Diffusion-based Graph-agnostic Clustering

Kun Xie
The Chinese University of Hong Kong
Hong Kong
xiekun@se.cuhk.edu.hk

Renchi Yang*
Hong Kong Baptist University
Hong Kong
renchi@hkbu.edu.hk

Sibo Wang
The Chinese University of Hong Kong
Hong Kong
swang@se.cuhk.edu.hk

Abstract

Clustering over a graph seeks to partition the nodes therein into disjoint groups such that nodes within the same cluster are tightly-knit, while those across clusters are distant from each other. In practice, graphs are often attended with rich attributes, which are termed *attributed graphs*. By leveraging the complementary nature of graph topology and node attributes in such graphs, *graph neural networks* (GNNs) have obtained encouraging performance in graph clustering. However, existing GNN-based approaches strongly rely on the *homophilic* assumption of the input graph, and thus, largely fail on *heterophilic* graphs and others embodying numerous missing or noisy links, which are widely present in real life.

To bridge this gap, this paper presents DGAC, an effective graph-agnostic solution for graph clustering. Particularly, DGAC overcomes the limitations of prior works by exploiting the high-order connectivity of nodes within not only the input graph $\mathcal{G}$ but also the affinity graph $\mathcal{H}$ underlying the attribute data. To achieve this goal, we first unify the embedding and clustering generations into a coherent framework optimizing the *Dirichlet Energy* on both $\mathcal{G}$ and $\mathcal{H}$. Based thereon, theoretically-grounded solvers are developed for efficient constructions of the embeddings and clusters via graph diffusion operations, which aggregate features from specific neighbors, enabling the capture of high-order semantics from $\mathcal{G}$ or $\mathcal{H}$. On top of that, DGAC includes three training loss functions that facilitate effective feature extraction and clustering. Extensive experiments, comparing DGAC against 12 baselines over 12 homophilic or heterophilic graph datasets, showcase that DGAC consistently and considerably outperforms all competitors in terms of clustering quality measured against ground truth labels.

CCS Concepts

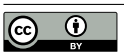
• **Information systems** → **Clustering**.

Keywords

Graph clustering, Dirichlet Energy, affinity graph, graph neural network, heterophily graph clustering, contrastive learning

ACM Reference Format:
Kun Xie, Renchi Yang, and Sibo Wang. 2025. Diffusion-based Graph-agnostic Clustering. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3696410.3714652

*Renchi Yang is the corresponding author.



WWW '25, April 28-May 2, 2025, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1274-6/25/04
https://doi.org/10.1145/3696410.3714652

1 Introduction

As a fundamental task in data mining, graph clustering has consistently garnered significant attention from researchers due to its broad applicability across various domains such as community detection [47, 73], bioinformatics [8, 13], anomaly identification [48, 60], recommender systems [22, 42] and so on.

In recent years, *graph neural networks* (GNNs) [40] have emerged as popular techniques for graph clustering by virtue of their powerful capabilities in fusing complex structural connectivity and nodal attributes [52] that are widely present in real graphs. Specifically, this methodology follows a two-step workflow, where the first step generates informative feature vectors for nodes in the input graph via GNNs, while the latter focuses on transforming node features into clustering results using the K-Means, spectral clustering, or MLP. The design recipes for GNNs are built on the *graph diffusion* [25, 41] operations, which aggregate features from direct or indirect neighboring nodes based on high-order connectivity.

As pinpointed by our theoretical investigation, GNNs essentially minimize the *Dirichlet Energy* (DE) [30] of the node representations over the input graph topology, which enforces node features align with the graph connectivity and comply with the homophily assumption [62]. That is, nodes that are adjacent or strongly connected should fall into the same classes or clusters. Intuitively, over graphs with high *homophily ratio* (HR) [103], i.e., the fraction of edges whose endpoints are in the same ground-truth class/cluster, optimizing the DE-based objectives leads to desired node features and clustering results. Nevertheless, this homophily assumption is not necessarily valid in practice. Real-world graph datasets are often *heterophilic* (i.e., the homophily ratio is low), noisy, or incomplete. As an aftermath, the majority of existing GNNs largely fail on such datasets.

Despite a series of heterophilic GNN models [57, 58, 101] proposed recently, they are primarily designed for node classification under supervised settings, and thus, cannot be readily applied for graph clustering due to the absence of node labels. To our knowledge, a paucity of efforts [26, 66] has been invested towards addressing heterophily issues for graph clustering. Similar in spirit to previous heterophilic GNNs for node classification, these models focus on increasing the homophily ratio of the input graph by rewiring its topology with node attributes as ancillary information. To ascertain the correlation between graph topology and node attributes, Table 1 reports the accuracy obtained by clustering the adjacency matrix $\mathbf{A}$ and attribute matrix $\mathbf{X}$ of real heterophilic graph datasets (See Table 3) with K-Means severally, as well as the degree of overlap between these two independent clustering results. It can be observed that the overlap degree is conspicuously lower than the clustering accuracy on $\mathbf{X}$, indicating that node attributes embody a wealth of *unique* information that is crucial for accurate clustering. However, existing works merely leverage it to refine the original

Table 1: Overlap of clustering results on real graphs.

Dataset	Acc. ($\mathbf{X}$)	Acc. ($\mathbf{A}$)	Overlap
Texas	0.49	0.54	0.40
Cornell	0.49	0.51	0.41
Flickr	0.34	0.20	0.23

graph, without considering the inherent structures underlying $\mathbf{X}$, and thus, compromise the result quality.

In response, this paper presents DGAC, a Diffusion-based Graph-Agnostic Clustering method, which achieves high clustering effectiveness over both homophilic and heterophilic graphs. At a high level, DGAC overcomes the deficiencies of prior works by capturing the high-order node connectivity over the input graph $\mathcal{G}$ and the affinity graph $\mathcal{H}$ underlying the attribute data simultaneously.

More concretely, DGAC first integrates the classic GNNs and spectral clustering into a unified framework that minimizes the DE of embeddings and clustering results on the input graph through rigorous theoretical analyses. Guided by this principle, we first propose to jointly optimize the DE-based objectives over $\mathcal{G}$ and $\mathcal{H}$ for representation learning, which leads to the design of our *dual graph diffusion networks* (DGDNs). Distinct from GNNs, DGDNs incorporate the high-order node connectivity on two graphs into the node features through topology- and attribute-based graph diffusion, followed by a feature fusion reconciling information from two data modalities i.e., structures and attributes. On top of that, we formulate another DE-based joint optimization problem upon the learned node features and input graph topology and develop a graph diffusion-based solver for efficient clustering. Furthermore, a hierarchical contrastive loss is proposed based on representations from two optimization perspectives across three levels, aiming to preserve the invariant and inherent properties of the input information. Along with an infoNCE loss for clustering quality assessment and a reconstruction loss to ensure the preservation of the input graph structures and attributes, DGAC are effectively trained. Our extensive experimental studies, comparing DGAC against 12 baselines over 12 homophilic or heterophilic graph datasets, demonstrate that DGAC is consistently superior to all baselines in terms of clustering accuracy evaluated against the ground-truth cluster labels. In particular, on the largest homophilic graph BlogCatalog, and heterophilic graph Flickr, DGAC is able to obtain a substantial gain of 18.59% and 11.12% in clustering accuracy, respectively.

2 Related Work

Heterophilic Graph Clustering. Dealing with heterophilic graphs in unsupervised settings is particularly challenging due to the absence of node label guidance, as noted in previous work [11, 66, 102]. Existing heterophilic graph clustering methods often utilize attribute information to refine the graph structure. Specifically, DGCN [66] and PFGC [87] leverage the attribute (dis)similarity between connected nodes to construct two graphs that are highly homophilic and heterophilic, respectively. These graphs are then processed using low-pass and high-pass filters to capture holistic information. HoLe [26] aims to enhance the homophily degree by refining the graph topology based on high-confidence clustering results. Additionally, HGRL [11] is a self-supervised learning method that learns informative representations for heterophilic graphs through two pretext tasks: original feature preserving and generalized neighbor capturing. These existing methods emphasize explicit discrimination between homophilic and heterophilic structures, yet they overlook the inherent affinity relationships underlying the attribute information that facilitates clustering.

