Attribute-Missing Graph Clustering Network

Wenxuan Tu[1], Renxiang Guan[1], Sihang Zhou[2], Chuan Ma[3], Xin Peng[1], Zhiping Cai[1], Zhe Liu[3], Jieren Cheng[4,5], Xinwang Liu[1]

[1]School of Computer, National University of Defense Technology, Changsha, China
[2]School of Intelligence Science and Technology, National University of Defense Technology, Changsha, China
[3]Zhejiang Lab, Hangzhou, China
[4]School of Computer Science and Technology, Hainan University, Haikou, China
[5]Hainan Blockchain Technology Engineering Research Center, Haikou, China
wenxuantu@163.com, sihangjoe@gmail.com, xinwangliu@nudt.edu.cn

Abstract

Deep clustering with attribute-missing graphs, where only a subset of nodes possesses complete attributes while those of others are missing, is an important yet challenging topic in various practical applications. It has become a prevalent learning paradigm in existing studies to perform data imputation first and subsequently conduct clustering using the imputed information. However, these "two-stage" methods disconnect the clustering and imputation processes, preventing the model from effectively learning clustering-friendly graph embedding. Furthermore, they are not tailored for clustering tasks, leading to inferior clustering results. To solve these issues, we propose a novel **A**ttribute-**M**issing **G**raph **C**lustering (**AMGC**) method to alternately promote clustering and imputation in a unified framework, where we iteratively produce the clustering-enhanced nearest neighbor information to boost the quality of data imputation and utilize the imputed information to implicitly refine the clustering distribution through model optimization. Specifically, in the imputation step, we take the learned clustering information as imputation prompts to help each attribute-missing sample gather highly correlated features within its clusters for data completion, such that the intra-class compactness can be improved. Moreover, to support reliable clustering, we maximize inter-class separability by conducting cost-efficient dual non-contrastive learning over the imputed latent features, which in turn promotes greater graph encoding capability for clustering subnetwork. Extensive experiments on five datasets have verified the superiority of AMGC against competitors.

Introduction

Deep graph clustering (DGC), as a fundamental task in unlabelled data analysis, aims to identify groups of nodes with similar properties or behaviors within a given graph. With the powerful representation learning capability of graph neural networks (Kipf and Welling 2017; Velickovic et al. 2018; Xu et al. 2019), DGC has been intensively researched and demonstrated encouraging performance across a wide range of practical applications, such as community detection (Park et al. 2022), face recognition (Wang et al. 2022b), and metagenomic binning (Xue et al. 2022).

The key prerequisite for the success of current DGC methods (Bo et al. 2020; Cui et al. 2020; Tu et al. 2021; Peng et al. 2021; Gong et al. 2022a,b; Liu et al. 2023; Hu et al. 2023; Yang et al. 2022, 2023) lies in the assumption that all samples within a graph are trustworthy and complete. However, such an assumption may not always hold in practical scenarios since it is hard to collect all information from graph data. The reasons behind this include but are not limited to the privacy-protecting policy, the copyright restriction, and the failures in data acquisition equipment. All these uncontrollable factors could easily trigger data-sparse and data-absent issues that adversely affect clustering performance. Based on the absence of node attributes, the graphs with absent attributes can be broadly classified into two categories: 1) attribute-incomplete graphs, where only a subset of attributes for all nodes is absent; and 2) attribute-missing graphs, where all attributes for specific nodes are absent. In this study, we focus specifically on the second category as it is more challenging and holds relevance to numerous real-world applications. For example, in a citation graph, some papers are entirely not accessible due to copyright protection. Moreover, within the co-purchase graph, consumers tend to abstain entirely from offering feedback for certain items, primarily driven by privacy concerns. However, most existing DGC methods lack a dedicated feature completion mechanism to handle attribute-missing samples. This poses a significant challenge in learning effective graph embedding for accurate clustering on attribute-missing graphs.

To learn strong graph neural networks (GNNs) capable of handling attribute-missing data for clustering, recent studies propose two types of graph learning methods. The first category is termed embedding alignment-based methods, which firstly conduct the structure-attribute embedding distribution matching to reconstruct latent features of attribute-missing samples (Chen et al. 2022; Li et al. 2022b) and then utilize the decoded information for clustering. The second category, termed data imputation-based methods, addresses this issue by completing missing attributes using data imputation techniques (Yoo et al. 2022; Tu et al. 2022; Jin et al. 2023) and subsequently performing clustering in the latent space. Though demonstrating impressive clustering performance, the above "two-stage" methods share the following limitations: 1) isolate the clustering and imputation processes, preventing two steps from negotiating with each other to learn clustering-friendly graph embedding; and 2) not tailored for clustering tasks. Most of them complete missing attributes

in the noise latent space by searching for reliable signals from a global aspect, which may potentially increase the risk of constructing a noisy connection between two samples not belonging to the same cluster. In this circumstance, the imputed information is less accurate, posing challenges in effectively estimating associations between the imputed nodes and clusters. Consequently, the intra-class samples exhibit scattering, while the inter-class samples tend to mix together, resulting in sub-optimal performance.

Based on the above observations, we find that attribute-missing DGC has emerged as a pressing real-world demand yet has not been sufficiently explored. This motivates us to propose an efficient yet effective DGC framework termed **A**ttribute-**M**issing **G**raph **C**lustering (AMGC), where we iteratively produce the clustering-enhanced nearest neighbor information to boost the quality of data imputation and utilize the imputed information to implicitly refine the clustering distribution through model optimization. To this end, we design a cluster-oriented imputation-then-refinement scheme, which is a key innovation of our work. On the one hand, we take the learned clustering pseudo labels as imputation prompts to help attribute-missing samples identify the most relevant clues within their respective clusters. By filtering noisy connections among different clusters and enhancing the most reliable ones within each cluster, the network is encouraged to boost the quality of data imputation, making the learned graph embedding more compact and distinguishable for clustering. On the other hand, we construct cost-efficient positive and negative cluster center pairs using the imputed information and develop a novel dual non-contrastive clustering loss to conduct discrimination between positive and negative pairs. By doing this, the clustering sub-network is enabled to generate well-separated clusters for subsequent imputation while avoiding the expensive negative sampling step evident in contrastive learning. As the clustering pseudo labels become more reliable and the imputed latent features become more accurate, the model can learn clustering-friendly graph embedding for better performance. We summarize the contributions of this work:

- To the best of our knowledge, this is the first attempt to investigate a unified learning paradigm for the attribute-missing DGC problem, which is more practical and challenging than the attribute-complete counterpart.
- A novel attribute-missing DGC method termed AMGC is proposed, where clustering and imputation are promoted alternately within a unified optimization framework. In addition, a novel cluster-oriented imputation scheme and a new dual non-contrastive clustering loss are designed to facilitate high-quality data completion and promote clustering-friendly graph embedding, respectively.
- Extensive experiments on five graph datasets have solidly demonstrated the effectiveness and superiority of AMGC compared to other competitors.

Related Work

Attribute-Complete Deep Graph Clustering

Graph Neural Networks (GNNs) have been de-facto standard deep learning models for graph data analysis (Liang et al. 2023a,b; Peng et al. 2024). Due to the strength of GNNs, graph clustering techniques have made remarkable advancements in recent years. As one of the most representative clustering paradigms, generative graph clustering aims to provide self-supervision signals by reconstructing node attributes or graph structure and predict clustering labels by applying or incorporating clustering techniques (Wang et al. 2019; Tao et al. 2019; Pan et al. 2020; Bo et al. 2020; Tu et al. 2021; Peng et al. 2021). Another vital research line in this field is contrastive graph clustering, which acquires clustering-friendly graph embedding for clustering by pulling positive samples close while pushing negative samples away (Gong et al. 2022b; Xia et al. 2022; Liu et al. 2022, 2023; Tu et al. 2023b). One common underlying assumption embraced by these methods is that all node attributes are available and trustworthy. While in practical applications, this assumption may be invalid, and the presence of missing data poses a significant challenge for achieving satisfactory clustering performance. In contrast, we address the problem of deep clustering with attribute-missing graphs and advocate a unified learning paradigm that alternates between clustering and imputation to enhance performance.

