i)$	The uncertainty of cluster C_i
$\text{HECE}(C_i)$	The hybrid ensemble-driven cluster estimation of C_i

3.1. Global Self-Attention-Driven Graph Clustering (GSAGC)

3.1.1. Attributed Feature Representation via Autoencoder

Learning an effective data representation is crucial for clustering. There are many unsupervised representation learning methods, such as variational autoencoder [52], convolutional autoencoder, and adversarial autoencoder [53]. They are the variations in the basic autoencoder. In GSAGCE, to maintain generality, we utilize a basic autoencoder (AE) to capture the low-dimensional attributed representations of raw data. We assume that there are t layers in AE, and t indicates the number of layers. Specifically, the i -th layer representation $\mathbf{H}^{(i)} \in \mathbb{R}^{n \times d_i}$ in the encoder part can be obtained as follows:

$$\mathbf{H}^{(i)} = \phi\left(\mathbf{W}_{en}^{(i)} \mathbf{H}^{(i-1)} + \mathbf{b}_{en}^{(i)}\right), i = \{1, 2, \dots, t\}, \quad (1)$$

where $\mathbf{W}_{en}^{(i)}$ and $\mathbf{b}_{en}^{(i)}$ denote the weight and bias of the i -th encoder layer. ϕ is a LeakyReLU activation function (negative input slope is set as 0.2). d_i is the dimension of the i -th layer network. Moreover, $\mathbf{H}^{(0)}$ is the raw attribute feature matrix $\mathbf{X}$, and we typically set $t = 4$. The corresponding decoder part is expressed as follows:

$$\hat{\mathbf{H}}^{(i)} = \phi\left(\mathbf{W}_{de}^{(i)} \hat{\mathbf{H}}^{(i-1)} + \mathbf{b}_{de}^{(i)}\right), i = \{1, 2, \dots, t\}, \quad (2)$$

Here, $\hat{\mathbf{H}}^{(i)}$ denotes the decoder outputs from the i -th layer, and $\hat{\mathbf{H}}^{(t)} = \hat{\mathbf{X}}$. $\hat{\mathbf{X}}$ is the reconstruction of raw data. $\mathbf{W}_{de}^{(i)}$ and $\mathbf{b}_{de}^{(i)}$ are the weight and bias of the i -th decoder layer. Then,

we can obtain the attribute information by minimizing the reconstruction loss between $\mathbf{X}$ and $\hat{\mathbf{X}}$:

$$\mathcal{L}_{R_H} = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2, \quad (3)$$

where $\|\cdot\|_F$ indicates the F-norm. By minimizing the reconstruction loss $\mathcal{L}_{R_H}$, the model is able to learn to accurately reconstruct the representations of node attributes, thus enabling us to obtain the corresponding attribute information in the graph structure data.

3.1.2. Structural Information via Global Self-Attention Graph Autoencoder

The AE module is able to learn the attributed feature representation from the node feature matrix while failing to exploit the underlying structure information among graph data. To sufficiently extract the graph structural information, we propose a global self-attention graph autoencoder (GSAGAE). GSAGAE firstly transforms the l -th encoder hidden layer representation $\mathbf{Z}^{(l)}$ ($l \leq t$) into two feature spaces (f, g) to calculate the self-attention with the following equation:

$$\begin{aligned} \beta_{i,j} &= \text{softmax}_{j \in \{1,2,\dots,n\}} [s_{i,j}], \\ \text{where } s_{i,j} &= \left(\mathbf{Z}_i^{(l)} \mathbf{W}_f \right) \left(\mathbf{Z}_j^{(l)} \mathbf{W}_g \right)^T, \end{aligned} \quad (4)$$

$\beta_{i,j}$ indicates the attention importance of node j on node i . $\mathbf{W}_f$ and $\mathbf{W}_g$ are two learned weight matrices, where the first role is to reduce the computational load by dimensionality reduction, and the second is to provide additional flexibility for trainable variables. After obtaining the attention importance map, the global attention feature can be expressed with the following equation:

$$\mathbf{o}_i^{(l)} = \left(\sum_{j=1}^n \beta_{i,j} \mathbf{Z}_j^{(l)} \mathbf{W}_h \right) \mathbf{W}_r, \quad (5)$$

where $\mathbf{W}_h$ is for dimensionality reduction and $\mathbf{W}_r$ projects the features back to the original size.

The global self-attention representation captures the information on a global level. However, for graph networks, local geometry is also crucial. Thus, we add the input feature map back to the global attention feature:

$$\mathbf{Z}^{(l+1)} = \sigma \left(\left(\mathbf{D}^{-\frac{1}{2}} (\mathbf{A} + \mathbf{I}) \mathbf{D}^{-\frac{1}{2}} \mathbf{Z}^{(l)} + \gamma \mathbf{O}^{(l)} \right) \mathbf{W}^{(l)} \right), \quad (6)$$

Here, $\mathbf{D} = \text{diag}(D_1, D_2, \dots, D_n)$ is the degree matrix, and $D_i = \sum_{v_j \in V} (a_{ij} + \mathbf{I}_{ij})$. $\mathbf{I}$ is the identity matrix. $\mathbf{D}^{-\frac{1}{2}} (\mathbf{A} + \mathbf{I}) \mathbf{D}^{-\frac{1}{2}}$ means the normalization for the original adjacency matrix $\mathbf{A}$. The reason for this normalization is that directly using the original adjacency matrix causes nodes to prefer high-degree nodes regardless of the information carried in the node. $\mathbf{O}^{(l)}$ is the output of the global self-attention, and $\mathbf{W}^{(l)}$ is the feature transformation weight of l -layer. In Equation (5), $\mathbf{o}_i^{(l)}$ represents the i -th row of $\mathbf{O}^{(l)}$. σ is the LeakyReLU activation function (negative input slope is set as 0.2). $\gamma > 0$ is a learnable scalar with initial value 0. Introducing a learnable γ allows the network to first explore the local neighborhood and then gradually assign more weights to the global level. During model training, when the tensor parameter 'requires_grad' is set to True, the value of γ will be automatically updated as the gradient descends.

The core limitation of existing attention graph neural networks (such as GAT) is the "limited attention range": they only calculate attention weights based on the local neighbors of nodes (such as K-nearest neighbors, first-order neighborhoods). Although it can capture the local structural correlations, it cannot directly model the long-range dependencies of

non-neighbor nodes. The innovation of GSAGAE lies in designing a “global attention feature + local graph convolution” dynamic fusion module instead of simply using global attention. Through γ , it achieves a dynamic balance of local and global information without manually setting the neighbor range and can simultaneously cover local topology and global correlation, which is a mechanism design that existing attention GNNs have not involved.

Then, in decoder, we use the inner product of $\mathbf{Z}$ and $\mathbf{Z}^T$ to reconstruct the graph topological structure $\hat{\mathbf{A}}$:

$$\hat{\mathbf{A}} = \text{sigmoid}(\mathbf{Z}\mathbf{Z}^T), \quad (7)$$

The reconstruction loss of GSAGAE is given by

$$\begin{aligned} \mathcal{L}_{R_A} &= \text{CrossEntropyLoss}(\mathbf{A}, \hat{\mathbf{A}}) \\ &= -\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n [\mathbf{A}_{ij} \log(\hat{\mathbf{A}}_{ij}) + (1 - \mathbf{A}_{ij}) \log(1 - \hat{\mathbf{A}}_{ij})], \end{aligned} \quad (8)$$

By integrating Equations (3) and (8), the reconstruction loss of the GSAGC module is defined as follows:

$$\mathcal{L}_R = \mathcal{L}_{R_H} + \mathcal{L}_{R_A}, \quad (9)$$

3.1.3. Feature Fusion Graph Network

To further enhance the capability of representation learning, we exploit a learning-aware feature fusion graph network to adaptively fuse the attributed and structural representation. Here, the attention mechanism is employed for feature fusion, aiming to address the issue of feature heterogeneity. The attribute representation $\mathbf{H}^{(i)}$ and the structural representation $\mathbf{Z}^{(i)}$ come from different information sources and are thus heterogeneous representations. Therefore, in the traditional fusion strategy, direct simple average accumulation will lead to the corresponding loss in heterogeneous expression fusion. Specifically, we first concatenate $\mathbf{H}^{(i)}$ and $\mathbf{Z}^{(i)}$ as $[\mathbf{H}^{(i)} \parallel \mathbf{Z}^{(i)}] \in \mathbb{R}^{n \times 2d_i}$. Then, a weight matrix $\mathbf{W}_a^{(i)} \in \mathbb{R}^{2d_i \times 2}$ is introduced to capture the relationship of the concatenated features. Moreover, we apply a LeakyReLU activation function (negative input slope is set as 0.2) on the product between $[\mathbf{H}^{(i)} \parallel \mathbf{Z}^{(i)}]$ and $\mathbf{W}_a^{(i)}$. Finally, through softmax function and ℓ_2 normalization, we formulate the corresponding expression as follows:

$$\mathbf{M}^{(i)} = \ell_2(\text{softmax}((\text{LeakyReLU}([\mathbf{H}^{(i)} \parallel \mathbf{Z}^{(i)}] \mathbf{W}_a^{(i)})))), \quad (10)$$

where $\mathbf{M}^{(i)}$ indicates the attention coefficient matrix. Thus, we can adaptively fuse the GSAGAE feature $\mathbf{Z}^{(i)}$ and the AE feature $\mathbf{H}^{(i)}$ as follows:

$$\tilde{\mathbf{Z}}^{(i)} = \mathbf{M}_1^{(i)} \odot \mathbf{H}^{(i)} + \mathbf{M}_2^{(i)} \odot \mathbf{Z}^{(i)}, \quad (11)$$