A detailed introduction to existing *attributed graph clustering methods* and *heterophilic graph neural networks* can be found in Appendix A, most of which rely on homophily assumption or node labels for supervision, limiting their applicability in heterophilic graph clustering.

3 Preliminaries

3.1 Notations and Terminology

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed graph, where $\mathcal{V}$ is a set of n nodes and $\mathcal{E}$ is a set of m edges. For each edge $(v_i, v_i) \in \mathcal{E}$, we say v_i and v_j are neighbors to each other, and we use $\mathcal{N}(v_i)$ to denote the set of neighbors of v_i, with the degree $d(v_i) = |\mathcal{N}(v_i)|$. $\mathbf{X} \in \mathbb{R}^{n \times f}$ is the input attribute matrix of nodes, where each row $\mathbf{X}_i$ stands for the attributes associated with node v_i and satisfies $\|\mathbf{X}_i\|_2 = 1$. We use $\mathbf{A}$ to symbolize the adjacency matrix of $\mathcal{G}$, where $\mathbf{A}_{i,j} = 1$ if there is an edge $(v_i, v_j) \in \mathcal{E}$, and otherwise $\mathbf{A}_{i,j} = 0$, and $\mathbf{D}$ to denote the degree matrix of $\mathcal{G}$. Accordingly, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ represents the Laplacian matrix of $\mathcal{G}$. $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ and $\mathbf{I} - \tilde{\mathbf{A}}$ are the normalized adjacency and Laplacian matrices of $\mathcal{G}$, respectively.

Graph Diffusion. Given a graph $\mathcal{G}$ and an initial mass x at seed node $v_s \in \mathcal{V}$, the *graph diffusion*[1] [25, 41] is to spread the mass from v_s to its direct or indirect neighboring nodes along the edges in $\mathcal{G}$. The resulting distribution of mass at nodes is referred to as a graph diffusion distribution, wherein the value at each node v_i reflects the strength of connections between v_s and v_i. In mathematical terms, this process can be expressed by

$$\mathbf{p} = \mathbf{x} \cdot \sum_{\ell=0}^{\infty} w_\ell \mathbf{P}^\ell, \text{ where } \sum_{\ell=0}^{\infty} w_\ell = 1, \tag{1}$$

$\mathbf{x}$ has a positive value at s-th entry and 0 everywhere else, w_ℓ signifies the fraction of mass disseminated to all the nodes at ℓ-hops away from seed node v_s. $\mathbf{P}$ stands for the diffusion matrix of $\mathcal{G}$, which can be the transition matrix $\mathbf{D}^{-1}\mathbf{A}$ or normalized adjacency matrix $\hat{\mathbf{A}}$. Particularly, when $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ and $\mathbf{x}_s = 1$, the graph diffusion distribution in Eq. (1) is essentially the prominent *personalized PageRank* [32, 33, 35, 80–83] if w_ℓ follows a geometric distribution and the *heat kernel PageRank* [41] if w_ℓ is drawn from a Poisson distribution. These two variants have been extensively studied and successfully applied to various graph-related tasks [20, 86, 98, 99].

Graph Clustering. Given an attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ and the number K of clusters, the overreaching goal of graph clustering is to partition the node set $\mathcal{V}$ into K disjoint groups $\{C_1, \ldots, C_K\}$ (i.e., $\bigcup_{k=1}^{K} C_k = \mathcal{V}$ and $C_i \cap C_j = \emptyset$ for $i \neq j$), such that nodes with strong connectivity and high attribute homogeneity are in the same group, while distant and dissimilar ones fall into distinct clusters. As in common practice, the clustering result can be represented by a *Normalized Cluster Indicator* (NCI) $\mathbf{Y} \in \mathbb{R}^{n \times K}$ defined as follows:

$$\mathbf{C}_{i,j} = \begin{cases} \frac{1}{\sqrt{|C_j|}}, & \text{if } v_i \in C_j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

[1] This is different from the generative diffusion models [46].

In particular, the columns of $\mathbf{C}$ are orthonormal, i.e., $\mathbf{C}^\top\mathbf{C} = \mathbf{I}$.

3.2 Spectral Graph Clustering

Spectral clustering [78] is a canonical technique for graph clustering, which seeks to partition nodes towards minimizing intra-cluster connectivity. One standard formulation of such objectives is the RatioCut [27]: $\min_{\{C_1,\ldots,C_K\}} \sum_{k=1}^{K} \frac{1}{K} \sum_{v_i\in C_k, v_j\in\mathcal{V}\setminus C_k} \frac{\tilde{\mathbf{A}}_{i,j}}{|C_k|}$, which is to minimize the average weight of edges connecting nodes in any two distinct clusters. As analysed in [78], the above objective is equivalent to finding an NCI optimizing the following trace minimization problem:

$$\min_{\mathbf{C}} \operatorname{trace}(\mathbf{C}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{C}), \tag{3}$$

which is an NP-hard problem given the constraint in Eq. (2) on $\mathbf{C}$. A common way is to compute an approximate solution by relaxing the discreteness condition on $\mathbf{C}$ and allowing it to take arbitrary values in $\mathbb{R}$ such that the column-orthonormal property, i.e., $\mathbf{C}^\top\mathbf{C} = \mathbf{I}$, still holds. By Ky Fan's trace maximization principle [23], it immediately leads to that the optimal solution is the k-largest eigenvectors of $\tilde{\mathbf{A}}$. The K-Means or rounding algorithms [91, 96] are then applied to convert the k-largest eigenvectors into an NCI.

3.3 Graph Neural Networks

As demystified in recent works [61, 90, 104], after removing the non-linearity, a majority of existing popular GNNs [12, 24, 25, 40, 84, 88] can be unified into the *graph Laplacian smoothing* [18] problem formulated below:

$$\min_{\mathbf{H}} (1-\alpha)\cdot\|\mathbf{H}-\mathbf{X}\|_F^2 + \alpha\cdot\operatorname{trace}(\mathbf{H}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{H}), \tag{4}$$

where $\mathbf{H}$ denotes the target node features and α stands for a coefficient striking a balance between two terms. The first term reduces the discrepancy between the initial node features $\mathbf{X}$ and its smoothed version $\mathbf{H}$, while the latter can be rewritten as $\alpha\cdot\sum_{(v_i,v_i)\in\mathcal{E}}\|\mathbf{H}_i/\sqrt{d(v_i)}-\mathbf{H}_j/\sqrt{d(v_j)}\|_2^2$, meaning that the node features of adjacent nodes are enforced to be similar.