Clustering with Attribute-Missing Data

Recently, clustering with attribute-missing data has attracted significant attention from researchers (Liu 2021; Jin et al. 2021; He et al. 2022; Cui et al. 2022; Rossi et al. 2022; Tu et al. 2022; Yoo et al. 2022; Xu et al. 2022; Jin et al. 2023). For example, SAT (Chen et al. 2022) and CSAT (Li et al. 2022b) perform distribution matching between the attribute embedding and the structure embedding to estimate missing attributes for clustering. Another type of method completes missing latent features for clustering by various data imputation techniques like structured variational inference (Yoo et al. 2022), node similarity preserving (Tu et al. 2022), and generative adversarial learning (Jin et al. 2023). Despite their success in achieving high-quality data completion, these methods disconnect the clustering and imputation processes, thereby hindering effective negotiation between the two learning processes. Moreover, none of them are specialized designs to handle clustering tasks, and thus the learned graph embedding is sub-optimal for clustering in extremely attribute-missing scenarios. To solve these issues, several advanced algorithms, such as incomplete multiple kernel clustering (IMKC) and incomplete multi-view clustering (IMVC), have been proposed to jointly impute and partition multi-view data with missing information in some specific views (Liu et al. 2020, 2021; Liu 2021; Xu et al. 2022; Xia et al. 2023). For instance, IMKAM (Liu 2021) initially treats absent kernel elements as variables that can be completed through optimizing a clustering objective, and the updated variables are used in turn to perform the subsequent clustering. DIMVC (Xu et al. 2022) alternately conducts the multi-view cluster complementarity for data completion and the multi-view clustering consistency over the completed latent features. Despite their achievements in processing non-graph data, the simple integration of clustering and imputation may be less effective in handling graph data. This is because the non-Euclidean property of graphs

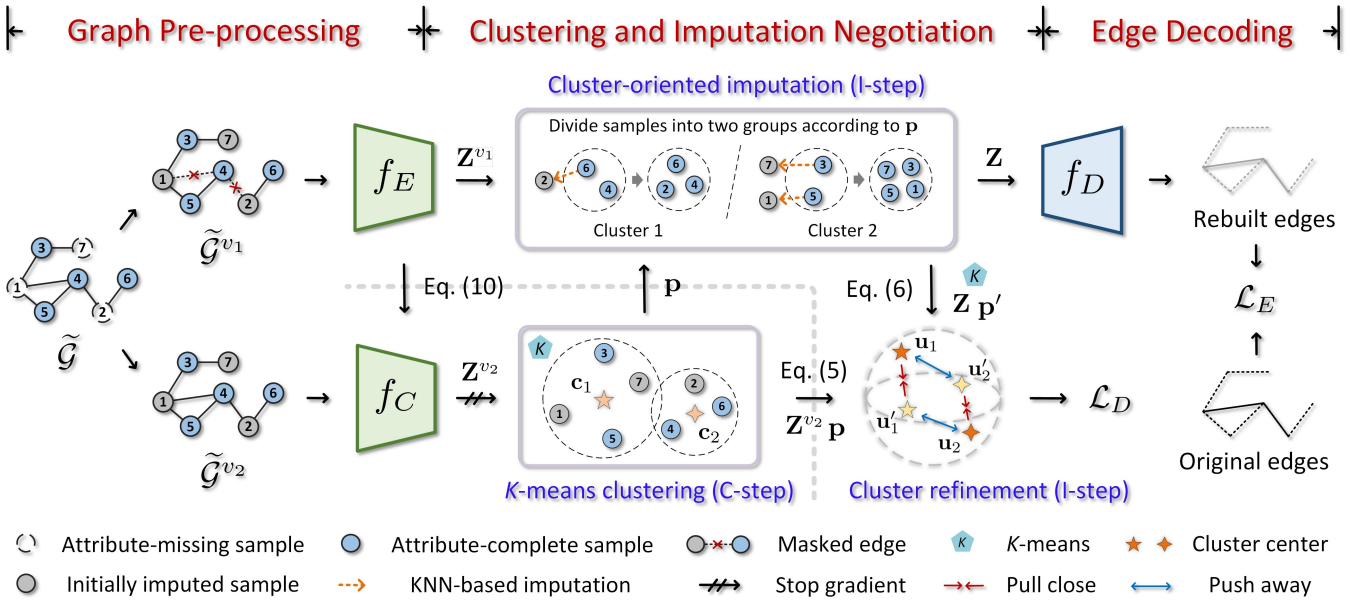

Figure 1: The overall architecture of AMGC. The clustering and imputation processes are integrated into a unified two-step alternate optimization framework. Specifically, in the clustering step, we estimate the clustering distribution from encoded graph embedding via K-means. While in the imputation step, the cluster-oriented imputation scheme and the dual non-contrastive clustering loss $\mathcal{L}_D$ encourage the model to enhance intra-class compactness and maximize inter-class separability, respectively.

could easily trigger uncertainty in estimating missing data (Tu et al. 2023a), possibly affecting subsequent clustering. To improve the imputation stability and quality, we simultaneously filter noisy connections between different clusters and strengthen the most trustworthy ones within each cluster using the learned clustering information. To support reliable clustering, we in turn leverage the imputed information to facilitate well-separated clusters through model optimization.

Method

In this part, we begin with an illustration of basic notations and definitions, and then present the proposed AMGC in detail. An overview of AMGC is shown in Fig. 1.

Notations and Definitions

Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with C clusters, where $\mathcal{V} = \{\mathbf{v}_n\}_{n=1}^{N}$, $\mathcal{E}$, and N are the node set, the edge set, and the number of all nodes, respectively. In classic graph learning, $\mathcal{G}$ is usually described by an attribute matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ and a normalized adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ (Kipf and Welling 2017), where D refers to the node attribute dimension.

Definition 1 (Attribute-Missing Graph). *In attribute-missing deep graph clustering, with the existence of missing attributes, we define $\mathcal{V}^c = \{\mathbf{v}_n^c\}_{n=1}^{N^c}$ and $\mathcal{V}^m = \{\mathbf{v}_n^m\}_{n=1}^{N^m}$ to form the set of attribute-complete nodes and the set of attribute-missing nodes, respectively. Accordingly, $\mathcal{V} = \mathcal{V}^c \cup \mathcal{V}^m$, $\mathcal{V}^c \cap \mathcal{V}^m = \varnothing$ and $N = N^c + N^m$. Based on these notations, an attribute-missing graph could be formulated as $\widetilde{\mathcal{G}} = \{\mathcal{V}^c, \mathcal{E}\}$.*

Definition 2 (Learning Task). *In this paper, we consider node clustering on attribute-missing graphs. The goal of our task is to learn a clustering sub-network f_C and an imputation sub-network $f_A = f_D \circ f_E$, where f_C is a GNN-based encoder, and f_A is composed of a GNN-based encoder f_E and an MLP-based decoder f_D. Specifically, f_C outputs the clustering-friendly graph embedding matrix $\mathbf{Z}^{v_2} \in \mathbb{R}^{N \times d}$ such that $d \ll D$, which provides the learned clustering information to guide the data completion. Similarly, f_A produces the well-imputed graph embedding matrix $\mathbf{Z} \in \mathbb{R}^{N \times d}$, which is subsequently utilized to refine the clustering distribution. Both clustering and imputation processes are integrated in a two-step alternate optimization framework.*