$\mathbf{M}_1^{(i)}, \mathbf{M}_2^{(i)}$ represent the expanded matrix of $\mathbf{M}^{(i)}$, and $\odot$ denotes the Hadamard product. Afterwards, we obtain the final fused representation $\tilde{\mathbf{Z}}$ and directly use the softmax function to obtain the clustering probability distribution:

$$\begin{aligned} \mathbf{Z}^{(L)} &= \text{softmax}(\tilde{\mathbf{Z}}), \\ \text{s.t.} \quad &\sum_{j=1}^K z_{ij} = 1, z_{ij} > 0, \end{aligned} \quad (12)$$

Thus, we can obtain the clustering label through $\mathbf{Z}^{(L)}$:

$$y_i = \arg \max_j z_{ij}, j = \{1, 2, \dots, K\}, \quad (13)$$

3.1.4. Self-Supervised Strategy

Due to the lack of reliable guidance for cluster probability distribution $\mathbf{Z}^{(L)}$ when training the previous networks, we integrate the information from AE and GSAGAE to obtain suitable clustering distribution for guidance. In particular, we calculate the soft assignment distribution of $\mathbf{H}$ and $\mathbf{Z}$ using Student's t-distribution [47]:

$$q_{h_{ij}} = \frac{\left(1 + \|\mathbf{h}_i - \mu_j\|^2\right)^{-1}}{\sum_{j'} \left(1 + \|\mathbf{h}_i - \mu_{j'}\|^2\right)^{-1}}, \quad (14)$$

$$q_{z_{ij}} = \frac{\left(1 + \|\mathbf{z}_i - v_j\|^2\right)^{-1}}{\sum_{j'} \left(1 + \|\mathbf{z}_i - v_{j'}\|^2\right)^{-1}},$$

where $q_{h_{ij}}$ indicates the similarity between $\mathbf{h}_i$ and cluster center μ_j , $\mathbf{h}_i$ is the i -th row of $\mathbf{H}$, and cluster center μ_j is initialized by k-means. The initialization of cluster centers is carried out after the pre-training process. The pre-trained representations are then subjected to the k-means algorithm to initialize the cluster centers μ_j . The reason for the existence of the Student's t-distribution lies in that its power-law form can better model the similarity between high-dimensional data points and the cluster centers, while also alleviating the interference of outliers on the soft assignment. The interpretation of $q_{z_{ij}}$ is similar to $q_{h_{ij}}$. Then, we construct the corresponding auxiliary distribution of $\mathbf{H}$ and $\mathbf{Z}$ as follows:

$$p_{h_{ij}} = \frac{q_{h_{ij}}^2 / \sum_i q_{h_{ij}}}{\sum_{j'} q_{h_{ij'}}^2 / \sum_i q_{h_{ij'}}}, \quad (15)$$

$$p_{z_{ij}} = \frac{q_{z_{ij}}^2 / \sum_i q_{z_{ij}}}{\sum_{j'} q_{z_{ij'}}^2 / \sum_i q_{z_{ij'}}},$$

Thus, we can utilize auxiliary distribution p_h to guide q_h and q_z , which help AE and GSAGAE to learn more reliable representation for clustering. Here, we minimize the KL divergence loss among them:

$$\begin{aligned} \mathcal{L}_{KL_H} &= KL(\mathbf{p}_h \parallel \mathbf{q}_h) + KL(\mathbf{p}_h \parallel \mathbf{q}_z), \\ &= \sum_i \sum_j p_{h_{ij}} \log \frac{p_{h_{ij}}}{q_{h_{ij}}} + \sum_i \sum_j p_{h_{ij}} \log \frac{p_{h_{ij}}}{q_{z_{ij}}}, \end{aligned} \quad (16)$$

In addition, to promote a highly consistent distribution alignment to train our model, we employ auxiliary distribution p_z to guide the update of q_z and cluster distribution $\mathbf{Z}^{(L)}$ by minimizing the KL divergence:

$$\begin{aligned} \mathcal{L}_{KL_Z} &= KL(\mathbf{p}_z \parallel \mathbf{q}_z) + KL(\mathbf{p}_z \parallel \mathbf{Z}^{(L)}), \\ &= \sum_i \sum_j p_{z_{ij}} \log \frac{p_{z_{ij}}}{q_{z_{ij}}} + \sum_i \sum_j p_{z_{ij}} \log \frac{p_{z_{ij}}}{\mathbf{Z}_{ij}^{(L)}}, \end{aligned} \quad (17)$$

In summary, combining the Equations (9), (16) and (17), the overall loss function of GSAGC can be written as follows:

$$\mathcal{L} = \mathcal{L}_R + \lambda_1 \mathcal{L}_{KL_H} + \lambda_2 \mathcal{L}_{KL_Z}, \quad (18)$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are the trade-off hyper-parameters. For clarity, we summarize the complete training process of GSAGC in Algorithm 1.

Algorithm 1 Training process of GSAGC

Input: $\mathbf{X}$: attribute feature matrix, $\mathbf{A}$: original adjacency matrix, K : number of clusters, λ_1, λ_2 : trade-off hyper-parameters (default 1), $Iter_{\max}$: maximum iterations (default 200).

Output: π : clustering result

- 1: Pre-trained basic AE via Equations (1)–(3);
 - 2: Pre-trained GSAGAE via Equations (4)–(8);
 - 3: Initialize the k-means cluster center μ and v ;
 - 4: **while** $iter \leq Iter_{\max}$ **do**
 - 5: Compute the attributed representations $\mathbf{H}^{(i)}$;
 - 6: Reconstruct raw attribute data $\hat{\mathbf{X}}$;
 - 7: Compute the loss function $\mathcal{L}_{R_H}$ via Equation (3);
 - 8: Compute the structural representation $\mathbf{Z}^{(i)}$ via Equation (6);
 - 9: Fuse $\mathbf{H}^{(i)}$ and $\mathbf{Z}^{(i)}$ via Equations (10) and (11) to obtain $\tilde{\mathbf{Z}}$;
 - 10: Generate representations of GSAGAE via Equation (6);
 - 11: Reconstruct the graph structure $\hat{\mathbf{A}}$;
 - 12: Compute the loss function $\mathcal{L}_{R_A}$ via Equation (8);
 - 13: Generate the cluster probability distribution $\mathbf{Z}^{(L)}$ via Equation (12);
 - 14: Compute the soft assignment distribution q_h and q_z via Equation (14);
 - 15: Compute the corresponding auxiliary distribution p_h and p_z via Equation (15);
 - 16: Compute the KL divergence loss $\mathcal{L}_{KL_H}$ and $\mathcal{L}_{KL_Z}$ via Equations (16) and (17);
 - 17: Compute the overall loss $\mathcal{L}$ via Equation (18);
 - 18: Backward propagate and update parameters in GSAGC;
 - 19: $iter = iter + 1$;
 - 20: **end while**
 - 21: Obtain clustering result π via Equation (13);
 - 22: **return** π
-

3.2. Double-Weighted Clustering Ensemble (DWCE)

Given the ensemble $\Pi = \{\pi^1, \dots, \pi^M\}$ consisting of M base clusterings, $\pi^m = \{C_1^m, C_2^m, \dots, C_{n^m}^m\}$ is the m -th base clustering. n^m is the number of clusters for π^m , and C_i^m is the i -th cluster.

3.2.1. Global Uncertainty Estimation

Considering the global diversity of base clusterings in ensemble Π , we propose a global uncertainty estimation to calculate the reliability of base clustering. Specifically, we first measure the similarity between two base clusterings π^m and π^n by the adjusted rand index (ARI) [54]:

$$\text{Sim}(\pi^m, \pi^n) = \text{ARI}(\pi^m, \pi^n), \quad (19)$$

Then, we can obtain the overall similarity for π^m :

$$\text{Sim}(\pi^m) = \frac{1}{M-1} \sum_{\pi^n \in \Pi, n \neq m} \text{Sim}(\pi^m, \pi^n), \quad (20)$$

After that, the global uncertainty estimation (GUE) in Π for π^m can be defined as follows:

$$\text{GUE}(\pi^m) = \frac{\text{Sim}(\pi^m)}{\max_{\pi^n \in \Pi} \text{Sim}(\pi^n)}, \quad (21)$$

where the value of GUE is between $[0, 1]$. The higher the GUE value of a base clustering, the higher the credibility of the clustering results.