Lemma 1 ([31]). *Let $\mathbf{M}$ be a matrix whose dominant eigenvalue λ satisfies $|\lambda|<1$. Then, $\mathbf{I}-\mathbf{M}$ is invertible, and its inverse $(\mathbf{I}-\mathbf{M})^{-1}$ can be expanded as a Neumann series: $(\mathbf{I}-\mathbf{M})^{-1} = \sum_{\ell=0}^{\infty}\mathbf{M}^\ell$.*

By setting the derivative of Eq. (4) w.r.t. $\mathbf{H}$ to zero, the optimal $\mathbf{H}$ can be given by $\mathbf{H} = (1-\alpha)\cdot(\mathbf{I}-\alpha\tilde{\mathbf{A}})^{-1}\mathbf{X}$ [90]. Given the fact that the eigenvalues of $\tilde{\mathbf{A}}$ lie in $[-1,1]$ [16], Lemma 1 implies that $\mathbf{H} = \sum_{\ell=0}^{\infty}(1-\alpha)\alpha^\ell\tilde{\mathbf{A}}^\ell\cdot\mathbf{X}$. For each i-th column $\mathbf{h}$ of $\mathbf{H}$, we can rewrite it as follows with the i-th column $\mathbf{x}$ of $\mathbf{X}$:

$$\mathbf{h}^\top = \mathbf{x}^\top\cdot\sum_{\ell=0}^{\infty}(1-\alpha)\alpha^\ell\tilde{\mathbf{A}}^\ell.$$

This is essentially a graph diffusion distribution with the initial mass for all nodes specified in $\mathbf{z}$. In other words, the columns of $\mathbf{H}$ correspond to the graph diffusion distributions of d attributes of nodes over $\mathcal{G}$. Aside from the above graph diffusion that is adopted in [24], other diffusion forms include $(\mathbf{I}+\tilde{\mathbf{A}})\mathbf{Z}$ in GCN [40], $\tilde{\mathbf{A}}^\ell\mathbf{X}$ in [84], $\sum_{\ell=0}^{\infty}\frac{e^{-t}\cdot t^i}{\ell!}\tilde{\mathbf{A}}^\ell\cdot\mathbf{Z}$ in [25], etc. Note that in these models, the node features $\mathbf{H}$ will further be transformed into the final node representations through layer-wise feature transformations, e.g., MLPs, linear layers. Most existing GNN-based graph clustering approaches convert node representations $\mathbf{H}$ into the NCI $\mathbf{C}$ through the K-Means [54] or MLP/linear layers [75].

Table 2: Homophily ratios of $\mathcal{G}$ and $\mathcal{H}$.

	Texas	Wisconsin	Cornell	Flickr
$\mathrm{HR}_{\mathcal{G}}$	0.108	0.196	0.305	0.239
$\mathrm{HR}_{\mathcal{H}}$	0.422↑	0.399↑	0.422↑	0.413↑

4 Solution Overview

This section first unifies spectral clustering and GNNs into a framework minimizing the *Dirichlet Energy* (DE) through in-depth theoretical analyses, followed by delineating the basic idea of DGAC.

4.1 Dirichlet Energy Minimization Framework

Definition 1 (Dirichlet Energy [30]). *Given $\mathcal{G}$ and a graph signal $\mathbf{x}\in\mathbb{R}^n$, the Dirichlet Energy of $\mathbf{x}$ over $\mathcal{G}$ is characterized by $\mathcal{D}(\mathbf{x},\mathbf{A}) = \frac{1}{2}\sum_{v_i,v_j\in\mathcal{V}}\mathbf{A}_{i,j}\cdot(\mathbf{x}_i-\mathbf{x}_j)^2 = \mathbf{x}^\top\mathbf{L}\mathbf{x}$.*

Definition 1 presents the formal definition of the DE, which intrinsically measures the locality-preserving power of a given signal $\mathbf{x}$, i.e., its consistency of the distribution of signal values at nodes within the input graph structure [89].

Let $\mathbf{Y}\in\mathbb{R}^{n\times K}$ be the ground-truth node-cluster membership matrix As shown in Lemma 2[2], we establish a theoretical relation between the homophily ratio $\mathrm{HR}_{\mathcal{G}}$ of $\mathcal{G}$ and the overall DE of $\mathbf{Y}$ over $\mathcal{G}$. Intuitively, the larger the homophily ratio of $\mathcal{G}$ is, the smaller the DE of $\mathbf{Y}$ is.

Lemma 2. $HR_{\mathcal{G}} = \frac{1}{2} - \frac{1}{|\mathcal{E}|}\sum_{k=1}^{K}\mathcal{D}(\mathbf{Y}_{\cdot,k},\mathbf{A})$.

Further, based on the definition of matrix trace, we can derive the following equivalence between DE and the objectives of spectral clustering and GNNs (i.e., Eq. (3) and Eq. (4)):

$$\begin{aligned}\operatorname{trace}(\mathbf{C}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{C}) &= \sum_{k=1}^{K}\mathcal{D}(\mathbf{C}_{\cdot,k},\tilde{\mathbf{A}}),\\ \operatorname{trace}(\mathbf{H}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{H}) &= \sum_{i=1}^{d}\mathcal{D}(\mathbf{H}_{\cdot,i},\tilde{\mathbf{A}}).\end{aligned} \tag{5}$$

This finding reveals that *the fundamental principles of both spectral clustering and GNNs are built on minimizing the overall DE of their results over the input graph $\mathcal{G}$*. According to Lemma 2, when the homophily ratio of $\mathcal{G}$ is high, doing so essentially renders the NCI $\mathbf{C}$ and node representations $\mathbf{H}$ yield clustering results close to the ground-truth $\mathbf{Y}$. As pinpointed in Section 3.3, GNNs leverage the graph diffusion to incorporate high-order connectivity between nodes into the construction of $\mathbf{H}$, thereby reinforcing the optimization of its DE.

4.2 High-Level Idea

As reported in Table 2, the homophily ratios of practical graph datasets are often low, due to their heterophilic nature or the presence of noisy and missing links. Naturally, on such graphs, optimizing the above-said DE-based objectives leads to undermined embedding and clustering quality. By contrast, in Table 2, their

[2]All proofs can be found in our technical report [1].

corresponding affinity graphs $\mathcal{H}$, that are constructed from their associated attribute matrices $\mathbf{X}$ with edge weights calculated via

$$\mathbf{S}_{ij} = \mathbf{X}_i \cdot \mathbf{X}_j^\top \ \forall v_i, v_j \in \mathcal{V}, \tag{6}$$

exihibit much higher degrees of homophily.

This motivates our rudimentary idea of capitalizing on the inherent structures of the attribute data, i.e., the affinity graph underlying $\mathbf{X}$, as complementary signals for the computations of $\mathbf{H}$ and $\mathbf{C}$. That is, we additionally minimize the overall DE of $\mathbf{H}$ over $\mathcal{H}$:

$$\sum_{i=1}^{d} \mathcal{D}(\mathbf{H}_{\cdot,i}, \tilde{\mathbf{S}}), \tag{7}$$

where $\tilde{\mathbf{S}}$ stands for the symmetrically normalized version of $\mathbf{S}$, i.e., $\mathrm{diag}(\sum_j \mathbf{S}_{:,j})^{-1/2} \cdot \mathbf{S} \cdot \mathrm{diag}(\sum_j \mathbf{S}_{:,j})^{-1/2}$. Ideally, $\mathbf{H}$ should capture the high-order node connectivity on both $\mathcal{G}$ and $\mathcal{H}$.

In the same vein, the NCI $\mathbf{C}$ can be derived from $\mathbf{H}$ and $\mathcal{G}$ based on minimizing the overall DE as follows:

$$\min_{\mathbf{C}^\top\mathbf{C}} \sum_{k=1}^{K} (1-\gamma) \cdot \mathcal{D}(\mathbf{C}_{\cdot,k}, \Delta) + \gamma \cdot \mathcal{D}(\mathbf{C}_{\cdot,k}, \tilde{\mathbf{A}}). \tag{8}$$

Δ corresponds to the affinity graph underlying the node features $\mathbf{H}$ and γ is a weight balancing two terms. The second term $\mathcal{D}(\mathbf{C}_{\cdot,k}, \tilde{\mathbf{A}})$ is to explicitly injects the graph connectivity, as $\mathbf{H}$ tends to lose such information as an aftermath of the inherent issues of GNNs, i.e., over-smoothing and over-squashing.

In turn, there remain two crucial technical issues to be addressed:

- How to construct node representations $\mathbf{H}$ by optimizing its DE over $\mathcal{G}$ and $\mathcal{H}$ simultaneously?
- How to derive the NCI $\mathbf{C}$ from $\mathbf{H}$ and $\mathcal{G}$ by solving Eq. (8)?