Graph Pre-Processing

Before training AMGC, we first generate two augmented graphs as input views from $\widetilde{\mathcal{G}}$. These views are then fed into the graph encoder f_E and the momentum graph encoder f_C, respectively. Specifically, to ease the model training under a high attribute-missing ratio r_a, we fill missing attributes with a set of initial values $\mathcal{T}^m = \{\mathbf{t}_n^m\}_{n=1}^{N^m}$, where $\mathbf{t}_n^m \in \mathbb{R}^D$ represents the zero-filling or Laplacian smoothing feature vector. Moreover, under an edge-masking ratio r_e, we sample a subset of masked edges $\widetilde{\mathcal{E}}$ from $\mathcal{E}$ following a Bernoulli distribution (Li et al. 2023a). With these mathematical formulations, two augmented graph views could be defined as $\widetilde{\mathcal{G}}^{v_1} = \{\mathcal{V}^c, \mathcal{T}^m, \complement_{\mathcal{E}}\widetilde{\mathcal{E}}\}$ and $\widetilde{\mathcal{G}}^{v_2} = \{\mathcal{V}^c, \mathcal{T}^m, \mathcal{E}\}$, respectively, where $\complement_{\mathcal{E}}\widetilde{\mathcal{E}}$ is the complementary set of $\widetilde{\mathcal{E}}$. After that, we map these two augmented graphs into the latent space for alternately optimizing clustering and imputation

processes in the next step:

$$\mathbf{Z}^{v_1} = f_E(\widetilde{\mathcal{G}}^{v_1} \,|\, \Theta_{f_E}) = f_E(\widetilde{\mathbf{A}}, \widetilde{\mathbf{X}} \,|\, \Theta_{f_E}), \tag{1}$$

$$\mathbf{Z}^{v_2} = f_C(\widetilde{\mathcal{G}}^{v_2} \,|\, \Theta_{f_C}) = f_C(\mathbf{A}, \widetilde{\mathbf{X}} \,|\, \Theta_{f_C}), \tag{2}$$

where Θ_{f_E} and Θ_{f_C} denotes the learnable parameters of two graph encoders. $\widetilde{\mathbf{A}} \in \mathbb{R}^{N\times N}$ and $\widetilde{\mathbf{X}} \in \mathbb{R}^{N\times D}$ denotes the normalized masked adjacency matrix and the initially imputed attribute matrix.

Clustering and Imputation Negotiation

As previously discussed, we are facing a challenging yet under-explored task: utilizing limited graph information to perform clustering and imputation simultaneously. To bridge this gap, an intuitive solution is to conduct these two steps in an alternate optimization manner: 1) complete attribute-missing samples under the guidance of clustering; 2) update the clustering distribution with imputed information. To this end, we design a simple yet effective **C**luster-oriented **I**mputation-**T**hen-r**E**finement (CITE) scheme. As shown in Fig. 1, the core components of CITE consist of two parts, *i.e.,* cluster-oriented imputation and cluster refinement.

Cluster-Oriented Imputation During the data imputation process, to collect K-nearest nodes within respective clusters, we develop a cluster-oriented information filtering and enhancing operation that mainly includes four steps. Firstly, we perform the classical K-means algorithm $\mathcal{K}(\cdot)$ (Hartigan and Wong 1979) over $\mathbf{Z}^{v_2}$ to obtain prior clustering information:

$$\{\mathbf{C}, \mathbf{p}\} \leftarrow \mathcal{K}(\mathbf{Z}^{v_2}), \tag{3}$$

where $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_C] \in \mathbb{R}^{C\times d}$ denotes the cluster center matrix, and $\mathbf{p} = [p_{\mathbf{z}_1^{v_2}}, p_{\mathbf{z}_2^{v_2}}, \ldots, p_{\mathbf{z}_N^{v_2}}] \in \mathbb{R}^N$ indicates the predicted cluster assignment vector. In other words, $p_{\mathbf{z}_n^{v_2}} = i$ implies that the predicted clustering pseudo label of sample $\mathbf{z}_n^{v_2}$ is i, where $i \in [1, C]$.

Next, according to $\mathbf{p}$, the graph embedding matrix $\mathbf{Z}^{v_1}$ could be grouped into C sample sets $\mathcal{H} = \{\mathcal{C}_i\}_{i=1}^{C}$, where all samples belonging to $\mathcal{C}_i$ have a common predicted clustering pseudo label i. After that, within these sample sets, we search K nearest neighbors for each attribute-missing node based on the sample similarity (*e.g.,* Euclidean distance). Finally, these selected samples are gathered to update $\mathbf{Z}^{v_1}$:

$$\mathbf{Z} = \mathcal{O}(\mathbf{Z}^{v_1},\, \mathbf{p},\, K), \tag{4}$$

where $\mathcal{O}(\cdot)$ refers to the cluster-oriented information filtering and enhancing operation. $\mathbf{Z} \in \mathbb{R}^{N\times d}$ denotes the imputed graph embedding matrix. By doing so, when completing attribute-missing samples, some noisy connections (*e.g.,* two connected samples not belonging to the same cluster) could be eliminated among different clusters while more reliable ones are encouraged to be identified within the same cluster. Consequently, the network is promoted to learn a more compact clustering pattern for better performance.

Cluster Refinement Besides a compact cluster structure, an ideal clustering distribution should exhibit well-separated clusters (Li et al. 2021; Huang et al. 2023). To this end, we leverage the imputed information to implicitly refine the clustering distribution by designing a dual non-contrastive clustering loss $\mathcal{L}_D$. It is worth noting that the cluster centers (*i.e.,* $\mathbf{C}$) obtained by K-means are detached and cannot be applied to the loss computation for gradient back-propagation directly. Inspired by the success of ProPos (Huang et al. 2023), we reestimate all cluster centers by associating the learned graph embedding with the cluster assignment posterior probability. Specifically, we calculate $\mathbf{C}' \in \mathbb{R}^{C\times d}$ and $\mathbf{p}' \in \mathbb{R}^N$ over $\mathbf{Z}$, similar to Eq. (3), and then redefine the cluster center vectors $\mathbf{c}_i$, $\mathbf{c}'_i \in \mathbb{R}^d$ as $\mathbf{u}_i$, $\mathbf{u}'_i \in \mathbb{R}^d$:

$$\mathbf{u}_i = \frac{\sum_{\mathbf{z}_n^{v_2}\in\mathcal{C}_i}^{|\mathcal{C}_i|} w_n \mathbf{z}_n^{v_2}}{\| \sum_{\mathbf{z}_n^{v_2}\in\mathcal{C}_i}^{|\mathcal{C}_i|} w_n \mathbf{z}_n^{v_2} \|_2}, \tag{5}$$

$$\mathbf{u}'_i = \frac{\sum_{\mathbf{z}_n\in\mathcal{C}'_i}^{|\mathcal{C}'_i|} w'_n \mathbf{z}_n}{\| \sum_{\mathbf{z}_n\in\mathcal{C}'_i}^{|\mathcal{C}'_i|} w'_n \mathbf{z}_n \|_2}, \tag{6}$$

where w_n and w'_n represent the normalized posterior probabilities of cluster assignment, calculated from $\mathbf{p}$ and $\mathbf{p}'$ respectively. $\mathbf{u}_i$ and $\mathbf{u}'_i$ are the updated cluster center vectors with gradient, and $\|\cdot\|_2$ denotes the ℓ_2-normalization.