3.2.2. Local Uncertainty Measurement

Inspired by the uncertainty measurement of different clusters [38], we consider the local cluster diversity inside the same base clustering. In particular, the uncertainty of cluster $C_i \in \pi^m$ ($\pi^m \in \Pi$) is calculated as follows:

$$H^m(C_i) = - \sum_{j=1}^{n^m} p(C_i, C_j^m) \log_2 p(C_i, C_j^m), \quad (22)$$

$$p(C_i, C_j^m) = \frac{|C_i \cap C_j^m|}{|C_i|},$$

where C_j^m is the j -th cluster in π^m , and $|C_i|$ represents the number of objects. Then, the entire local uncertainty of C_i in ensemble Π can be obtained via the following equation:

$$H^\Pi(C_i) = \sum_{m=1}^M H^m(C_i), \quad (23)$$

3.2.3. Hybrid Ensemble-Driven Cluster Estimation

Based on global uncertainty estimation in Equation (21) and local uncertainty measurement in Equation (23), we can compute the double-weighted cluster uncertainty of C_i in ensemble Π :

$$D^\Pi(C_i) = \frac{H^\Pi(C_i)}{\text{GUE}(\pi^m)}, \quad (24)$$

When the value of $\text{GUE}(\pi^m)$ is larger, the reliability of the base clustering that generates C_i is higher. Thus, the uncertainty of C_i is expected to decrease. In the way, it makes the cluster-level uncertainty more accurate and further estimates the uncertainty of the cluster from the base clustering level.

After computing the double-weighted cluster uncertainty, we construct a novel hybrid ensemble-driven cluster estimation (HECE) to measure the reliability of the clusters, which is defined as follows:

$$\text{HECE}(C_i) = e^{-\frac{D^\Pi(C_i)}{\theta \cdot M}}, \quad (25)$$

Here, $\theta > 0$ adjusts the impact of uncertainty on the HECE. θ is set to 0.4 in this paper. It is easy to see that a smaller uncertainty value leads to a greater HECE value.

Finally, based on HECE, we design a double-weighted graph partitioning consensus function. The weighted graph is expressed as $G = (V, L)$, in which $V = X \cup \mathcal{C}$ is the node set and L is the link set. The link weight between two nodes v_i and v_j is calculated as follows:

$$l_{ij} = \begin{cases} \text{HECE}(v_j), & \text{if } v_i \in X, v_j \in \mathcal{C}, \text{ and } v_i \in v_j, \\ \text{HECE}(v_i), & \text{if } v_j \in X, v_i \in \mathcal{C}, \text{ and } v_j \in v_i, \\ 0, & \text{otherwise.} \end{cases} \quad (26)$$

where $\mathcal{C} = \{C_1, C_2, \dots, C_{N_c}\}$ consists of all clusters in Π . $N_c = n^1 + \dots + n^M$ denotes the total number of clusters in Π . The final consensus clustering π^* is obtained by using the Tcut algorithm [39].

3.3. Method Discussions

This section concludes with discussions of the computational and space complexity of our approach. In addition, we introduce the training steps of the GSAGC part.

Computational complexity: The computational complexity of GSAGCE consists of two parts: global self-attention-driven graph clustering (GSAGC) and double-weighted clustering ensemble (DWCE). For a given data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, n is the number of samples, and d is the feature dimension. t indicates the number of network layers. d_i is the dimension of the i -th layer network. The graph structure $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is divided into K categories of nodes. $|\mathcal{E}|$ denotes the number of edges. For the GSAGC, the complexity comes from the basic autoencoder (AE), global self-attention graph autoencoder (GSAGAE), feature fusion graph network (FFGN), and self-supervised module. The complexity of AE is $\mathcal{O}(n \prod_{i=2}^t d_{i-1} d_i)$. For the GSAGAE, as the operation can be computed efficiently using sparse matrix computation, the complexity is only $\mathcal{O}(|\mathcal{E}| \prod_{i=2}^t d_{i-1} d_i + nd \prod_{i=2}^t d_{i-1} d_i + n^2)$ according to [55]. The complexity of FFGN is $\mathcal{O}(\prod_{i=2}^t d_{i-1} d_i + K)$, and the complexity of the self-supervised module is $\mathcal{O}(nK + n \log n)$. For the DWCE, the complexity is $\mathcal{O}(Mn^m + Kne + K(n^m)^2)$, where M is the number of base clusterings, n^m is the number of clusters in π^m , and e is the average number of links connecting to a node in the graph partitioning.

Space complexity: In the training process of GSAGC, the space complexity mainly composes of attribute feature matrix $\mathbf{X}$, adjacency matrix $\mathbf{A}$, encoder matrices $\mathbf{H}^{(i)}$, decoder matrices $\hat{\mathbf{H}}^{(i)}$, global attention representation matrix $\mathbf{O}^{(l)}$, GSAGAE representation matrix $\mathbf{Z}^{(l)}$, and fused representation matrix $\tilde{\mathbf{Z}}$. Thus, the whole space cost can be analyzed as $\mathcal{O}(nd + n^2 + 4 \sum_{i=1}^t nd_i + nd_t)$.

GSAGC training steps: In order to learn more effective feature representations and avoid overfitting problems, we adopted pre-training and self-supervision strategies to train our GSAGC part. Specifically, we pre-train the AE and GSAGAE, respectively, and then load the pre-trained weights into the overall framework for joint training in combination with the self-supervised strategy and clustering loss.

4. Experiments and Results

In this section, to verify the effectiveness of the proposed GSAGCE algorithm, we conduct comprehensive experiments on seven benchmark datasets commonly used to compare GSAGCE against ten clustering ensemble methods and six deep clustering algorithms. Moreover, we also conduct ablation experiments, hyper-parameter sensitivity studies and visualization results to demonstrate the effectiveness of all parts in our model.

Regarding the clustering performance evaluation metric, we choose four widely used metrics, including accuracy (ACC), normalized mutual information (NMI), adjusted rand index (ARI), and macro F1. The larger the values of these metrics, the better the clustering performance. The detailed computation of these evaluation metrics is shown in the literature [56].

4.1. Datasets

We evaluate the proposed GSAGCE on seven benchmark datasets, including both graph type data and non-graph type data. Cora, ACM, Pubmed, and IMDB are typically popularly used in graph clustering literature. USPS, HAR, and REUT datasets are classically used to evaluate the performance of graph clustering algorithms on non-graph data. We

summarize the brief information of datasets in Table 2. For a detailed description of datasets, please refer to the relevant literature [28,51]. For the non-graph datasets, we follow literature [27] and construct the adjacency matrix with the heat kernel method.

Table 2. Statistics of the datasets.

Data Name	Sample	Feature	Category	Type	Edges
Cora	2708	1433	7	graph	5278
ACM	3025	1870	3	graph	13,128
Pubmed	19,717	500	3	graph	44,326
IMDB	4780	1232	2	graph	49,005
USPS	9298	256	10	image	-
HAR	10,299	561	6	record	-
REUT	10,000	2000	4	text	-

4.2. Experimental Settings

For all comparison baseline algorithms, for each dataset, we run 10 times and compute the average and standard deviation result for the evaluation measure. For clustering ensemble baselines, their ensemble size M is 20, and the number of clusters K is set to the true number of categories on the dataset.

For our proposed GSAGCE, the layers of the AE and GSAGAE modules are set to 4. The corresponding network dimensions of AE are 500-500-2000-20. The network dimensions of GSAGAE are the same as the AE. Before the formal training, we first pre-trained the AE module and the GSAGAE module separately. The epoch is set to 50 in the pre-training phase. The optimizer is selected as Adam, and the learning rate is dynamically adjusted using the cosine annealing strategy provided by PyTorch1.13. For the hyper-parameters λ_1, λ_2 in Equation (18), we perform a parameter sensitivity experiment to select the best performing value for each data. In most cases, we can set λ_1 and λ_2 to 1 by default. The remaining parameters are automatically updated by setting the tensor parameters 'requires_grad' to true, thereby enabling automatic differentiation. For non-graph data, we utilize KNN to construct a 3-NN graph with the heat kernel method. To ensure fair comparison, all the comparison methods of graph neural networks are based on the same graph construction method.

4.3. Analysis of Comparison Experiment

To fully verify the superiority of the GSAGCE, we compare with ten clustering ensemble methods, including RSEC [35], SECWK [36], LWEA [38], LWGP [38], DREC [37], LRTCE [57], ECCMS [58], ACMK [39], SCCABG [40], and CEAM [41]. Moreover, we also conduct comparisons with six popular deep clustering algorithms, including AE-kmeans [59], DEC [47], IDEC [48], SDCN [27], DFCN [28], and DCRN [51]. Among them, the first three are deep clustering methods that only utilize attribute information. The remaining three are deep graph clustering methods that simultaneously employ both attribute and structural information. Here is a brief introduction to these comparative methods.

- **Clustering ensemble methods:** Base clusterings employ a shallow clustering model, which can only utilize node attribute information and cannot utilize structural information.
 - RSEC [35] addresses the noise issue in the co-association matrix by learning a robust representation through low-rank constraint.
 - SECWK [36] aims to alleviate the high time and space complexities of clustering ensemble through a more efficient utilization of the co-association matrix.
 - LWEA and LWGP [38] consider ensemble-driven cluster uncertainty estimation and local weighting strategy in clustering ensemble.