5 The DGAC Method

In this section, we present our graph-agnostic clustering solution DGAC for addressing the problems remarked in preceding section. The pipeline of DGAC is illustrated in Figure 1, which involves three major components. We begin with elucidating our *dual graph diffusion networks* (DGDN) designed for constructing node representations $\mathbf{H}$ in Sections 5.1. The succeeding section (Section 5.2) describes the algorithmic details of our *graph diffusion clustering* (GDC) for NCI computation. In Sections 5.3 and 5.4, we introduce the training objectives for the model and provide related theoretical analyses, respectively.

5.1 Dual Graph Diffusion Networks

Following our idea in Section 4.2, DGDN creates $\mathbf{H}$ by optimizing the following DE-based objectives simultaneously:

$$\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}} \sum_{i=1}^{d} \mathcal{D}(\mathbf{H}_{\cdot,i}, \tilde{\mathbf{A}}) + \mathcal{D}(\mathbf{H}_{\cdot,i}, \tilde{\mathbf{S}}) \tag{9}$$

The orthogonality constraint $\mathbf{H}^\top\mathbf{H} = \mathbf{I}$ is introduced to avoid trivial solution since it limits the feasible domain to a *Stiefel manifold* [50]. In doing so, we can additionally mitigate the feature correlation issue [37, 49]. As per Eq. (5), the above optimization objective can be rewritten as a joint trace minimization problem:

$$\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}} \mathrm{trace}(\mathbf{H}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{H}) + \mathrm{trace}(\mathbf{H}^\top(\mathbf{I}-\tilde{\mathbf{S}})\mathbf{H}).$$

As remarked earlier in Section 3.2, by Ky Fan's trace maximization principle [23], the optimal solutions to trace minimization problems $\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}} \mathrm{trace}(\mathbf{H}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{H})$ and $\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}} \mathrm{trace}(\mathbf{H}^\top(\mathbf{I}-\tilde{\mathbf{S}})\mathbf{H})$ are the d-largest eigenvectors $\mathbf{B}$ and $\mathbf{U}$ of $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{S}}$, respectively. As such, we can transform Eq. (9) into two optimization problems:

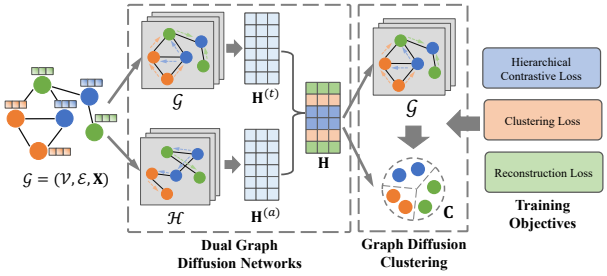


Figure 1: Overview of DGAC.

$$\min_{\mathbf{H}} \alpha \cdot \mathrm{trace}(\mathbf{H}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{H}) + \|\mathbf{H}-\mathbf{U}\|_F^2, \tag{10}$$

$$\min_{\mathbf{H}} \|\mathbf{H}-\mathbf{B}\|_F^2 + \alpha \cdot \mathrm{trace}(\mathbf{H}^\top(\mathbf{I}-\tilde{\mathbf{S}})\mathbf{H}), \tag{11}$$

each of which partially optimizes Eq. (9). On this basis, our idea is then seeking the solutions $\mathbf{H}^{(t)}$ and $\mathbf{H}^{(a)}$ to these two problems and coalesce them as the final node representations $\mathbf{H}$. Specifically, DGDN computes $\mathbf{H}^{(t)}$ and $\mathbf{H}^{(a)}$ via topology-based and attribute-based graph diffusion, respectively.

Topology-based Graph Diffusion. Firstly, we can see that the optimization problem in Eq. (10) is exactly a graph Laplacian smoothing mentioned in Section 3.3. Using Lemma 1, its solution with L_t iterations is a graph diffusion of $\mathbf{U}$ over $\mathcal{G}$:

$$\mathbf{H}^{(t)} = \sum_{\ell=0}^{L_t} \alpha^\ell \tilde{\mathbf{A}}^\ell \mathbf{U}, \tag{12}$$

where the columns of $\mathbf{U}$ are the d-largest eigenvectors of $\tilde{\mathbf{S}}$.

Lemma 3. *The columns of* $\mathbf{U}$ *are the top-d left singular vectors of* $\overline{\mathbf{X}}$, *where* $\overline{\mathbf{X}} = \mathrm{diag}(\mathbf{d})^{-1/2}\mathbf{X}$ *and* $\mathbf{d} = \mathbf{X} \cdot (\sum_{v_i \in \mathcal{V}} \mathbf{X}_i)^\top$.

Lemma 3 suggests that we can obtain $\mathbf{U}$ via a truncated *singular value decomposition* (SVD) of $\overline{\mathbf{X}}$, without direct eigen-decomposition of $\tilde{\mathbf{S}}$, and hence, eliminate the need of the explicit materialization of affinity matrix $\tilde{\mathbf{S}}$.

With the close connection of SVD to *principal component analysis* (PCA), $\mathbf{U}$ are actually the principal components, i.e., key features, from the normalized attribute matrix $\overline{\mathbf{X}}$. That is to say, our topology-based graph diffusion in Eq. (12) generates node features that incorporate the high-order connectivity of nodes on $\mathcal{G}$ and key features extracted from the attribute data $\mathbf{X}$. Given this interpretation, another practical way is to apply an MLP over $\mathbf{X}$ or $\mathbf{X}\mathbf{X}^\top$ to extract such key features as $\mathbf{U}$.

Lemma 4. *The dominant eigenvalue* λ *of* $\tilde{\mathbf{S}}$ *satisfies* $|\lambda| \leq 1$.

Attribute-based Graph Diffusion. Next, we solve another optimization problem in Eq. (11) to get $\mathbf{H}^{(a)}$, which can also be cast into a graph Laplacian smoothing problem in Eq. (4) with Lemma 1 and 4. Let $\mathbf{B}$ be the d-largest eigenvectors of the normalized adjacency matrix $\tilde{\mathbf{A}}$, which commonly serves as the structural embeddings of nodes in the literature [21, 74, 92, 100]. The node features $\mathbf{H}^{(a)}$ thus can be calculated by graph diffusion of $\mathbf{B}$ over the affinity graph:

$$\mathbf{H}^{(a)} = \sum_{\ell=0}^{L_a} \alpha^\ell \tilde{\mathbf{S}}^\ell \mathbf{B}. \tag{13}$$

Note that since $\tilde{\mathbf{S}} = \overline{\mathbf{X}}\overline{\mathbf{X}}^\top$ with $\overline{\mathbf{X}}$ defined in Lemma 3, we can also bypass the materialization of $\tilde{\mathbf{S}}$ in the course of computing $\mathbf{H}^{(a)}$ by reordering the matrix multiplications in Eq. (13).

In analogy to our interpretation of $\mathbf{H}^{(a)}$, our attribute-based graph diffusion is to integrate the high-order node connectivity in $\mathcal{H}$ and structural features from $\mathcal{G}$ into node features $\mathbf{H}^{(a)}$.

Feature Fusion. As aforementioned, $\mathbf{H}^{(t)}$ and $\mathbf{H}^{(a)}$ focus on encoding the node connectivity on $\mathcal{G}$ and $\mathcal{H}$, respectively, both of which partially optimize the overall objective in Eq. (9). In addition, since they are mainly constructed based on data from two different modalities, i.e., graph and attributes, we first align their semantics using layer-wise feature transformations parameterized by learnable weights $\mathbf{W}^{(t)}$ and $\mathbf{W}^{(a)}$, respectively, before the fusion. Mathematically, DGDN combines $\mathbf{H}^{(t)}$ and $\mathbf{H}^{(a)}$ as $\mathbf{H}$ by

$$\mathbf{Z}^{(t)} = \mathbf{H}^{(t)}\mathbf{W}^{(t)}, \mathbf{Z}^{(a)} = \mathbf{H}^{(a)}\mathbf{W}^{(a)}, \mathbf{H} = \beta \cdot \mathbf{Z}^{(t)} + (1-\beta) \cdot \mathbf{Z}^{(a)}, \quad (14)$$

where $\beta \in (0,1)$ is a trade-off parameter. An L_2 normalization will be applied to each row of $\mathbf{Z}$ afterward, to ensure each feature vector has a unit length, i.e., $\|\mathbf{H}_i\|_2 = 1\ \forall v_i \in \mathcal{V}$.