According to Eq. (5) and Eq. (6), the final dual non-contrastive clustering loss $\mathcal{L}_D$ can be formulated as follows:

$$\mathcal{L}_D = \frac{1}{C}\Big(\sum_{i=1}^{C} \frac{\mathbf{u}_i \cdot \mathcal{T}(\{\mathbf{u}'_j\}_{j\neq i}^{C})}{\|\mathbf{u}_i\| \|\mathcal{T}(\{\mathbf{u}'_j\}_{j\neq i}^{C})\|} - \sum_{i=1}^{C} \frac{\mathbf{u}_i \cdot \mathbf{u}'_i}{\|\mathbf{u}_i\| \|\mathbf{u}'_i\|}\Big), \tag{7}$$

where $\mathcal{T}(\cdot)$ is the random cluster center-sampling operation. The first term minimizes the similarity between the negative cluster center pair across two views, while the second term maximizes the consistency within the positive cluster center pair. With Eq. (7), the clustering sub-network is efficiently promoted to enlarge the distance between inter-cluster samples through model optimization, leading to more clustering-friendly graph embedding and clearer clustering partitions.

Edge Decoding and Optimization

Edge Decoding To boost the learning stability of the imputation sub-network, we impose a simple MLP-based decoder f_D to recover masked edges. The masked edge reconstruction loss can be formulated as follows:

$$s_{n,n'} = \mathrm{Sigm}\big(f_D(\mathbf{z}_n, \mathbf{z}_{n'})\big), \tag{8}$$

$$\mathcal{L}_E = -\frac{1}{|\widetilde{\mathcal{E}}|} \sum_{e_{nn'}\in\widetilde{\mathcal{E}}} \log s_{n,n'} - \frac{1}{|\widehat{\mathcal{E}}|} \sum_{e_{ll'}\in\widehat{\mathcal{E}}} \log(1 - s_{l,l'}), \tag{9}$$

where $\mathrm{Sigm}(\cdot)$ is sigmoid activation function. $s_{n,n'}$ is the probability of node $\mathbf{v}_n$ and node $\mathbf{v}_{n'}$ being connected. $\mathbf{z}_n$ and $\mathbf{z}_{n'}$ denote the corresponding node embedding vectors encoded by f_E. $\widetilde{\mathcal{E}}$ is a set of positive edges while $\widehat{\mathcal{E}}$ is a set of negative edges sampled from the graph, where $|\widetilde{\mathcal{E}}| = |\widehat{\mathcal{E}}|$.

Algorithm 1: The learning procedure of AMGC

Input: Augmented graphs $\widetilde{\mathcal{G}}^{v_1}$, $\widetilde{\mathcal{G}}^{v_2}$; maximum epoch E; edge-masking ratio r_e; nearest neighbors K, learning rate η.
Output: Clustering results $\mathbf{p}$.

Initialize $\{\Theta_{f_C}, \Theta_{f_E}, \Theta_{f_D}\}$ with an Xavier manner
for $e = 1$ to E **do**
  Update $\widetilde{\mathcal{G}}^{v_1}$ by randomly masking edges with r_e
  /* Clustering step (C-step) */
  Update Θ_{f_C} by Eq. (10)
  Obtain $\mathbf{Z}^{v_2}$ from $\widetilde{\mathcal{G}}^{v_2}$ with f_C by Eq. (2)
  Obtain $\mathbf{C}$ and $\mathbf{p}$ from $\mathbf{Z}^{v_2}$ with K-means by Eq. (3)
  /* Imputation step (I-step) */
  Obtain $\mathbf{Z}^{v_1}$ from $\widetilde{\mathcal{G}}^{v_1}$ with f_E by Eq. (1)
  Obtain $\mathbf{Z}$ from $\mathbf{Z}^{v_1}$ based on $\mathbf{p}$ and K by Eq. (4)
  Calculate $\mathcal{L}_D$ and $\mathcal{L}_E$ by Eq. (5) - Eq. (9)
  Calculate $\mathcal{L}$ by Eq. (11)
  Update $\{\Theta_{f_E}, \Theta_{f_D}\}$ by calculating:
  $\Theta_{f_E} \leftarrow \Theta_{f_E} - \eta \nabla_{\Theta_{f_E}} \mathcal{L}$
  $\Theta_{f_D} \leftarrow \Theta_{f_D} - \eta \nabla_{\Theta_{f_D}} \mathcal{L}$
end for
return $\mathbf{p}$

Optimization To facilitate the negotiation between clustering and imputation, the proposed AMGC is implemented in a two-step alternate optimization framework.

Optimizing Θ_{f_C} with fixed Θ_{f_E} and Θ_{f_D}. In the clustering step, we update the parameters of f_C by exponential moving average (EMA) (Grill et al. 2020) with those of f_E and f_D fixed at each epoch. Specifically, we define Θ_{f_E} and Θ_{f_C} as the parameters of f_E and f_C, respectively, and Θ_{f_C} are updated by incorporating historical information from Θ_{f_E}:

$$\Theta_{f_C} \leftarrow \gamma \Theta_{f_C} + (1-\gamma)\Theta_{f_E}, \tag{10}$$

where γ is the momentum factor that has been determined empirically and fixed as 0.999. The goal of this step is to predict the clustering pseudo label for each sample, which serves as a hint for subsequent data imputation.

Optimizing Θ_{f_E} and Θ_{f_D} with fixed Θ_{f_C}. In the imputation step, we update the imputation sub-network by incorporating $\mathcal{L}_D$ and $\mathcal{L}_E$ into the final objective function:

$$\mathcal{L} = \min \ \mathcal{L}_D + \mathcal{L}_E. \tag{11}$$

Based on Eq. (11), f_E and f_D are optimized through back-propagation. It is worth noting that f_C is detached from the gradient back-propagation. The detailed learning procedure of AMGC is shown in Algorithm 1.

Computational Complexity

The time complexity of the proposed AMGC could be discussed from the network and loss computation aspects. For two GNN-based encoders, the time complexities of f_C and f_E are $\mathcal{O}\big(Nd^2(L-1) + NdD + |\mathcal{E}|dL\big)$, where N, L, $|\mathcal{E}|$ are the number of all nodes, encoder layers, and edges, respectively. Moreover, D and d are the dimensions of raw attributes and latent features, respectively. For the MLP-based decoder, the time complexity of f_D is $\mathcal{O}\big(Nd^2(L-1) + NdD\big)$. For the loss functions, the time complexities of $\mathcal{L}_D$ and $\mathcal{L}_E$ are $\mathcal{O}(C)$ and $\mathcal{O}(|\widetilde{\mathcal{E}}|)$. The overall time complexity of AMGC for each training iteration is $\mathcal{O}\big(Nd^2(L-1) + NdD + |\mathcal{E}|dL + C + |\widetilde{\mathcal{E}}|\big) \approx \mathcal{O}(N + |\mathcal{E}|)$. We can observe that the time complexity of AMGC is linear with the numbers of all nodes N and edges $|\mathcal{E}|$, making the proposed AMGC theoretically efficient and scalable.

Dataset	Samples	Edges	Dimension	Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3
Co.CS	18,333	81,894	6,805	15
Co.Physics	34,493	247,962	8,415	5

Table 1: Dataset summary.

Experiments

Experiment Setup

Datasets To evaluate the proposed AMGC, we choose five public graph datasets, *i.e.,* small-scale Cora and Citeseer, medium-scale Pubmed and Coauthor CS (Co.CS), and large-scale Coauthor Physics (Co.Physics). Details of these graph datasets are introduced below.