- DREC [37] uses a dense representation-based ensemble clustering algorithm by weakening the influence of outliers.
- LRTCE [57] refines the co-association matrix from a global perspective through a novel constrained low-rank tensor approximation model.
- ECCMS [58] utilizes a co-association matrix self-enhancement method to strengthen its quality.
- ACMK [39] exploits an adaptive consensus multiple k-means method by improving the quality of base clustering.
- SCCABG [40] gradually incorporates data from more reliable to less reliable sources in consensus learning using adaptive bipartite graph learning.
- CEAM [41] learns an updated representation using a manifold ranking model through adaptive multiplex.
- **Deep clustering methods:**
 - AE-kmeans [59] learns embedded vectors of raw data using a basic autoencoder and then performs the k-means algorithm on the embedding.
 - DEC [47] utilizes denoising autoencoders for representation learning and enhances cluster cohesion through a KL divergence loss function.
 - IDEC [48] enhances DEC by incorporating a decoder network to preserve local structures.
 - SDCN [27] integrates structural information into deep clustering for the first time by transferring representations learned by autoencoders to graph convolutional networks.
 - DFCN [28] enhances clustering performance by dynamically combining autoencoders and graph neural networks while leveraging both attribute and structural information.
 - DCRN [51] is a self-supervised deep graph clustering method that addresses the issue of representation collapse during graph node encoding.

The clustering performance results of the comparison experiment are shown in Tables 3 and 4. For ease of presentation, we have retained four significant figures and multiplied the values of the four evaluation metrics by 100%. Table 3 shows the comparative results of clustering ensemble methods, while Table 4 presents the comparative results of deep clustering algorithms. The best clustering performance obtained for each dataset is highlighted in bold. All the results in bold are statistically superior (based on a paired *t*-test at the 5% significance level) to the other non-bold methods.

From Table 3, we can observe that GSAGCE significantly outperforms the comparative clustering ensemble methods. The reason is that their base clustering generation algorithm adopts a shallow clustering model, which cannot leverage the implicit information in these high-dimensional sparse data, especially for graph type data, as they overlook the graph structure information. Graph structure information is conducive to capturing the relationships between data and discovering cluster structures. The quality of base clustering plays a crucial role in the ensemble phase, low-quality base clustering will inevitably greatly reduce the performance of the ensemble. Regarding the comparative clustering ensemble baselines, we found that LWEA and LWGP achieve performance close to GSAGCE on USPS data, but significantly underperformed GSAGCE on HAR and REUT datasets. SCCABG fails to obtain reasonable clustering results on HAR data, so the evaluation metrics are empty (-). The ECCMS, ACMK, and SCCABG methods performed poorly on the four graph datasets. The reason is that they are based on adaptive strategies to explore ensemble clustering, and are not well suited for sparse graph structure data. The CEAM algorithm sometimes produces SVD errors when applied to graph type

data. In terms of computational efficiency, RSEC, LRTCE, ECMMS, and ACMK exhibit exponential growth in runtime as the number of data samples and dimensions increase. SECWK is the fastest, followed by LEWA, LWGP, and GSAGCE, while the remaining methods have similar runtime durations.

Table 3. The experimental results of clustering ensemble comparative methods on seven datasets. The best results are highlighted in bold.

Method	Metrics	Dataset						
		Cora	ACM	Pubmed	IMDB	USPS	HAR	REUT
RSEC	ACC	33.62 ± 3.23	66.89 ± 1.88	50.65 ± 7.48	62.17 ± 5.88	57.17 ± 4.63	67.13 ± 4.98	60.63 ± 6.40
	NMI	12.63 ± 2.13	33.36 ± 1.17	15.86 ± 7.16	1.20 ± 0.46	60.51 ± 3.40	67.38 ± 5.01	40.44 ± 8.29
	ARI	8.88 ± 1.98	30.93 ± 1.11	13.20 ± 8.16	−3.87 ± 0.81	46.39 ± 5.36	57.11 ± 5.58	34.01 ± 13.09
	F1	31.43 ± 3.41	67.10 ± 2.03	49.16 ± 11.48	44.15 ± 0.88	52.81 ± 4.69	64.69 ± 7.06	53.57 ± 7.23
SECWK	ACC	34.95 ± 2.28	58.91 ± 7.37	51.86 ± 6.65	74.79 ± 1.78	53.52 ± 5.79	49.01 ± 5.57	67.80 ± 10.58
	NMI	14.95 ± 2.18	22.33 ± 8.59	21.11 ± 11.09	0.34 ± 0.29	54.53 ± 4.84	51.25 ± 8.77	43.14 ± 11.62
	ARI	9.25 ± 2.00	19.83 ± 8.26	17.98 ± 11.63	−0.68 ± 1.75	35.47 ± 8.94	32.44 ± 10.60	40.76 ± 16.86
	F1	29.76 ± 2.57	55.79 ± 10.01	43.28 ± 12.37	44.64 ± 1.78	51.22 ± 6.32	45.66 ± 6.19	54.57 ± 13.69
LWEA	ACC	35.67 ± 2.94	73.36 ± 2.65	52.44 ± 4.19	74.03 ± 0.18	74.05 ± 2.37	69.37 ± 6.29	67.24 ± 7.85
	NMI	16.34 ± 3.04	38.94 ± 2.69	15.49 ± 5.32	0.45 ± 0.05	75.67 ± 1.15	74.91 ± 1.88	48.54 ± 5.44
	ARI	8.62 ± 2.60	37.52 ± 3.06	14.44 ± 7.41	−2.08 ± 0.14	67.68 ± 2.59	63.92 ± 3.50	45.04 ± 10.72
	F1	26.09 ± 5.73	73.55 ± 2.59	49.17 ± 6.28	43.99 ± 0.08	71.53 ± 2.78	63.23 ± 8.51	56.33 ± 8.82
LWGP	ACC	32.95 ± 3.06	65.57 ± 1.94	46.08 ± 4.31	74.18 ± 0.41	73.44 ± 5.46	65.41 ± 3.39	55.18 ± 7.11
	NMI	15.13 ± 1.98	32.69 ± 0.68	6.92 ± 2.92	0.36 ± 0.14	75.84 ± 1.15	69.61 ± 1.81	39.73 ± 10.02
	ARI	7.60 ± 2.52	30.07 ± 1.06	4.51 ± 4.72	−1.76 ± 0.38	67.48 ± 4.36	59.27 ± 2.16	27.73 ± 10.58
	F1	25.69 ± 4.08	65.73 ± 2.02	38.03 ± 7.66	44.21 ± 0.59	70.80 ± 6.45	59.76 ± 4.31	41.14 ± 9.09
DREC	ACC	36.91 ± 2.16	69.64 ± 1.85	47.29 ± 4.11	74.02 ± 0.88	68.13 ± 2.85	71.93 ± 3.39	74.46 ± 3.59
	NMI	16.77 ± 2.42	35.43 ± 1.60	8.24 ± 3.93	0.50 ± 0.17	68.75 ± 1.23	68.89 ± 2.32	47.97 ± 5.56
	ARI	11.83 ± 2.10	33.31 ± 2.08	5.30 ± 5.22	−2.13 ± 0.58	57.28 ± 1.98	59.45 ± 2.85	50.65 ± 9.55
	F1	34.79 ± 2.37	69.93 ± 1.82	37.38 ± 6.81	43.88 ± 0.17	66.67 ± 3.66	70.79 ± 4.85	62.31 ± 4.56
LRTCE	ACC	29.03 ± 9.68	68.40 ± 7.31	59.54 ± 8.39	57.59 ± 4.28	65.56 ± 2.16	56.91 ± 4.39	68.17 ± 8.12
	NMI	9.72 ± 8.34	34.24 ± 6.61	31.19 ± 7.21	0.22 ± 0.30	62.06 ± 0.89	60.20 ± 0.71	48.49 ± 8.88
	ARI	7.05 ± 6.17	31.96 ± 6.45	28.05 ± 8.67	−1.04 ± 1.17	53.58 ± 1.71	44.22 ± 2.59	42.27 ± 11.66
	F1	27.39 ± 9.50	68.78 ± 7.33	58.21 ± 9.03	47.32 ± 1.43	63.42 ± 2.26	55.00 ± 4.70	58.62 ± 4.65
ECCMS	ACC	30.29 ± 0.03	35.04 ± 0.19	40.93 ± 2.10	73.85 ± 0.41	46.72 ± 11.49	39.50 ± 7.88	43.27 ± 6.64
	NMI	0.48 ± 0.05	0.26 ± 0.23	1.38 ± 1.66	0.48 ± 0.28	60.15 ± 12.24	57.47 ± 6.34	17.45 ± 7.20
	ARI	0.03 ± 0.03	0.02 ± 0.02	0.44 ± 0.80	−2.07 ± 0.67	40.02 ± 14.39	36.44 ± 7.62	3.04 ± 9.72
	F1	6.91 ± 0.08	17.61 ± 0.25	22.15 ± 3.73	44.16 ± 0.72	28.51 ± 11.62	24.92 ± 9.22	25.75 ± 7.87
ACMK	ACC	18.47 ± 0.95	36.67 ± 2.68	39.89 ± 1.62	51.14 ± 0.94	63.03 ± 3.28	47.32 ± 7.31	35.76 ± 6.77
	NMI	1.18 ± 0.29	0.88 ± 1.44	2.51 ± 0.84	0.08 ± 0.05	59.16 ± 3.32	44.44 ± 7.72	8.71 ± 11.16
	ARI	0.45 ± 0.16	0.87 ± 1.54	1.92 ± 0.84	0.02 ± 0.13	51.10 ± 3.22	30.08 ± 8.52	6.50 ± 7.70
	F1	17.59 ± 0.81	36.61 ± 2.63	39.94 ± 1.70	47.14 ± 0.98	60.49 ± 3.75	45.84 ± 7.43	32.03 ± 4.98
SCCABG	ACC	30.79 ± 0.75	35.09 ± 0.05	40.24 ± 0.45	73.10 ± 0.90	37.72 ± 20.66	-	39.16 ± 0.01
	NMI	1.61 ± 1.42	0.15 ± 0.04	0.94 ± 0.80	0.15 ± 0.14	35.84 ± 34.46	-	14.43 ± 1.46
	ARI	0.15 ± 0.15	0.31 ± 0.05	0.02 ± 0.10	−0.44 ± 0.75	25.95 ± 25.89	-	−2.06 ± 0.16
	F1	8.20 ± 3.41	17.39 ± 0.07	20.26 ± 1.07	43.53 ± 0.21	24.60 ± 20.86	-	21.21 ± 0.16
CEAM	ACC	28.37 ± 2.15	59.45 ± 7.21	41.06 ± 2.27	69.28 ± 2.02	47.65 ± 14.09	55.41 ± 5.59	48.03 ± 10.52
	NMI	10.56 ± 1.52	21.72 ± 7.41	0.51 ± 0.55	0.68 ± 0.03	50.67 ± 13.57	59.27 ± 4.31	27.68 ± 13.12
	ARI	5.96 ± 2.19	20.96 ± 7.11	0.29 ± 0.60	−3.62 ± 0.23	36.24 ± 15.67	44.01 ± 6.29	15.79 ± 15.21
	F1	23.02 ± 1.11	59.42 ± 7.63	22.48 ± 5.57	45.01 ± 0.47	38.23 ± 18.60	51.28 ± 6.48	35.95 ± 11.89
GSAGCE	ACC	69.08 ± 2.21	91.76 ± 1.03	63.57 ± 3.21	75.27 ± 2.32	74.38 ± 2.23	82.47 ± 1.13	76.63 ± 1.86
	NMI	51.22 ± 1.47	72.00 ± 0.86	23.30 ± 2.46	4.27 ± 1.02	75.89 ± 1.36	82.91 ± 1.85	50.26 ± 2.18
	ARI	45.46 ± 1.36	78.49 ± 0.70	23.28 ± 3.01	13.71 ± 1.54	67.85 ± 2.40	74.89 ± 2.91	53.89 ± 2.03
	F1	65.44 ± 2.04	91.77 ± 0.92	63.79 ± 2.89	58.56 ± 2.11	72.72 ± 1.87	81.78 ± 2.58	65.38 ± 2.87