5.2 Graph Diffusion Clustering

Given node representations $\mathbf{H}$, its underlying affinity graph can be constructed by $\mathbf{H}\mathbf{H}^\top$. Based thereon, the optimization objective for NCI computation in Eq. (8) can be reformulated as

$$\min_{\mathbf{C}^\top\mathbf{C}} \operatorname{trace}(\mathbf{C}^\top(\mathbf{I}-\mathbf{\Delta})\mathbf{C}) + \gamma \cdot \operatorname{trace}(\mathbf{C}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{C}) \quad (15)$$

using the fact in Eq. (5). Again, by the Ky Fan's trace maximization principle [23], a simple and direct approach to solve this problem is to conduct the eigen-decomposition of $\mathbf{\Delta} + \gamma \cdot \tilde{\mathbf{A}}$. Unfortunately, this way requires materializing $\mathbf{\Delta} + \gamma \cdot \tilde{\mathbf{A}}$ explicitly, leading to prohibitively expensive computational and space costs of $O(n^2)$.

Lemma 5. *Let* $\overline{\mathbf{H}}_k = \frac{1}{|C_k|}\sum_{v_i \in C_k} \mathbf{H}_i$ *be the centroid embedding of cluster* C_k. *Then,* $\min_{\mathbf{C}^\top\mathbf{C}=\mathbf{I}} \operatorname{trace}(\mathbf{C}^\top(\mathbf{I}-\mathbf{\Delta})\mathbf{C})$ *is equivalent to* $\min_{\mathbf{C}^\top\mathbf{C}=\mathbf{I}} \sum_k^K \sum_{v_i \in C_k} \|\mathbf{H}_i - \overline{\mathbf{H}}_k\|_2^2$.

Our theoretical outcome in Lemma 5 reveals that minimizing the first term of our objective in Eq. (15) is equivalent to minimizing the *within cluster sum of squares* (WCSS) of the node feature vectors in $\mathbf{H}$, which is exactly the objective function of the popular K-Means algorithm. As such, a solution $\mathbf{C}^{(0)}$ that optimizes $\min_{\mathbf{C}^\top\mathbf{C}=\mathbf{I}} \operatorname{trace}(\mathbf{C}^\top(\mathbf{I}-\mathbf{\Delta})\mathbf{C})$ can be efficiently created by executing the K-Means with $\mathbf{H}$. We can further transform the problem in Eq. (15) into the following form:

$$\min_{\mathbf{C}} \|\mathbf{C} - \mathbf{C}^{(0)}\|_F^2 + \gamma \cdot \operatorname{trace}(\mathbf{C}^\top(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{C}),$$

which is again a graph Laplacian smoothing as in Eq. (4). Accordingly, the closed-form solution of the NCI $\mathbf{C}$ can be approximated with L_C iterations:

$$\mathbf{C} = \sum_{\ell=0}^{L_C} \gamma^\ell \tilde{\mathbf{A}}^\ell \cdot \mathbf{C}^{(0)}, \quad (16)$$

which is also a graph diffusion of $\mathbf{C}^{(0)}$ over graph $\mathcal{G}$.

5.3 Training Objectives

To facilitate the model training and the learning of parameters $\mathbf{W}^{(t)}$ and $\mathbf{W}^{(a)}$ in Eq. (14), we introduce three training loss functions

$$\mathcal{L}_{cont} + \mathcal{L}_{cluster} + \mathcal{L}_{recons}. \quad (17)$$

Hierarchical Contrastive Loss. In Section 5.1, we obtain $\mathbf{Z}^{(t)}$ and $\mathbf{Z}^{(a)}$, both of which are transformed from solutions to Eq. (9) from different optimization views. We propose a hierarchical contrastive loss to minimize the variance between the two views. Firstly, we expect the topology and attribute representations of the same node to be similar to capture the inherent invariance:

$$\mathcal{L}_{nod} = \|\mathbf{Z}^{(t)} - \mathbf{Z}^{(a)}\|_F^2.$$

However, constructing positive pairs from the same node under two optimization views neglects the local and intra-cluster invariance [28]. In other words, nodes with strong affinity or topological connections and nodes within the same cluster should also be considered positive pairs. Thus, we additionally introduce the contrastive losses from neighbor and cluster levels. Specifically, we aim to maximize the similarity between the anchor representation and the semantic representation derived from the local neighborhood:

$$\mathcal{L}_{nei}(\mathbf{Z}^{(t)}, \mathbf{Z}^{(a)}) = \sum_{v_i \in \mathcal{V}} \|\mathbf{Z}_i^{(t)} - \frac{1}{d(v_i)} \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{Z}_j^{(a)}\|_2^2.$$

Furthermore, we introduce the cluster-level loss to preserve the intra-cluster semantic similarity across different views:

$$\mathcal{L}_{clu}(\mathbf{Z}^{(t)}, \mathbf{Z}^{(a)}) = \sum_{k=1}^{K} \sum_{v_i \in C_k} \|\mathbf{Z}_i^{(t)} - \overline{\mathbf{Z}}_k^{(a)}\|_2^2,$$

where $\overline{\mathbf{Z}}_k^{(a)}$ is the centroid of cluster C_k from the view of attribute-aware embedding $\mathbf{Z}^{(a)}$, and can be computed by

$$\overline{\mathbf{Z}}_k^{(a)} = \sum_{v_i \in C_k} \mathbf{C}_{ik}\mathbf{Z}_i^{(a)} / \sum_{v_i \in C_k} \mathbf{C}_{ik},$$

with $\mathbf{C}$ obtained in Section 5.2. Finally, we include another term to decorrelate dimensions of learned representations, avoiding learning degenerated embeddings [37, 49]:

$$\mathcal{L}_{dec} = \|\mathbf{Z}^{(t)\top}\mathbf{Z}^{(t)} - \mathbf{I}\|_F^2 + \|\mathbf{Z}^{(a)\top}\mathbf{Z}^{(a)} - \mathbf{I}\|_F^2.$$

Putting these terms together, we can obtain our hierarchical contrastive loss, which reduces the variance between representations under different optimization views from three levels and is proven to maximize the invariant input information in Theorem 1:

$$\begin{aligned}\mathcal{L}_{cont} = w_{dec} \cdot \mathcal{L}_{dec} + w_{cont} \cdot (&\mathcal{L}_{nod} \\ &+ \mathcal{L}_{nei}(\mathbf{Z}^{(t)}, \mathbf{Z}^{(a)}) + \mathcal{L}_{nei}(\mathbf{Z}^{(a)}, \mathbf{Z}^{(t)}) \\ &+ \mathcal{L}_{clu}(\mathbf{Z}^{(t)}, \mathbf{Z}^{(a)}) + \mathcal{L}_{clu}(\mathbf{Z}^{(a)}, \mathbf{Z}^{(t)})).\end{aligned} \quad (18)$$

Clustering Loss. Additionally, we aim to pull each node toward its respective cluster centroid while pushing it away from other clusters in the latent representation space. Specifically, we incorporate an infoNCE loss [65] to enhance the discrimination among clusters by improving the intra-cluster cohesion and reducing the inter-cluster similarity:

$$\mathcal{L}_{cluster} = -\frac{1}{n}\sum_{k=1}^{K} \sum_{v_i \in C_k} \log \frac{\exp(\cos(\mathbf{H}_i, \overline{\mathbf{H}}_k)/\tau)}{\sum_{j \neq k} \exp(\cos(\mathbf{H}_i, \overline{\mathbf{H}}_j)/\tau)}, \quad (19)$$

where $\overline{\mathbf{H}}_k$ is the centroid of cluster C_k based on the fused representation $\mathbf{H}$, $\cos(\cdot,\cdot)$ calculates the cosine similarity, and τ is the temperature parameter [85] controlling the sharpness of the loss.