- **Citation Graphs.** In citation graphs, nodes typically represent papers, where node attributes correspond to keywords extracted from the papers. Furthermore, edges denote cross-citation connections, while categories reflect the topics covered in the papers. Please note that nodes within citation graphs may occasionally represent authors, institutions, or other entities (Wu et al. 2023). The used citation graphs include Cora, Citeseer, and Pubmed.
- **Co-Authorship Graphs.** Coauthor CS and Coauthor Physics are co-authorship graphs based on the Microsoft Academic Graph from the KDD Cup 2016 challenge. Here, nodes are authors, that are connected by an edge if they co-authored a paper. Node features represent paper keywords for each author's papers, and class labels indicate the most active fields of study for each author.

Baseline Methods We compare AMGC against six state-of-the-art attribute-complete deep graph clustering methods, including SDCN (Bo et al. 2020), GDCL (Zhao et al. 2021), AGCN (Peng et al. 2021), AGC-DRR (Gong et al. 2022b), HSAN (Liu et al. 2023), and CCGC (Yang et al. 2023). In addition, we report the performance of two non-graph attribute-missing clustering methods (*i.e.,* IMKAM (Liu 2021) and DIMVC (Xu et al. 2022)) and three graph counterparts (*i.e.,* SAT (Chen et al. 2022), ITR (Tu et al. 2022), and SVGA (Yoo et al. 2022)).

Implementation Details To ensure a fair comparison, the reported results of AMGC[1] and all compared methods were conducted on the same device and under identical configuration settings. In our attribute-missing settings, we randomly

[1] https://github.com/WxTu/AMGC

Dataset	Metric	SDCN WWW 20	GDCL IJCAI 21	AGCN MM 21	AGC-DRR IJCAI 22	HSAN AAAI 23	CCGC AAAI 23	AMGC Ours
Cora	ACC	34.61±2.32	24.76±1.84	39.88±3.18	43.26±7.09	57.94±1.27	37.93±2.68	**66.65±2.04**
	NMI	12.56±4.95	4.74±2.21	19.16±1.43	23.26±8.51	42.01±1.24	22.37±3.03	**47.99±1.67**
	ARI	6.38±4.74	0.74±1.04	14.09±3.75	16.23±8.41	33.14±1.78	11.45±2.81	**43.40±1.90**
	F1	21.78±4.23	7.82±2.16	33.96±2.64	31.09±8.21	58.77±0.89	36.91±4.12	**61.02±2.38**
Citeseer	ACC	39.16±1.32	27.39±2.19	42.22±2.78	30.95±4.26	44.22±0.52	39.63±1.95	**60.92±2.01**
	NMI	16.19±1.41	7.53±2.38	19.01±1.74	13.26±3.41	22.29±1.39	17.13±1.34	**32.93±1.71**
	ARI	7.29±2.37	2.25±1.08	13.31±2.17	2.35±3.43	13.32±1.08	11.83±2.34	**33.73±2.25**
	F1	33.14±3.09	12.16±4.02	37.22±1.67	29.10±5.26	42.18±2.54	38.88±2.54	**57.33±2.05**
Pubmed	ACC	48.69±3.21	47.00±0.39	41.85±2.28	60.71±1.72	OOM	42.7±1.49	**64.56±1.50**
	NMI	4.99±2.78	10.84±0.49	0.93±1.08	16.26±2.14		2.43±1.75	**24.58±2.21**
	ARI	5.00±2.80	7.39±0.27	1.37±1.18	17.35±2.44		1.24±0.87	**24.19±2.39**
	F1	38.78±5.65	28.36±10.56	36.05±0.79	59.80±2.43		34.28±3.16	**64.52±1.26**
Co.CS	ACC	48.85±2.56	40.18±1.63	52.13±5.88	59.08±2.26	OOM	64.88±1.51	**72.61±1.08**
	NMI	45.65±3.66	41.67±1.29	49.51±5.30	64.16±1.97		63.21±0.82	**73.91±0.39**
	ARI	38.39±3.64	15.32±1.68	41.44±7.43	49.27±2.86		55.08±2.22	**64.62±0.66**
	F1	23.59±2.38	3.62±2.78	25.02±5.54	45.09±1.41		54.17±2.76	**68.34±1.88**
Co.Physics	ACC	66.59±5.87	OOM	63.64±4.06	OOM	OOM	OOM	**77.68±3.78**
	NMI	40.22±8.84		34.98±7.68				**62.77±1.46**
	ARI	41.75±14.36		43.94±6.91				**69.00±6.25**
	F1	44.15±6.04		43.19±6.61				**67.58±2.68**

Table 2: Performance comparison among six attribute-complete deep graph clustering methods and the proposed AMGC. The average clustering results of ten runs on five graph datasets are reported. "OOM" means the out-of-memory failure on 24GB RTX 3090 GPU and 64G RAM. The best and runner-up results are shown in bold and underline, respectively.

select 40% of the nodes with complete attributes from the raw graph, while removing all attributes from the remaining 60% of nodes. For baseline methods, we use their source code and report the reproduced performance. For AMGC, we first train the model for at least 100 epochs until convergence using an Adam optimizer, and then perform K-means over the graph embedding matrix learned by f_C. Please note that the graph encoder f_E could be pre-trained with only the edge reconstruction loss $\mathcal{L}_E$ before it is trained together with other components of the model. According to the results of hyper-parameter sensitivity testing, we set the nearest neighbors K and the edge-masking ratio r_e as 2 and 0.7, respectively. Moreover, the learning rate, the latent dimension, the dropout rate, and the weight decay are set to 1e-3, 256, 0.4, and 5e-4 as default, respectively. To alleviate the adverse influence of randomness, we repeat each experiment 10 times and report the average value with standard deviation.

Clustering Metrics We adopt four widely-used clustering metrics to evaluate all compared methods, *i.e.,* Accuracy (ACC), Normalized Mutual Information (NMI), Average Rand Index (ARI), and macro F1-score (F1) (Wang et al. 2022a; Wan et al. 2022, 2023; Li et al. 2022a, 2023b).

Performance Comparison

Table 2 and Table 3 report the node clustering performance of twelve methods on five datasets. From these results, some significant observations can be summarized: 1) taking the results on Co.CS for example, AMGC significantly outperforms AGC-DRR and CCGC by 13.53% and 7.73% ACC, respectively. These results verify that AMGC can effectively learn clustering-friendly graph embedding in the attribute-missing scenario; 2) AMGC consistently outperforms SAT, SVGA, and ITR, with margins going up to 31.04% and 25.01% ACC on Cora and Pubmed. These improvements demonstrate the effectiveness of alternately promoting clustering and imputation into a unified optimization framework; 3) compared to IMKAM and DIMVC, our method achieves very promising performance. This is because instead of simply integrating clustering and imputation, AMGC effectively filters noisy connections and enhances the most reliable ones by incorporating clustering information into the data imputation process. In turn, it leverages the imputed information to facilitate well-separated clusters through model optimization; and 4) on Co.Physics, most baseline methods suffer from out-of-memory failure while ours does not, verifying its potential efficiency in large-scale cases. Moreover, the clustering results of SVGA are more stable than ours, possibly attributed to imposing strict Gaussian distribution assumptions on the latent variables. However, this would make it hard for the model to accurately reflect the true data distribution, leading to biased graph embedding and inferior performance.