From Table 4, we can observe that GSAGCE has achieved the best clustering performance compared to these deep clustering algorithms. The reason is that GSAGCE not only utilizes attribute information and structural information simultaneously but also incorporates global self-attention to enhance the capability of graph convolution network (GCN) to capture long-range vertex dependencies and feature convolutions. This capability provides additional expressive power to GCN. On the other hand, GSAGCE further enhances the

capability of representation by adaptively fusing dual information in a synergistic manner. For the comparative deep clustering baselines, we found that DCRN performed poorly on three non-graph data types and encountered a CUDA out of memory (oom) issue on the Pubmed data. Even when the DCRN algorithm was imported to a larger GPU memory platform, the problem persisted.

Table 4. The experimental results of deep clustering comparative methods on seven datasets. The best results are highlighted in bold.

Dataset	Metrics	AE-kmeans	DEC	IDEC	SDCN	DFCN	DCRN	GSAGCE
Cora	ACC	35.44 ± 2.12	45.53 ± 2.23	45.20 ± 1.74	46.76 ± 6.00	44.67 ± 4.85	54.90 ± 8.37	69.08 ± 2.21
	NMI	14.52 ± 2.02	24.26 ± 1.94	25.44 ± 1.81	27.17 ± 4.45	33.09 ± 7.39	45.55 ± 6.41	51.22 ± 1.47
	ARI	10.69 ± 1.72	19.21 ± 1.87	18.82 ± 2.08	21.67 ± 4.69	28.70 ± 5.24	33.43 ± 7.64	45.46 ± 1.36
	F1	33.43 ± 2.09	44.11 ± 2.32	45.00 ± 1.78	39.90 ± 5.69	27.43 ± 4.92	48.01 ± 9.00	65.44 ± 2.04
ACM	ACC	56.03 ± 6.59	59.63 ± 6.76	73.57 ± 7.46	87.56 ± 1.44	90.77 ± 0.25	90.93 ± 0.42	91.76 ± 1.03
	NMI	19.18 ± 3.12	22.84 ± 4.78	36.64 ± 8.03	62.24 ± 2.77	69.32 ± 0.63	69.66 ± 0.95	72.00 ± 0.86
	ARI	18.61 ± 3.79	21.50 ± 5.36	39.94 ± 10.15	67.22 ± 3.19	74.77 ± 0.62	75.13 ± 1.07	78.49 ± 0.70
	F1	55.97 ± 6.71	59.94 ± 6.89	73.38 ± 7.72	87.46 ± 1.51	90.71 ± 0.26	90.93 ± 0.39	91.77 ± 0.92
Pubmed	ACC	53.42 ± 6.32	57.48 ± 4.73	57.11 ± 3.39	57.70 ± 4.77	50.67 ± 0.36	oom	63.57 ± 3.21
	NMI	17.17 ± 4.50	20.46 ± 3.38	19.73 ± 2.67	17.72 ± 4.59	6.80 ± 0.20	oom	23.30 ± 2.46
	ARI	14.26 ± 4.94	17.42 ± 4.25	16.26 ± 3.13	15.15 ± 5.03	6.75 ± 0.36	oom	23.28 ± 3.01
	F1	53.62 ± 6.54	57.94 ± 5.17	57.38 ± 3.62	58.06 ± 5.50	41.23 ± 0.35	oom	63.79 ± 2.89
IMDB	ACC	52.67 ± 1.55	52.73 ± 1.37	52.28 ± 1.98	53.35 ± 2.47	67.94 ± 9.65	71.34 ± 0.65	75.27 ± 2.32
	NMI	0.23 ± 0.27	0.88 ± 0.87	0.98 ± 1.10	1.94 ± 0.99	2.19 ± 1.52	0.43 ± 0.29	4.27 ± 1.02
	ARI	0.10 ± 0.37	−0.33 ± 0.33	0.03 ± 0.72	−0.02 ± 1.43	1.09 ± 2.29	−2.58 ± 1.15	13.71 ± 1.54
	F1	47.76 ± 1.93	47.86 ± 2.72	48.10 ± 3.34	50.27 ± 3.48	48.98 ± 5.34	45.44 ± 1.01	58.56 ± 2.11
USPS	ACC	65.90 ± 3.17	66.97 ± 2.80	71.25 ± 3.30	72.74 ± 5.08	73.51 ± 0.30	23.43 ± 3.72	74.38 ± 2.23
	NMI	63.65 ± 1.98	65.07 ± 1.26	73.61 ± 1.70	75.78 ± 2.53	75.49 ± 0.18	16.32 ± 7.63	75.89 ± 1.36
	ARI	55.39 ± 2.47	56.87 ± 1.74	64.17 ± 2.57	67.54 ± 4.10	67.45 ± 0.30	3.21 ± 5.62	67.85 ± 2.40
	F1	63.71 ± 4.22	64.99 ± 3.60	69.26 ± 4.87	70.43 ± 6.68	72.42 ± 0.35	15.14 ± 0.45	72.72 ± 1.87
HAR	ACC	62.94 ± 6.19	62.63 ± 3.23	70.91 ± 4.79	62.66 ± 5.52	77.26 ± 6.44	41.41 ± 2.51	82.47 ± 1.13
	NMI	57.41 ± 3.21	62.14 ± 2.99	77.00 ± 4.72	67.62 ± 2.42	81.09 ± 4.93	51.27 ± 0.64	82.91 ± 1.85
	ARI	49.71 ± 3.37	53.25 ± 3.27	66.75 ± 4.59	53.26 ± 3.81	71.29 ± 6.05	30.65 ± 1.52	74.89 ± 2.91
	F1	60.53 ± 7.33	60.82 ± 3.99	68.12 ± 5.27	54.17 ± 7.48	76.34 ± 7.91	34.17 ± 2.43	81.78 ± 2.58
REUT	ACC	56.69 ± 3.98	56.39 ± 2.23	59.72 ± 2.80	61.33 ± 6.09	63.82 ± 5.23	50.34 ± 4.61	76.63 ± 1.86
	NMI	27.77 ± 4.12	28.24 ± 3.45	34.54 ± 2.89	37.21 ± 10.39	41.23 ± 4.36	22.56 ± 8.51	50.26 ± 2.18
	ARI	26.40 ± 5.01	26.49 ± 3.39	30.63 ± 3.33	36.20 ± 11.11	42.52 ± 6.72	15.99 ± 6.40	53.89 ± 2.03
	F1	51.79 ± 4.27	51.51 ± 2.49	54.26 ± 3.73	54.07 ± 9.14	57.86 ± 6.18	33.31 ± 3.81	65.38 ± 2.87

4.4. Analysis of Ablation Experiment

To demonstrate the effectiveness of global self-attention graph autoencoder (GSAGAE), self-supervised strategy, and hybrid ensemble-driven cluster estimation (HECE) in Section 3, we conduct ablation experiments with four variants of GSAGCE in this section. “noGSA” is the variant of GSAGCE without global self-attention, which aims to verify the expressive power of GSAGAE. In noGSA, we only employ a standard graph convolution network without using global self-attention. “noKLh” and “noKLz”, respectively, denote the variant of GSAGCE that does not utilize the $\mathcal{L}_{KL_H}$ loss and $\mathcal{L}_{KL_Z}$ KL divergence loss. “noHECE” represents a variant of GSAGCE that excludes the double-weighted cluster uncertainty metrics and adopts only local uncertainty measurement. The ablation experiment results are shown in Table 5.