Reconstruction loss. Finally, to stabilize the training process and preserve input topology and attribute information, we minimize the reconstruction error between the generated embedding $\mathbf{H}$ and the smoothed features $\overline{\mathbf{X}} = \sum_{\ell=0}^{L_X} \eta^\ell \tilde{\mathbf{A}}^\ell \cdot \mathbf{X}$. To align the dimension

Table 3: Dataset Statistics.

	#Nodes	#Edges	#Attributes	#Clusters	HR
Texas	183	325	1,703	5	0.108
Wisconsin	251	515	1,703	5	0.196
Cornell	183	298	1,703	5	0.305
Squirrel	5,201	108,536	2,089	5	0.223
Chameleon	2,277	18,050	2,325	5	0.235
Flickr	7,575	239,738	12,047	9	0.239
Cora	2,708	5,429	1,433	7	0.810
Citeseer	3,327	4,732	3,703	6	0.739
Pubmed	19,717	44,324	500	3	0.771
BlogCatalog	5,196	171,743	8,189	6	0.401
BAT	131	1,038	81	4	0.450
UAT	1,190	13,599	239	4	0.698

between and $\mathbf{H}$ and $\overline{\mathbf{X}}$ and reduce the noise in the input feature, we first decompose $\overline{\mathbf{X}}$ through SVD with rank d, obtaining $\overline{\mathbf{X}} = \mathbf{P}\mathbf{Q}\mathbf{R}^\top$. Then, we minimize the discrepancy between $\mathbf{H}$ and $\hat{\mathbf{X}} = \mathbf{P}\mathbf{Q}$ with the Scaled Cosine Error objective [34]:

$$\mathcal{L}_{recons} = \frac{1}{n}\sum_{v_i \in V}(1 - \cos(\mathbf{H}_i, \hat{\mathbf{X}}_i))^\epsilon, \tag{20}$$

where $\epsilon \geq 1$ is the sharpening parameter controlling the weight for hard samples.

5.4 Theoretical Analyses

Complexity analysis. As established in Lemma 3, we can obtain the d-largest eigenvectors $\mathbf{U}$ of $\tilde{\mathbf{S}}$ via the truncated SVD of $\overline{\mathbf{X}} \in \mathbb{R}^{n\times f}$, reducing the complexity to $O(nfd)$. Thus, the complexity of deriving topology-based embedding $\mathbf{H}^{(t)}$ given in Eq. (12) is $O(mn + nfd)$. When deriving the attribute-based embedding $\mathbf{H}^{(a)}$ in Eq. (13), the complexity of obtaining the d-largest eigenvectors of sparse $\tilde{\mathbf{A}}$ is $O(md)$. Utilizing $\tilde{\mathbf{S}} = \overline{\mathbf{X}}\overline{\mathbf{X}}^\top$ and reordering $\tilde{\mathbf{S}}^\ell\mathbf{U} = \overline{\mathbf{X}}(\overline{\mathbf{X}}^\top\overline{\mathbf{X}})^{\ell-1}(\overline{\mathbf{X}}^\top\mathbf{U})$, the complexity of generating $\mathbf{H}^{(a)}$ is $O(f^3 + nf^2 + nfd)$.

Note that the graph diffusion process in Eq. (12) and Eq. (13) need no update during training and can be taken as a pre-processing step, with a total complexity $O(f^3 + nf^2 + mn + nfd)$. The computation complexity in each training epoch includes only the transformation in Eq. (14), which is $O(nd^2)$, and the graph diffusion clustering in Eq. (16), which is $O(T \cdot ndK + mn)$ with T denoting the maximum iteration in k-Means.

Connection between GDC and Label Propagation. Lemma 5 establishes a connection between Dirichlet Energy of $\mathbf{C}$ over the affinity graph $\mathbf{H}\mathbf{H}^\top$ and the objective of k-Means clustering. Building on this, we transform the minimization of Dirichlet Energy over both input graph $\tilde{\mathbf{A}}$ and affinity graph $\mathbf{H}\mathbf{H}^\top$ into Eq. (16). In this process, we diffuse the k-Means clustering result $\mathbf{C}^{(0)}$ along the input graph $\tilde{\mathbf{A}}$, which shares a formulation similar to that of label propagation algorithm [69]. In the label propagation algorithm, the labels are propagated through the graph, and the labels of unknown nodes are determined by aggregating labels from their neighbors. To the best of our knowledge, there are currently no existing methods that combine clustering and clustering result propagation. Additionally, Section 5.2 demonstrates that diffusing k-Means clustering results over the graph $\mathcal{G}$ is equivalent to minimizing the Dirichlet Energy over both the input graph and affinity graph, which builds a connection between graph diffusion clustering and Dirichlet Energy minimization over attribute graph.

Mutual information interpretation of hierarchical contrastive loss. $\mathbf{Z}^{(t)}$ and $\mathbf{Z}^{(a)}$ are representations of input attribute data under different optimization views. Most existing contrastive graph learning methods augment input data into different views through topology or feature revision [95, 97], while our method implicitly constructs views of input data via different optimization processes, as presented in Section 5.1. We preserve information that remains invariant across different optimization views and minimize noisy information that varies during augmentation, by minimizing the hierarchical contrastive loss $\mathcal{L}_{cont}$ between $\mathbf{Z}^{(t)}$ and $\mathbf{Z}^{(a)}$, as presented in the following theorem:

Theorem 1. *Let X and S denote the random variable of input data and the implicitly constructed views, respectively. Let $\mathbf{Z}^S$ denote the representation of data S and $MI(\cdot,\cdot)$ represents mutual information. Then we have*

$$\min \mathcal{L}_{cont} \Rightarrow \max MI(\mathbf{Z}^S, X) \textit{and} \min MI(\mathbf{Z}^S, S|X).$$

6 Experiments

In this section, we evaluate the effectiveness of DGAC over 12 datasets with different degrees of homophily, against 12 baseline methods[3]. Additionally, we conduct the ablation study, parameter analysis, and embedding visualization to provide deeper insight. All experiments are conducted on a Linux machine equipped with 48GB NVIDIA RTX A6000.

6.1 Experimental Setup

Datasets. We evaluate DGAC on a diverse set of 12 real attributed graphs, comprising 6 heterophilic and 6 homophilic networks. The heterophilic datasets include *Texas* [68], *Wisconsin* [68], and *Cornell* [68], *Squirrel* and *Chameleon* [68, 71], as well as Flickr [51]. For homophilic graphs, we incorporate *Cora* [37],*Citeseer* [37], *Pubmed* [37], *BlogCatalog* [51], and *BAT* and *UAT* [63, 70]. Detailed descriptions of these datasets are provided in Appendix B.1. The statistics of datasets are presented in Table 3, including the number of nodes, edges, attributes, and the number of ground-truth clusters of each dataset. Following the previous work [3, 11, 17, 26, 54, 66, 75], we utilize the labels as the ground-truth clusters. Additionally, we report the homophily ratio [103] of each graph, which is computed as HR= $\sum_{(v_i,v_j)\in\mathcal{E}} \mathbb{1}(y_{v_i} = y_{v_j})/|\mathcal{E}|$, where y_{v_i} is the label of v_i.