T-SNE Visualization

We compare T-SNE (der Maaten and Hinton 2008) visualization results of seven methods on Cora and Co.CS. In Fig. 2, the samples with different colors indicate different categories predicted by methods. As seen, AMGC presents

Dataset	Metric	IMKAM * TPAMI 21	DIMVC * AAAI 22	SAT TPAMI 22	SVGA WWW 22	ITR IJCAI 22	AMGC Ours
Cora	ACC	23.06±1.72	29.58±3.57	49.87±1.2	40.87±1.92	35.61±0.76	**66.65±2.04**
	NMI	2.98±0.95	9.93±2.34	30.82±0.53	26.49±2.58	17.72±0.75	**47.99±1.67**
	ARI	0.33±1.44	5.33±1.96	23.49±0.84	14.39±1.51	3.44±0.29	**43.40±1.90**
	F1	16.04±1.68	24.22±4.13	49.53±0.76	37.91±3.96	31.12±0.99	**61.02±2.38**
Citeseer	ACC	21.12±0.29	29.39±3.66	34.55±0.32	44.19±1.47	32.61±1.19	**60.92±2.01**
	NMI	5.12±0.19	7.75±2.49	11.0±0.25	23.45±1.37	15.17±1.59	**32.93±1.71**
	ARI	-0.13±0.04†	4.64±2.71	9.01±0.15	13.08±1.23	3.41±0.46	**33.73±2.25**
	F1	14.69±0.61	24.37±4.30	32.01±0.24	42.96±2.22	33.16±1.51	**57.33±2.05**
Pubmed	ACC	OOM	42.77±1.65	47.89±0.02	39.55±0.16	43.57±0.82	**64.56±1.50**
	NMI		4.75±0.70	12.5±0.01	0.13±0.02	8.2±2.98	**24.58±2.21**
	ARI		1.82±0.41	10.4±0.01	-0.11±0.03†	5.55±2.35	**24.19±2.39**
	F1		38.54±1.86	45.87±0.0	32.15±0.2	30.44±2.75	**64.52±1.26**
Co.CS	ACC	OOM	27.92±3.14	37.63±0.85	56.56±1.71	OOM	**72.61±1.08**
	NMI		20.82±4.57	40.64±0.24	61.28±0.78		**73.91±0.39**
	ARI		6.68±2.56	26.61±0.26	39.2±1.37		**64.62±0.66**
	F1		11.77±1.73	29.03±0.76	47.48±3.34		**68.34±1.88**
Co.Physics	ACC	OOM	42.79±3.12	OOM	60.22±0.37	OOM	**77.68±3.78**
	NMI		6.06±2.38		53.61±0.15		**62.77±1.46**
	ARI		-2.03±3.52†		39.39±0.17		**69.00±6.25**
	F1		18.12±4.35		64.62±0.27		**67.58±2.68**

* IMKAM and DIMVC are non-graph attribute-missing clustering methods, where we take the attribute-missing matrix and adjacency matrix as two input views in our settings.
† Note that the range of ARI values is [-1, 1], and negative results indicate poor performance.

Table 3: Performance comparison among six attribute-missing clustering methods. The average clustering results of ten runs on five graph datasets are reported. "OOM" means the out-of-memory failure on 24GB RTX 3090 GPU and 64G RAM. The best and runner-up results are shown in bold and underline, respectively.

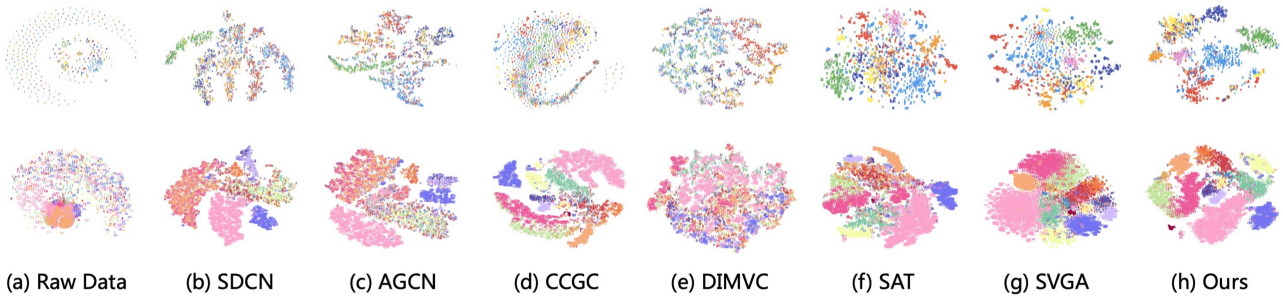

Figure 2: T-SNE visualization. The first and second rows correspond to the visual results on Cora and Co.CS, respectively. In our settings, we utilize the T-SNE toolkit to visualize the attribute-complete samples in the raw data space and all samples in the latent space. The proposed AMGC presents clearer partitions and denser cluster structures than the other six competitors.

clearer partitions and denser cluster structures than its competitors, indicating that AMGC can learn more compact and discriminative graph embedding for clustering.

Ablation Study

To demonstrate the effectiveness of the proposed CITE scheme, we compare AMGC with its three variants on five datasets. Concretely, "w/o $\mathcal{L}_D$" and "w/o CI" denote two AMGC variants with the dual non-contrastive clustering loss and the cluster-oriented imputation scheme being removed, respectively. "Base" denotes AMGC without both components. As seen in Table 4, we can find that: 1) compared to "Base", "w/o $\mathcal{L}_D$" and "w/o CI" produce ACC performance gains of 16.47% / 23.02%, 5.08% / 8.26%, 4.18% / 23.67%, 5.04% / 18.78%, and 4.13% / 6.93% across five datasets, indicating that either the dual non-contrastive clustering loss or the cluster-oriented imputation scheme plays an essential role in effectively handling attribute-missing deep graph clustering; 2) when compared to "w/o $\mathcal{L}_D$", AMGC achieves 4.53% - 21.54% ACC gains on five datasets. Similar observations can be obtained on other metrics. These results show that $\mathcal{L}_D$ effectively leverages the imputed information

Dataset	Metric	Base	w/o $\mathcal{L}_D$	w/o CI	AMGC
Cora	ACC	40.70	57.17	63.72	**66.65**
	NMI	27.11	40.97	44.57	**47.99**
	ARI	19.16	35.45	39.87	**43.40**
	F1	38.05	53.50	59.85	**61.02**
Citeseer	ACC	48.82	53.90	57.08	**60.92**
	NMI	24.83	28.40	29.57	**32.93**
	ARI	24.04	27.77	29.52	**33.73**
	F1	44.82	50.00	53.86	**57.33**
Pubmed	ACC	38.84	43.02	62.51	**64.56**
	NMI	1.19	6.72	22.48	**24.58**
	ARI	1.45	4.39	21.87	**24.19**
	F1	34.23	42.18	62.63	**64.52**
Co.CS	ACC	51.40	56.44	70.18	**72.61**
	NMI	56.49	60.27	71.10	**73.91**
	ARI	44.74	47.17	62.60	**64.62**
	F1	42.43	48.40	65.29	**68.34**
Co.Physics	ACC	69.02	73.15	75.95	**77.68**
	NMI	51.89	58.81	60.41	**62.77**
	ARI	64.81	67.50	66.26	**69.00**
	F1	43.64	49.95	65.03	**67.58**

Table 4: Ablation study for the CITE scheme. "w/o $\mathcal{L}_D$" and "w/o CI" denote two AMGC variants with the dual non-contrastive clustering loss and the cluster-oriented imputation scheme being removed, respectively. "Base" denotes AMGC without both components.