From Table 5, we can observe that the clustering performance of GSAGCE, considering all components together, is superior to these variant algorithms. Moreover, we found that the clustering performance of the noKLz algorithm is the worst across almost all datasets except for the ACM and REUT datasets. It indicates that in the various components of

GSAGCE, utilizing p_z to guide q_z and cluster distribution $\mathbf{Z}^{(L)}$ training is crucial. This is because the clustering performance is entirely dependent on the training level of the cluster distribution $\mathbf{Z}^{(L)}$. Only with reliable training guidance can a clustering-friendly cluster distribution be obtained. The variant with the second worst clustering performance is noKLh, and its performance gap compared to GSAGCE is quite significant. This indicates that utilizing p_h to guide q_h and q_z training is necessary, as it can help AE and GSAGAE learn more reliable representations. noGSA has achieved very poor clustering performance on REUT data and obtained a significant performance gap to GSAGCE on other datasets. This indicates that integrating global self-attention representations into GCN is highly effective. Furthermore, we have discovered an interesting phenomenon during the training of noGSA. In formal training, noGSA often reaches its highest clustering performance only in the initial iterations of training. In subsequent training, the clustering performance deteriorates until it stabilizes at a certain value, which is usually significantly lower than the highest value. This unreasonable phenomenon would not occur in GSAGCE. Regarding the variant noHECE, its performance on all datasets is also lower than GSAGCE. This indicates that the construction of a double-weighted cluster uncertainty metric is effective, performing better than considering only local uncertainty measurement.

Table 5. The ablation experiment results on seven datasets. The best results are highlighted in bold.

Dataset	Metrics	noGSA	noKLh	noKLz	noHECE	GSAGCE
Cora	ACC	66.84 ± 1.72	66.46 ± 1.46	47.41 ± 5.73	67.14 ± 1.89	69.08 ± 2.21
	NMI	47.45 ± 1.63	46.15 ± 1.44	28.72 ± 4.64	50.41 ± 2.01	51.22 ± 1.47
	ARI	42.69 ± 1.24	43.15 ± 2.16	20.35 ± 5.78	43.84 ± 1.56	45.46 ± 1.36
	F1	62.09 ± 0.53	61.20 ± 1.24	35.86 ± 5.20	64.81 ± 2.47	65.44 ± 2.04
ACM	ACC	91.21 ± 0.83	65.78 ± 1.76	75.30 ± 6.32	91.23 ± 0.86	91.76 ± 1.03
	NMI	69.84 ± 2.04	28.94 ± 5.04	47.22 ± 10.01	70.58 ± 0.63	72.00 ± 0.86
	ARI	75.90 ± 2.03	25.66 ± 3.91	48.34 ± 11.03	76.82 ± 1.05	78.49 ± 0.70
	F1	91.26 ± 0.86	66.61 ± 1.70	72.62 ± 7.70	91.20 ± 1.14	91.77 ± 0.92
Pubmed	ACC	61.58 ± 2.86	60.17 ± 3.05	52.05 ± 2.41	60.32 ± 2.68	63.57 ± 3.21
	NMI	20.00 ± 2.12	12.42 ± 2.24	16.42 ± 1.79	20.16 ± 2.43	23.30 ± 2.46
	ARI	18.69 ± 2.65	14.96 ± 3.14	17.75 ± 2.86	20.16 ± 2.79	23.28 ± 3.01
	F1	61.80 ± 2.25	55.13 ± 3.39	48.11 ± 2.91	60.58 ± 2.93	63.79 ± 2.89
IMDB	ACC	71.69 ± 0.23	72.35 ± 2.33	66.78 ± 6.87	74.30 ± 2.46	75.27 ± 2.32
	NMI	3.44 ± 0.23	3.72 ± 1.59	0.37 ± 2.72	3.94 ± 0.67	4.27 ± 1.02
	ARI	9.98 ± 6.44	9.55 ± 3.79	0.91 ± 4.80	11.89 ± 1.94	13.71 ± 1.54
	F1	55.31 ± 7.25	54.28 ± 4.20	43.31 ± 7.22	57.86 ± 2.33	58.56 ± 2.11
USPS	ACC	72.45 ± 3.51	68.83 ± 6.87	53.37 ± 4.33	73.83 ± 2.10	74.38 ± 2.23
	NMI	72.08 ± 6.02	65.78 ± 2.72	56.29 ± 4.94	75.32 ± 1.17	75.89 ± 1.36
	ARI	63.10 ± 5.57	57.10 ± 4.80	39.17 ± 6.65	67.51 ± 2.32	67.85 ± 2.40
	F1	65.79 ± 6.68	67.27 ± 7.22	43.36 ± 6.14	72.12 ± 2.06	72.72 ± 1.87
HAR	ACC	80.32 ± 1.27	78.66 ± 1.51	63.06 ± 4.85	81.36 ± 0.68	82.47 ± 1.13
	NMI	80.88 ± 1.53	71.06 ± 1.49	61.99 ± 3.42	81.76 ± 1.20	82.91 ± 1.85
	ARI	72.36 ± 3.06	63.38 ± 2.83	47.41 ± 4.58	72.54 ± 2.87	74.89 ± 2.91
	F1	80.42 ± 3.22	77.88 ± 2.53	55.16 ± 5.57	80.64 ± 3.05	81.78 ± 2.58
REUT	ACC	49.85 ± 1.47	39.36 ± 1.82	47.54 ± 2.25	75.31 ± 2.28	76.63 ± 1.86
	NMI	14.33 ± 1.29	3.10 ± 3.55	3.66 ± 2.56	48.88 ± 2.53	50.26 ± 2.18
	ARI	7.55 ± 1.57	1.23 ± 2.61	7.37 ± 3.00	52.79 ± 2.76	53.89 ± 2.03
	F1	32.65 ± 2.51	26.79 ± 4.96	32.02 ± 6.14	64.90 ± 3.05	65.38 ± 2.87

4.5. Analysis of Hyper-Parameters Sensitivity

In the previous ablation experiments, we found that the KL divergence loss functions, $\mathcal{L}_{KL_H}$ and $\mathcal{L}_{KL_Z}$, play crucial roles in the entire GSAGCE component. Therefore, selecting appropriate hyper-parameters, λ_1 and λ_2 , is particularly important. In this section, we conduct sensitivity experiments on two hyper-parameters. Specifically, we take λ_1 and λ_2

through $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ and carry out experiments to take the mean value of ARI on the Cora and ACM dataset. Selecting these two datasets as representatives for hyper-parameter sensitivity analysis, the results are shown in Figure 2.

From Figure 2, we can observe that the clustering performance on the Cora dataset is less affected by two hyper-parameters, and the performance improves as the values of λ_1 and λ_2 approach 1. On the ACM dataset, the performance is greatly influenced by two hyper-parameters; especially when λ_1 is less than λ_2 , the clustering performance is poor. Moreover, when λ_1 is less than 0.1, regardless of the value of λ_2 , the clustering performance is poor. This indicates that for the ACM dataset, when guiding the training for reliable representations simultaneously in a self-supervised strategy, the importance of guidance from p_h is greater. The main reason for the difference in hyper-parameter sensitivity between the Cora dataset and the ACM dataset lies in the complexity of the data. Cora is a classic citation dataset, and compared to ACM, it has a relatively smaller number of nodes and edges, with a simpler structure. Therefore, the model's performance on the Cora dataset is less sensitive to hyper-parameter choices, as the inherent characteristics of the data make it easier for the model to converge to a good solution. The ACM dataset has a larger number of nodes and edges, as well as a more complex structure. This implies that models may require more precise hyper-parameter tuning during training to achieve optimal performance. The optimal hyper-parameter acquisition approach for other datasets is the same as these two datasets. In most datasets, we can set λ_1 and λ_2 to 1 by default.