Baselines. We compare our method with representative and state-of-the-art graph clustering methods and self-supervised learning methods. Specifically, our competitors include **(i)** methods proposed for heterophily graph clustering: DGCN [66], HoLe [26]; **(ii)** self-supervised learning methods for heterophily graphs: HGRL [11], PolyGCL [10]; **(iii)** methods for homophily graph clustering: AGE [17], MinCutPool [5], SCGC [54], DMoN [75], DGCluster [3]; and **(iv)** self-supervised learning methods for homophily graphs: VGAE [39], DGI [77], CCA-SSG [97]. For self-supervised learning methods, k-Means algorithm is utilized on the learned representations to obtain the final clustering results following [11, 56]. Introduction to the

[3]Code available at https://github.com/kkkkk001/DGAC

Table 4: Clustering results on heterophily graphs. The best and second-best results are indicated in bold and underlined. A baseline is excluded if it cannot finish clustering within 7 days.

Methods	Texas		Wisconsin		Cornell		Squirrel		Chameleon		Flickr		Ave. Rank
	Acc.	NMI	Acc.	NMI	Acc.	NMI	Acc.	NMI	Acc.	NMI	Acc.	NMI	
AGE	53.55	12.81	47.25	10.00	40.77	5.88	28.95	4.61	35.68	11.25	46.14	31.22	6.8
MinCutPool	56.07	2.84	47.73	1.36	56.07	4.23	30.37	6.48	35.42	10.42	33.02	17.37	7.5
SCGC	45.46	13.46	43.67	8.25	36.94	5.30	27.24	5.24	28.61	4.74	19.17	8.41	10.1
DMoN	59.56	13.54	51.63	6.04	55.74	3.46	26.23	1.59	32.96	11.18	40.44	23.48	7.6
DGCluster	41.64	11.60	31.47	6.99	37.27	3.38	22.71	0.92	31.36	7.56	26.31	11.77	11.3
VGAE	53.55	12.24	47.17	9.51	47.43	4.68	23.72	0.99	33.29	10.39	34.00	19.20	8.7
DGI	48.31	15.94	45.58	15.26	42.51	6.01	27.50	4.40	29.51	4.97	18.88	5.29	8.8
CCA-SSG	55.85	6.81	52.03	14.54	57.92	11.84	24.55	2.68	26.48	3.32	27.46	15.52	8.3
DGCN	62.19	22.89	59.68	20.63	56.17	5.96	32.84	9.24	41.64	16.95	21.31	5.14	4.7
HoLe	46.78	12.54	37.69	15.18	40.87	11.59	30.33	4.49	34.32	7.17	63.00	47.10	7.1
HGRL	70.27	41.59	62.23	40.73	69.84	41.51	30.94	8.52	38.73	21.00	50.32	37.44	2.5
PolyGCL	56.50	8.63	62.31	26.15	46.56	5.89	28.16	5.96	36.44	17.59	-	-	5.5
DGAC	**75.08**	**46.19**	**80.88**	**57.58**	**73.33**	**43.24**	**34.43**	**12.24**	**42.02**	**21.99**	**81.59**	**66.36**	1.0

Table 5: Clustering results on homophily graphs. The best and second-best results are indicated in bold and underlined.

Methods	Cora		Citeseer		Pubmed		BlogCatalog		BAT		UAT		Ave. Rank
	Acc.	NMI	Acc.	NMI	Acc.	NMI	Acc.	NMI	Acc.	NMI	Acc.	NMI	
AGE	73.64	**57.47**	66.66	41.00	**71.04**	31.41	60.53	39.99	68.55	49.15	54.81	25.70	3.1
MinCutPool	55.30	41.11	49.46	26.10	57.07	23.00	25.47	7.64	51.76	21.58	54.22	23.28	10.5
SCGC	70.07	52.57	66.05	39.68	38.95	0.47	34.38	18.23	74.35	51.14	54.17	27.21	6.7
DMoN	53.22	39.25	50.31	26.97	59.36	19.77	56.20	34.72	57.71	26.47	55.78	25.96	7.9
DGCluster	57.98	46.46	39.83	20.72	69.03	28.09	44.46	25.46	51.45	20.77	53.39	21.55	9.3
VGAE	71.65	52.52	55.76	29.22	70.28	32.32	25.09	6.96	65.04	38.89	52.64	23.87	7.5
DGI	71.44	53.15	68.79	43.17	65.09	27.25	49.19	27.04	50.38	24.45	54.32	23.65	6.2
CCA-SSG	67.98	53.35	64.30	38.62	63.78	28.19	30.91	15.64	62.29	41.11	49.70	24.45	7.3
GiGaMAE	72.75	57.34	67.74	42.61	68.59	32.39	39.60	23.15	35.11	11.58	49.29	14.99	7.3
DGCN	33.18	5.88	38.58	14.64	46.30	5.14	30.23	10.75	62.44	40.75	50.18	25.94	10.3
HoLe	67.51	53.49	60.55	34.69	42.87	0.33	64.09	45.59	56.18	41.16	52.12	23.57	7.5
HGRL	69.15	51.14	65.04	38.89	58.67	26.92	58.66	45.29	53.89	35.60	48.18	23.13	7.8
PolyGCL	28.99	9.68	30.85	8.24	62.99	24.47	26.62	7.14	48.85	21.07	45.66	14.70	12.5
DGAC	**75.91**	56.18	**69.59**	**43.77**	70.82	**34.26**	**75.21**	**59.90**	**77.56**	**52.62**	**58.57**	**28.40**	1.2

Table 6: Ablation study results.

Methods	Texas		Cornell		Flickr	
	Acc.	NMI	Acc.	NMI	Acc.	NMI
DGAC	75.08	46.19	73.33	43.24	84.82	70.72
DGAC-DGDN	70.38	44.90	66.67	40.71	79.39	63.27
DGAC-GDC	73.50	46.08	72.02	42.46	81.49	66.17
DGAC-$\mathcal{L}_{cont}$	71.58	45.41	70.60	40.38	79.75	64.24

baseline models, and detailed experimental settings can be found in Appendix B.2 and Appendix B.3, respectively.

6.2 Graph Clustering Results

We evaluate the graph clustering performance with four metrics: accuracy, *normalized mutual information* (NMI), *adjusted rand index* (ARI), and F_1 score, which are also adopted in [11, 54, 55, 66]. Table 4 and Table 5 report the clustering performance of DGAC against 12 competitors in terms of accuracy and NMI on 6 heterophilic graphs and 6 homophilic graphs, respectively. Clustering performance in terms of ARI and F_1 score exhibits a similar distribution and can be found in our technical report [1]. All results reported are the mean over five repeated runs.

As we can see, DGAC exhibits superior clustering performance on real-world graphs with different homophily degrees. On heterophilic graphs, DGAC consistently outperforms all other competitors on all 6 graphs in terms of both accuracy and NMI. Specifically, on the most heterophilic graph *Texas* with $HR = 0.11$, our DGAC achieves 4.81% and 4.6% performance improvement regarding accuracy and NMI, respectively. On the social network *Flickr*, our DGAC outperforms the second-best competitor, HoLe, with a notable improvement of 19.26% regarding NMI. Additionally, it can be observed that the average ranks of the 4 heterophily methods are higher than other homophily methods, which demonstrates the challenge of heterophilic graph clustering and the necessity of designing heterophily-specific methods. On the homophily graphs, DGAC outperforms other baselines on most homophily graphs and achieves an average rank of *1.2* over 6 datasets. Specifically, compared with the second-best baselines, DGAC takes a lead by 11.21% on BlogCatalog and 2.79% on BAT in terms of clustering accuracy. Compared to other heterophily methods, our DGAC exhibits stable and superior clustering performance on homophilic graphs, demonstrating the effectiveness of the proposed DE-based objective which takes into consideration both input graph $\mathcal{G}$ and the underlying affinity graph $\mathcal{H}$.