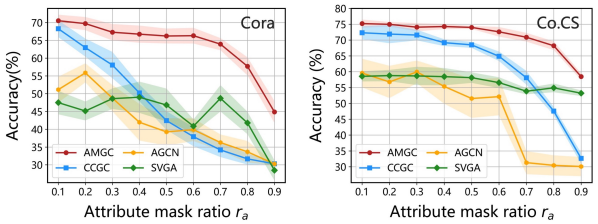

Figure 3: Performance comparison among four methods with different attribute-missing ratios.

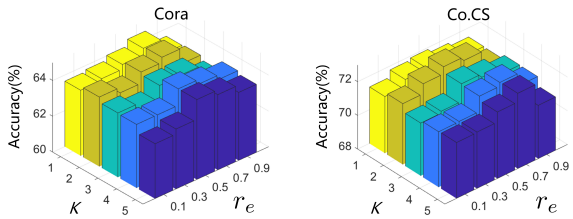

Figure 4: Model sensitivity with the variation of two hyper-parameters. The X-axis, Y-axis, and Z-axis refer to the K value, r_e value, and accuracy performance, respectively.

to promote greater graph encoding capability for clustering sub-network; and 3) AMGC consistently outperforms "w/o CI" on all datasets, demonstrating the effectiveness of taking the learned clustering signals as imputation prompts to help each attribute-missing sample gather highly correlated features within its clusters.

Less Attribute-Complete Samples

To further investigate the superiority of AMGC, it is necessary to show whether the proposed method can still achieve effective clustering performance when less visible attributes are available. To this end, we compare AMGC with AGCN, CCGC, and SVGA by varying the attribute-missing ratio r_a from 10% to 90%. From the results in Fig. 3, we can observe that 1) AMGC consistently performs better than compared baseline methods in all situations on Cora and Co.CS. For example, AMGC outperforms AGCN by a large margin in terms of ACC when r_a varies from 10% to 90% on Cora. The observations on Co.CS are similar. These improvements indicate that AMGC has stronger robustness against severe data absence; and 2) taking the results on Co.CS for example, AMGC with 80% missing attributes can still perform better than AGCN and SVGA with 10% missing attributes. These results indicate that AMGC can achieve high-quality data imputation as well as promising clustering performance with extremely limited observed signals.

Hyper-Parameter Analysis

We investigate model sensitivity analysis on the number of nearest neighbors K and the edge-masking ratio r_e. Firstly, we evaluate AMGC by varying K from 1 to 5 with a step size of 1. From the results in Fig. 4, we can see that on Co.CS, the performance shows a trend of first rising and then dropping slightly with the variation of K. This implies that AMGC needs a proper K to collect and preserve the most reliable information during the data imputation process. In addition, the optimal K value falls within the range of [1, 3] on both datasets. This indicates that selecting a relatively small K is reasonable. Secondly, we measure how the performance is affected by varying r_e from 0.1 to 0.9 with a step size of 0.2. As shown in Fig. 4, we get the best results when r_e reaches 0.5 or 0.7 on Cora and Co.CS. This indicates that the edge-masking operation is indeed effective for AMGC, but a proper r_e is required to balance the visible and masked edge information. Based on the above observations, we set the values of K and r_e to 2 and 0.7, respectively.

Conclusion

In this paper, we investigate a practical yet challenging problem, *i.e.,* deep graph clustering (DGC) on attribute-missing graphs. To solve this issue, we propose a novel DGC framework termed AMGC, where the clustering and imputation processes negotiate with each other through two-step alternate optimization. In our method, the proposed cluster-oriented imputation scheme and dual non-contrastive clustering loss can help the model learn denser cluster structures and well-separated clusters for better clustering. Extensive experiments on five datasets have verified that AMGC can effectively solve the problem of attribute-missing DGC and outperform competitors by a large margin. Future work may leverage the mutual information (MI) to explore the relationship between visible and missing samples, and extend the proposed AMGC to an imputation-free version.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62325604, 62276271, 62006237, 62002170, 62162024, 62162022), the Postgraduate Scientific Research Innovation Project in Hunan Province (No. CX20220076), and the Key Research and Development Program of Hainan Province (No. ZDYF2023GXJS163). Corresponding authors: Sihang Zhou and Xinwang Liu.

References

Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; and Cui, P. 2020. Structural Deep Clustering Network. In *Proceedings of the International World Wide Web Conference*, 1400–1410.

Chen, X.; Chen, S.; Yao, J.; Zheng, H.; Zhang, Y.; and Tsang, I. W. 2022. Learning on Attribute-Missing Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2): 740–757.

Cui, G.; Zhou, J.; Yang, C.; and Liu, Z. 2020. Adaptive Graph Encoder for Attributed Graph Embedding. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 976–985.

Cui, H.; Lu, Z.; Li, P.; and Yang, C. 2022. On Positional and Structural Node Features for Graph Neural Networks on Non-attributed Graphs. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 3898–3902.

der Maaten, L. V.; and Hinton, G. 2008. Visualizing Data Using t-SNE. *Journal of Machine Learning Research*, 9(11).

Gong, L.; Tu, W.; Zhou, S.; Zhao, L.; Liu, Z.; and Liu, X. 2022a. Deep Fusion Clustering Network With Reliable Structure Preservation. *IEEE Transactions on Neural Networks and Learning Systems*.

Gong, L.; Zhou, S.; Tu, W.; and Liu, X. 2022b. Attributed Graph Clustering with Dual Redundancy Reduction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3015–3021.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *Proceedings of the Conference on Neural Information Processing Systems*, 21271–21284.

Hartigan, J. A.; and Wong, M. A. 1979. A K-Means Clustering Algorithm. *Applied Stats*, 28(1): 100–108.

He, D.; Liang, C.; Huo, C.; Feng, Z.; Jin, D.; Yang, L.; and Zhang, W. 2022. Analyzing Heterogeneous Networks with Missing Attributes by Unsupervised Contrastive Learning. *IEEE Transactions on Neural Networks and learning systems*.

Hu, D.; Liang, K.; Zhou, S.; Tu, W.; Liu, M.; and Liu, X. 2023. scDFC: A Deep Fusion Clustering Method for Single-cell RNA-seq Data. *Briefings in Bioinformatics*, 24(4).

Huang, Z.; Chen, J.; Zhang, J.; and Shan, H. 2023. Learning Representation for Clustering Via Prototype Scattering and Positive Sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6): 7509–7524.

Jin, D.; Huo, C.; Liang, C.; and Yang, L. 2021. Heterogeneous Graph Neural Network via Attribute Completion. In *Proceedings of the International World Wide Web Conference*, 391–400.

Jin, D.; Wang, R.; Wang, T.; He, D.; Ding, W.; Huang, Y.; Wang, L.; and Pedrycz, W. 2023. Amer: A New Attribute-Missing Network Embedding Approach. *IEEE Transactions on Cybernetics*, 57(3): 4306–4319.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations*.

Li, J.; Wu, R.; Sun, W.; Chen, L.; Tian, S.; Zhu, L.; Meng, C.; Zheng, Z.; and Wang, W. 2023a. What's Behind the Mask: Understanding Masked Graph Modeling for Graph Autoencoders. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1268–1279.

Li, J.; Zhou, P.; Xiong, C.; and Hoi, S. C. H. 2021. Prototypical Contrastive Learning of Unsupervised Representations. In *Proceedings of the International Conference on Learning Representations*.

Li, L.; Wang, S.; Liu, X.; Zhu, E.; Shen, L.; Li, K.; and Li, K. 2022a. Local Sample-Weighted Multiple Kernel Clustering With Consensus Discriminative Graph. *IEEE Transactions on Neural Networks and Learning Systems*.