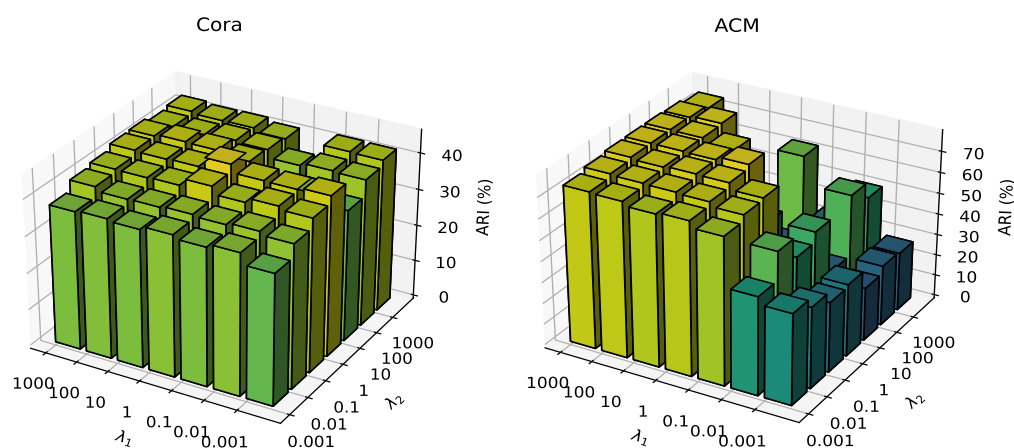


Figure 2. The ARI results of hyper-parameter sensitivity in Cora and ACM.

4.6. Analysis of Visualization

For a more intuitive display of the clustering effects in the global self-attention-driven graph clustering (GSAGC), we utilize t-SNE for the visualization of clustering on ACM and HAR data. Specifically, t-SNE visualizations are performed on the raw data matrix, the q_h distribution, the q_z distribution, and the cluster label distribution $\mathbf{Z}^{(L)}$. The visualization results are shown in Figure 3. From Figure 3, we can observe that compared to the raw data representation, the representations generated after GSAGC training are significantly better than the original data, exhibiting a more clustering-friendly nature. Therefore, this results in high-quality base clusterings generated by GSAGC, laying the foundation for the subsequent ensemble performance.

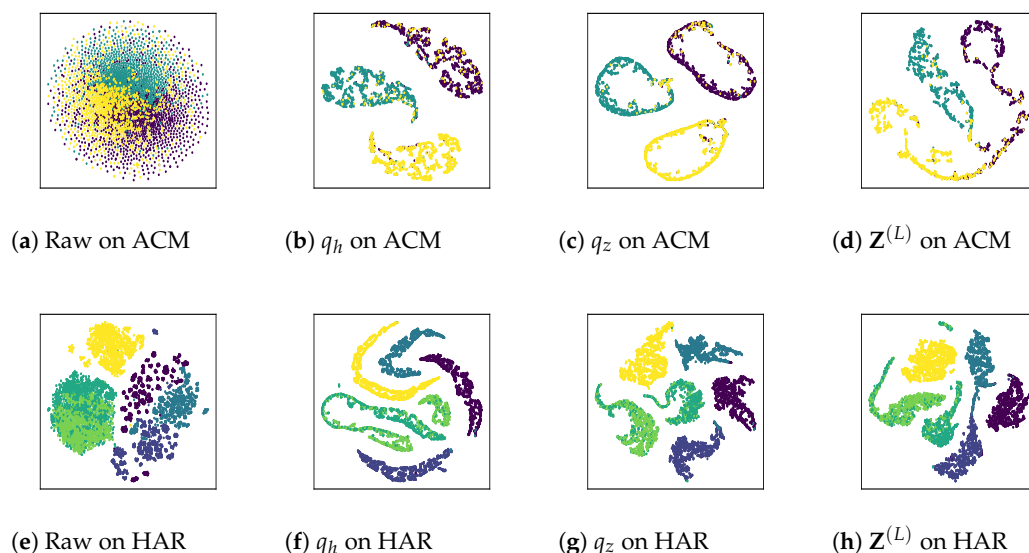


Figure 3. The visualization results on ACM and HAR.

5. Conclusions

In this paper, we present a novel clustering-ensemble algorithm called the global self-attention-driven graph clustering ensemble, which introduces graph clustering to the clustering ensemble. Considering the sparse and high-dimensional characteristic of graph-structured data, as well as the inability of existing clustering-ensemble methods to fully utilize the structural information in graph data, global self-attention-driven graph clustering (GSAGC) is constructed to simultaneously fuse attribute information and structural information. Moreover, we employ dual information to guide more reliable base clustering training. To enhance the consensus performance, we propose a global and local uncertainty estimation in the ensemble phase. The sufficient experiments on benchmark datasets validate the superiority of GSAGCE. The drawback and limitation of our method lies in the fact that it cannot be applied directly when dealing with incomplete or missing graph data. When applied to large-scale graphs, direct use may fail (due to out of CUDA memory). Dimensionality reduction must be performed first before this method can be used. Meanwhile, our method is not very suitable for heterogeneous graphs. It was mainly designed for isomorphic graphs. The gap between the benchmark data performance of the method and its adaptability in practical applications mainly lies in the fact that the benchmark data are standard and complete graph data. When there are missing parts or incorrect graph node structures in real data, it will significantly affect its performance. Combining incomplete clustering or multi-view learning with the proposed method is the main direction of our future work.

Author Contributions: Conceptualization, L.Z., S.Y., and Y.Q.; methodology, L.Z., S.Y., and Y.Q.; software, S.Y.; validation, L.Z., S.Y., L.X., and Y.C.; formal analysis, Y.H. and S.Y.; investigation, S.Y. and Y.Q.; resources, L.Z., Y.H., and Y.Q.; data curation, L.Z., S.Y., and Y.C.; writing—original draft preparation, L.Z., S.Y., and Y.Q.; writing—review and editing, L.Z., S.Y., Y.Q., L.X., and Y.C.; visualization, L.X., Y.H., and Y.C.; supervision, S.Y., Y.Q., L.X., and Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. Cora data can be found at <https://github.com/ki-ljl/PyG-GCN> (accessed on 20 January 2022). ACM, Pubmed, IMDB, USPS, HAR, and REUT data can be found at <https://github.com/yueliu1999/Awesome-Deep-Graph-Clustering/tree/main/dataset> (accessed on 10 January 2021).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Hu, X.; Li, T.; Zhou, T.; Peng, Y. Deep Spatial-Spectral Subspace Clustering for Hyperspectral Images Based on Contrastive Learning. *Remote Sens.* **2021**, *13*, 4418. [\[CrossRef\]](#)
- Wang, W.; Wang, W.; Liu, H. Correlation-Guided Ensemble Clustering for Hyperspectral Band Selection. *Remote Sens.* **2022**, *14*, 1156. [\[CrossRef\]](#)
- Zhang, M. Weighted clustering ensemble: A review. *Pattern Recognit.* **2022**, *124*, 108428. [\[CrossRef\]](#)
- Golalipour, K.; Akbari, E.; Hamidi, S.S.; Lee, M.; Enayatifar, R. From clustering to clustering ensemble selection: A review. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104388. [\[CrossRef\]](#)
- Zhai, H.; Zhang, H.; Li, P.; Zhang, L. Hyperspectral image clustering: Current achievements and future lines. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 35–67. [\[CrossRef\]](#)
- Zhao, Y.; Li, X. Deep spectral clustering with regularized linear embedding for hyperspectral image clustering. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5509311. [\[CrossRef\]](#)
- Guo, D.; Zhang, S.; Zhang, J.; Yang, B.; Lin, Y. Exploring Contextual Knowledge-Enhanced Speech Recognition in Air Traffic Control Communication: A Comparative Study. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 16085–16099. [\[CrossRef\]](#)
- Li, J.; Deng, W.; Dang, X.; Zhao, H. Cross-Domain Adaptation Fault Diagnosis with Maximum Classifier Discrepancy and Deep Feature Alignment Under Variable Working Conditions. *IEEE Trans. Reliab.* **2025**, *74*, 4106–4115. [\[CrossRef\]](#)
- Liu, Q.; Zhao, X.; Wang, G. A Clustering Ensemble Method for Cell Type Detection by Multiobjective Particle Optimization. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2023**, *20*, 1–14. [\[CrossRef\]](#)
- Nie, X.; Qin, D.; Zhou, X.; Duo, H.; Hao, Y.; Li, B.; Liang, G. Clustering ensemble in scRNA-seq data analysis: Methods, applications and challenges. *Comput. Biol. Med.* **2023**, *159*, 106939. [\[CrossRef\]](#)
- Yu, J.; Kang, J. Clustering ensemble-based novelty score for outlier detection. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106164. [\[CrossRef\]](#)
- Ray, B.; Ghosh, S.; Ahmed, S.; Sarkar, R.; Nasipuri, M. Outlier detection using an ensemble of clustering algorithms. *Multimed. Tools Appl.* **2022**, *81*, 2681–2709. [\[CrossRef\]](#)
- Yang, Y.; Lv, H.; Chen, N. A survey on ensemble learning under the era of deep learning. *Artif. Intell. Rev.* **2023**, *56*, 5545–5589. [\[CrossRef\]](#)
- Demirbaga, U.; Aujla, G.S.; Jindal, A.; Kalyon, O. *Big Data Analytics: Theory, Techniques, Platforms, and Applications*; Springer Nature: Berlin/Heidelberg, Germany, 2024.
- Chi, M.; Plaza, A.; Benediktsson, J.A.; Sun, Z.; Shen, J.; Zhu, Y. Big data for remote sensing: Challenges and opportunities. *Proc. IEEE* **2016**, *104*, 2207–2219. [\[CrossRef\]](#)
- Chen, H.; Sun, Y.; Li, X.; Zheng, B.; Chen, T. Dual-Scale Complementary Spatial-Spectral Joint Model for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 6772–6789. [\[CrossRef\]](#)
- Deng, W.; Shen, J.; Ding, J.; Zhao, H. Robust Dual-Model Collaborative Broad Learning System for Classification Under Label Noise Environments. *IEEE Internet Things J.* **2025**, *12*, 21055–21067. [\[CrossRef\]](#)
- Chunaev, P. Community detection in node-attributed social networks: A survey. *Comput. Sci. Rev.* **2020**, *37*, 100286. [\[CrossRef\]](#)
- Kumar, S.; Mallik, A.; Khetarpal, A.; Panda, B.S. Influence maximization in social networks using graph embedding and graph neural network. *Inf. Sci.* **2022**, *607*, 1617–1636. [\[CrossRef\]](#)
- Ramos, R.H.; Ferreira, C.O.L.; Simao, A. Human protein-protein interaction networks: A topological comparison review. *Heliyon* **2024**, *10*, e27278. [\[CrossRef\]](#)
- Jha, K.; Saha, S.; Singh, H. Prediction of protein-protein interaction using graph neural networks. *Sci. Rep.* **2022**, *12*, 8360. [\[CrossRef\]](#)
- Wu, L.; Cui, P.; Pei, J.; Zhao, L.; Guo, X. Graph neural networks: Foundation, frontiers and applications. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 4840–4841.
- Zhao, Q.; Li, L.; Chu, Y.; Yang, Z.; Wang, Z.; Shan, W. Efficient supervised image clustering based on density division and graph neural networks. *Remote Sens.* **2022**, *14*, 3768. [\[CrossRef\]](#)
- Marfo, W.; Tosh, D.K.; Moore, S.V. Enhancing Network Anomaly Detection Using Graph Neural Networks. In Proceedings of the 22nd Mediterranean Communication and Computer Networking Conference (MedComNet), Nice, France, 11–13 June 2024; pp. 1–10.
- Li, M.; Micheli, A.; Wang, Y.G.; Pan, S.; Lió, P.; Gnecco, G.S.; Sanguineti, M. Guest Editorial: Deep Neural Networks for Graphs: Theory, Models, Algorithms, and Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 4367–4372. [\[CrossRef\]](#)
- Villaizan-Valladolid, M.; Salvatori, M.; Carro, B.; Sanchez-Esguevillas, A.J. Graph neural network contextual embedding for deep learning on tabular data. *Neural Netw.* **2024**, *173*, 106180. [\[CrossRef\]](#)
- Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; Cui, P. Structural deep clustering network. In Proceedings of the Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 1400–1410.
- Tu, W.; Zhou, S.; Liu, X.; Guo, X.; Cai, Z.; Zhu, E.; Cheng, J. Deep fusion clustering network. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35; pp. 9978–9987.

29. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: A comprehensive review. *Comput. Soc. Netw.* **2019**, *6*, 11. [\[CrossRef\]](#)
30. Yu, P.; Tan, Z.; Lu, G.; Bao, B.K. Multi-view graph convolutional network for multimedia recommendation. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 October–3 November 2023; pp. 6576–6585.
31. Majumdar, A. Graph structured autoencoder. *Neural Netw.* **2018**, *106*, 271–280. [\[CrossRef\]](#)
32. Tu W.; Liao, Q.; Zhou, S.; Peng, X.; Ma, C.; Liu, Z.; Liu, X.; Cai, Z.; He, K. RARE: Robust Masked Graph Autoencoder. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 5340–5353. [\[CrossRef\]](#)
33. Jurek, A.; Bi, Y.; Wu, S.; Nugent, C. A survey of commonly used ensemble-based classification techniques. *Knowl. Eng. Rev.* **2014**, *29*, 551–581. [\[CrossRef\]](#)
34. Vega-Pons, S.; Ruiz-Shulcloper, J. A survey of clustering ensemble algorithms. *Int. J. Pattern Recognit. Artif. Intell.* **2011**, *25*, 337–372. [\[CrossRef\]](#)
35. Tao, Z.; Liu, H.; Li, S.; Fu, Y. Robust spectral ensemble clustering. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 367–376.
36. Liu, H.; Wu, J.; Liu, T.; Tao, D.; Fu, Y. Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1129–1143. [\[CrossRef\]](#)
37. Zhou, J.; Zheng, H.; Pan, L. Ensemble clustering based on dense representation. *Neurocomputing* **2019**, *357*, 66–76. [\[CrossRef\]](#)
38. Huang, D.; Wang, C.D.; Lai, J.H. Locally weighted ensemble clustering. *IEEE Trans. Cybern.* **2017**, *48*, 1460–1473. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Zhou, P.; Du, L.; Li, X. Adaptive consensus clustering for multiple k-means via base results refining. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 10251–10264. [\[CrossRef\]](#)
40. Zhou, P.; Liu, X.; Du, L.; Li, X. Self-paced adaptive bipartite graph learning for consensus clustering. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 62. [\[CrossRef\]](#)
41. Zhou, P.; Hu, B.; Yan, D.; Du, L. Clustering ensemble via diffusion on adaptive multiplex. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 1463–1474. [\[CrossRef\]](#)
42. Ren, Y.; Pu, J.; Yang, Z.; Xu, J.; Li, G.; Pu, X.; Philip, S.Y.; He, L. Deep clustering: A comprehensive survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 5858–5878. [\[CrossRef\]](#)
43. Chen, Y.; Wu, W.; Ou-Yang, L.; Wang, R.; Kwong, S. GRESS: Grouping Belief-Based Deep Contrastive Subspace Clustering. *IEEE Trans. Cybern.* **2024**, *55*, 148–160. [\[CrossRef\]](#)
44. Ye, X.; Zhao, J.; Zhang, L.; Guo, L. A nonparametric deep generative model for multimanifold clustering. *IEEE Trans. Cybern.* **2019**, *49*, 2664–2677. [\[CrossRef\]](#)
45. Chen, R.; Tang, Y.; Tian, L.; Zhang, C.; Zhang, W. Deep convolutional self-paced clustering. *Appl. Intell.* **2022**, *52*, 4858–4872.
46. Cai, J.; Zhang, Y.; Wang, S.; Fan, J.; Guo, W. Wasserstein Embedding Learning for Deep Clustering: A Generative Approach. *IEEE Trans. Multimed.* **2024**, *26*, 7567–7580. [\[CrossRef\]](#)
47. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 478–487.
48. Guo, X.; Gao, L.; Liu, X.; Yin, J. Improved deep embedded clustering with local structure preservation. In Proceedings of the International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 1753–1759.
49. Guo, X.; Liu, X.; Zhu, E.; Yin, J. Deep clustering with convolutional autoencoders. In Proceedings of the 24th International Conference on Neural Information Processing, Guangzhou, China, 14–18 November 2017; pp. 373–382.
50. Alqahtani, A.; Xie, X.; Deng, J.; Jones, M. A deep convolutional auto-encoder with embedded clustering. In Proceedings of the 25th IEEE International Conference on Image Processing, Athens, Greece, 7–10 October 2018; pp. 4058–4062.
51. 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 Conference on Artificial Intelligence, Tel-Aviv, Israel, 22 February–1 March 2022; Volume 36, No. 7, pp. 7603–7611.
52. Akkem, Y.; Biswas, S.K.; Varanasi, A. A comprehensive review of synthetic data generation in smart farming by using variational autoencoder and generative adversarial network. *Eng. Appl. Artif. Intell.* **2024**, *131*, 107881. [\[CrossRef\]](#)
53. Shirazi, H.; Muramudalige, S.R.; Ray, I.; Jayasumana, A.P.; Wang, H. Adversarial autoencoder data synthesis for enhancing machine learning-based phishing detection algorithms. *IEEE Trans. Serv. Comput.* **2023**, *16*, 2411–2422. [\[CrossRef\]](#)
54. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [\[CrossRef\]](#)
55. Pan, S.; Hu, R.; Fung, S.F.; Long, G.; Jiang, J.; Zhang, C. Learning graph embedding with adversarial training methods. *IEEE Trans. Cybern.* **2020**, *50*, 2475–2487. [\[CrossRef\]](#)
56. Oyewole, G.J.; Thopil, G.A. Data clustering: Application and trends. *Artif. Intell. Rev.* **2023**, *56*, 6439–6475. [\[CrossRef\]](#) [\[PubMed\]](#)
57. Jia, Y.; Liu, H.; Hou, J.; Zhang, Q. Clustering ensemble meets low-rank tensor approximation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, No. 9, pp. 7970–7978.

-
58. Jia, Y.; Tao, S.; Wang, R.; Wang, Y. Ensemble clustering via co-association matrix self-enhancement. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 11168–11179. [[CrossRef](#)] [[PubMed](#)]
 59. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [[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.