Li, L.; Zhang, J.; Wang, S.; Liu, X.; Li, K.; and Li, K. 2023b. Multi-View Bipartite Graph Clustering With Coupled Noisy Feature Filter. *IEEE Transactions on Knowledge and Data Engineering*, 35(12): 12842–12854.

Li, M.; Zhang, Y.; Zhang, W.; Zhao, S.; Piao, X.; and Yin, B. 2022b. Csat: Contrastive Sampling-Aggregating Transformer for Attribute-Missing Graph Learning. *Social Science Research Network*.

Liang, K.; Liu, Y.; Zhou, S.; Tu, W.; Wen, Y.; Yang, X.; Dong, X.; and Liu, X. 2023a. Knowledge Graph Contrastive Learning Based on Relation-Symmetrical Structure. *IEEE Transactions on Knowledge and Data Engineering*, 36(1): 226–238.

Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; and Liu, X. 2023b. Learn from Relational Correlations and Periodic Events for Temporal Knowledge Graph Reasoning. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1559–1568.

Liu, X. 2021. Incomplete Multiple Kernel Alignment Maximization for Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, X.; Li, M.; Tang, C.; Xia, J.; Xiong, J.; Liu, L.; Kloft, M.; and Zhu, E. 2021. Efficient and Effective Regularized Incomplete Multi-View Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8): 2634–2646.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Zhu, E.; Liu, T.; Kloft, M.; Shen, D.; Yin, J.; and Gao, W. 2020. Multiple Kernel K-means with Incomplete Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(5): 1191–1204.

Liu, Y.; Tu, W.; Zhou, S.; Liu, X.; Song, L.; Yang, X.; and Zhu, E. 2022. Deep Graph Clustering via Dual Correlation Reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7603–7611.

Liu, Y.; Yang, X.; Zhou, S.; Liu, X.; Wang, Z.; Liang, K.; Tu, W.; Li, L.; Duan, J.; and Chen, C. 2023. Hard Sample Aware Network for Contrastive Deep Graph Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8914–8922.

Pan, S.; Hu, R.; Fung, S.-F.; Long, G.; Jiang, J.; and Zhang, C. 2020. Learning Graph Embedding with Adversarial Training Methods. *IEEE Transactions on Cybernetics*, 50(6): 2475–2487.

Park, N.; Rossi, R.; Koh, E.; Burhanuddin, I. A.; Kim, S.; Du, F.; Ahmed, N.; and Faloutsos, C. 2022. Cgc: Contrastive Graph Clustering for Community Detection And Tracking. In *Proceedings of the International World Wide Web Conference*, 1115–1126.

Peng, X.; Cheng, J.; Tang, X.; Zhang, B.; and Tu, W. 2024. Multi-view Graph Imputation Network. *Information Fusion*.

Peng, Z.; Liu, H.; Jia, Y.; and Hou, J. 2021. Attention-driven Graph Clustering Network. In *Proceedings of the ACM International Conference on Multimedia*, 935–943.

Rossi, E.; Kenlay, H.; Gorinova, M. I.; Chamberlain, B. P.; Dong, X.; and Bronstein, M. M. 2022. On the Unreasonable Effectiveness of Feature Propagation in Learning on Graphs With Missing Node Features. In *Proceedings of the First Learning on Graphs Conference*.

Tao, Z.; Liu, H.; Li, J.; Wang, Z.; and Fu, Y. 2019. Adversarial Graph Embedding for Ensemble Clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3562–3568.

Tu, W.; Liao, Q.; Zhou, S.; Peng, X.; Ma, C.; Liu, Z.; Liu, X.; and Cai, Z. 2023a. RARE: Robust Masked Graph Autoencoder. *IEEE Transactions on Knowledge and Data Engineering*.

Tu, W.; Zhou, S.; Liu, X.; Ge, C.; Cai, Z.; and Liu, Y. 2023b. Hierarchically Contrastive Hard Sample Mining for Graph Self-supervised Pre-training. *IEEE Transactions on Neural Networks and Learning Systems*.

Tu, W.; Zhou, S.; Liu, X.; Guo, X.; Cai, Z.; Zhu, E.; and Cheng, J. 2021. Deep Fusion Clustering Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 9978–9987.

Tu, W.; Zhou, S.; Liu, X.; Liu, Y.; Cai, Z.; Zhu, E.; Zhang, C.; and Cheng, J. 2022. Initializing Then Refining: A Simple Graph Attribute Imputation Network. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3494–3500.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations*.

Wan, X.; Liu, J.; Liang, W.; Liu, X.; Wen, Y.; and Zhu, E. 2022. Continual Multi-View Clustering. In *Proceedings of the ACM International Conference on Multimedia*, 3676–3684.

Wan, X.; Liu, X.; Liu, J.; Wang, S.; Wen, Y.; Liang, W.; Zhu, E.; Liu, Z.; and Zhou, L. 2023. Auto-Weighted Multi-View Clustering for Large-Scale Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 10078–10086.

Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; and Zhang, C. 2019. Attributed Graph Clustering: A Deep Attentional Embedding Approach. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3670–3676.

Wang, S.; Liu, X.; Liu, L.; Tu, W.; Zhu, X.; Liu, J.; Zhou, S.; and Zhu, E. 2022a. Highly-efficient Incomplete Large-scale Multi-view Clustering with Consensus Bipartite Graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9776–9785.

Wang, Y.; Zhang, Y.; Zhang, F.; Wang, S.; Lin, M.; Zhang, Y.; and Sun, X. 2022b. Ada-NETS: Face Clustering via Adaptive Neighbour Discovery in The Structure Space. In *Proceedings of the International Conference on Learning Representations*.

Wu, L.; Lin, H.; Tan, C.; Gao, Z.; and Li, S. Z. 2023. Self-supervised Learning on Graphs: Contrastive, Generative, or Predictive. *IEEE TKDE*, 35(4): 4216–4235.

Xia, D.; Yang, Y.; Yang, S.; and Li, T. 2023. Incomplete Multi-view Clustering via Kernelized Graph Learning. *Information Sciences*, 625: 1–19.

Xia, W.; Wang, Q.; Gao, Q.; Yang, M.; and Gao, X. 2022. Self-consistent Contrastive Attributed Graph Clustering with Pseudo-label Prompt. *IEEE Transactions on Multimedia*, 25: 6665–6677.

Xu, J.; Li, C.; Ren, Y.; Peng, L.; Mo, Y.; Shi, X.; and Zhu, X. 2022. Deep Incomplete Multi-view Clustering via Mining Cluster Complementarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8761–8769.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *Proceedings of the International Conference on Learning Representations*.

Xue, H.; Mallawaarachchi, V.; Zhang, Y.; Rajan, V.; and Lin, Y. 2022. RepBin: Constraint-based Graph Representation Learning for Metagenomic Binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4637–4645.

Yang, X.; Liu, Y.; Zhou, S.; Wang, S.; Tu, W.; Zheng, Q.; Liu, X.; Fang, L.; and Zhu, E. 2023. Cluster-guided Contrastive Graph Clustering Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 10834–10842.

Yang, Y.; Guan, Z.; Wang, Z.; Zhao, W.; Xu, C.; Lu, W.; and Huang, J. 2022. Self-supervised Heterogeneous Graph Pretraining Based on Structural Clustering. In *Proceedings of the Conference on Neural Information Processing Systems*, 16962–16974.

Yoo, J.; Jeon, H.; Jung, J.; and Kang, U. 2022. Accurate Node Feature Estimation with Structured Variational Graph Autoencoder. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2336–2346.

Zhao, H.; Yang, X.; Wang, Z.; Yang, E.; and Deng, C. 2021. Graph Debiased Contrastive Learning with Joint Representation Clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3434–3440.