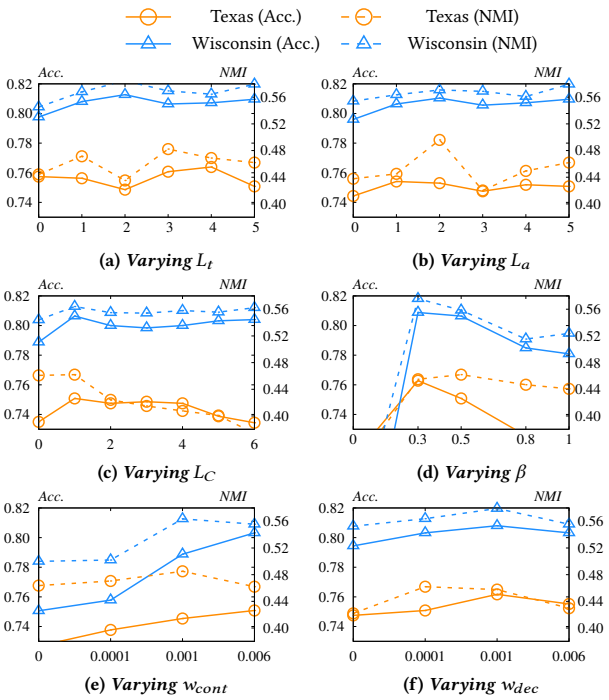


Figure 2: Parameter analysis.

6.3 Ablation study

To analyze the effectiveness of the newly proposed modules inDGAC, we introduce three incomplete variants:

- **DGAC-DGDN** removes the dual graph diffusion networks proposed in Section 5.1 and utilizes only the topology-smoothed features, which are commonly utilized in GNN-based clustering;
- **DGAC-GDC** prohibits the graph diffusion clustering proposed in Section 5.2 and employs only the k-Means algorithm;
- **DGAC-$\mathcal{L}_{cont}$** removes the hierarchical contrastive loss $\mathcal{L}_{cont}$ as presented in Eq. (18).

Table 6 presents the performance of these variants on Texas, Cornell, and Flickr datasets. As we can observe, removing any of these modules leads to notable performance decreases in terms of both accuracy and NMI, demonstrating the effectiveness of these newly proposed modules. Among all variants, DGAC-DGDN brings the most dramatic performance decrease of 4.7% on Texas, 6.66% on Cornell, and 5.43% on Flickr in terms of accuracy, which reflects the importance of minimizing the DE over both $\mathcal{G}$ and $\mathcal{H}$ during the generation of embeddings. Additionally, removing the proposed hierarchical contrastive loss $\mathcal{L}_{cont}$ decreases the clustering performance by 5.07% and 6.48% on Flicker in terms of accuracy and NMI, respectively, showing the effectiveness of $\mathcal{L}_{cont}$ in leaning invariant information between different views. We present the ablation results on homophily graphs in Appendix B.4.

6.4 Parameter Analysis

In this section, we analyze the impact of hyperparameters on the performance of DGAC. We present the clustering performance on Texas and Wisconsin with varying hyperparameters in Figure 2.

Firstly, we vary the number of diffusion layers L_t on $\tilde{\mathbf{A}}$ in Eq. (12), and L_a on $\tilde{\mathbf{S}}$ in Eq. (13), while keeping other parameters fixed. The results are shown in Figures 2(a) and 2(b). Setting L_t or L_a to zero leads to performance decreases on both datasets, demonstrating the importance of diffusing attribute and connection information through $\mathcal{G}$ and $\mathcal{H}$. Increasing L_t and L_a from 0 to 5 improves clustering performance on Wisconsin, while performance on Texas peaks at $L_t = 3$ and $L_a = 2$.

Next, we vary the number of diffusion layers of DGC in Eq. (16), with results presented in Figure 2(c). A notable performance decrease is observed at $L_C = 0$, particularly in accuracy, highlighting the effectiveness of our proposed graph diffusion clustering mechanism. Increasing L_C from 1 to 6 leads to a performance decline on Texas, indicating the importance of an appropriate diffusion depth for clustering results. Then, we analyze the fusion parameter β in Eq. (14), as shown in Figure 2(d). With $\beta = 0$, only $\mathbf{Z}^{(a)}$ is used for clustering, while $\beta = 1$ utilizes only $\mathbf{Z}^{(t)}$. The results demonstrate that a proper β value is crucial for clustering, reflecting the importance of considering representations from both views.

Furthermore, we investigate the coefficients w_{cont} and w_{dec} in $\mathcal{L}_{cont}$ (Eq. (18)). Figures 2(e) and 2(f) illustrate the performance under varying coefficients. A notable increase in clustering performance on both datasets is observed when the w_{cont} increases from 0 to 0.006, demonstrating the effectiveness of the proposed three-level contrastive loss. Additionally, regularizing the representations with the decorrelation w_{dec} is shown to benefit graph clustering. Additionally, we visualize the learned embedding of top-five methods to provide more insight, as shown in Appendix B.5.

7 Conclusion

In this paper, we present DGAC, an effective graph clustering method applicable to both homophilic and heterophilic graphs. We first formulate the objectives of spectral clustering and existing GNNs as a Dirichlet Energy minimization problem and elucidate its connection with graph homophily degree. Building on this insight, we propose a coherent framework with a unified objective to capture the high-order connectivity in both the input graph $\mathcal{G}$ and the affinity graph $\mathcal{H}$ underlying the attribute data, which is formulated as Dirichlet Energy minimization on $\mathcal{G}$ and $\mathcal{H}$. Guided by this unified objective, we develop the dual graph diffusion networks and graph diffusion clustering mechanism, both of which are derived from efficient and theoretically-grounded solutions to the objective. Extensive experiments conducted on 12 real-world graphs, compared against 12 baseline methods, demonstrate the superior performance of DGAC in clustering both heterophilic and homophilic graphs.

Acknowledgments

This work is supported by the Hong Kong RGC ECS grant (No. 22202623), Hong Kong RGC GRF grant (No. 14217322), the National Natural Science Foundation of China (No. 62302414), Hong Kong ITC ITF grant (No. MRP/071/20X), and Tencent Rhino-Bird Focused Research Grant.

References

[1] Technical report. https://github.com/kkkkk001/DGAC/blob/main/DGAC_technical_report.pdf.

[2] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*. 21–29.

[3] Aritra Bhowmick, Mert Kosan, Zexi Huang, Ambuj K. Singh, and Sourav Medya. 2024. DGCLUSTER: A Neural Framework for Attributed Graph Clustering via Modularity Maximization. In *AAAI*. 11069–11077.

[4] Filippo Maria Bianchi. 2023. Simplifying Clustering with Graph Neural Networks. In *NLDL*.

[5] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. 2020. Spectral Clustering with Graph Neural Networks for Graph Pooling. In *ICML*. 874–883.

[6] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond low-frequency information in graph convolutional networks. In *AAAI*. 3950–3957.

[7] Cécile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenkova. 2015. Clustering attributed graphs: models, measures and methods. *Network Science* 3, 3 (2015), 408–444.

[8] David Buterez, Ioana Bica, Ifrah Tariq, Helena Andrés-Terré, and Pietro Liò. 2022. CellVGAE: an unsupervised scRNA-seq analysis workflow with graph attention networks. *Bioinformatics* 38, 5 (2022), 1277–1286.

[9] Sudhanshu Chanpuriya and Cameron Musco. 2022. Simplified graph convolution with heterophily. In *NeurIPS*.

[10] Jingyu Chen, Runlin Lei, and Zhewei Wei. 2024. PolyGCL: Graph Contrastive Learning via Learnable Spectral Polynomial Filters. In *ICLR*.

[11] Jingfan Chen, Guanghui Zhu, Yifan Qi, Chunfeng Yuan, and Yihua Huang. 2022. Towards Self-supervised Learning on Graphs with Heterophily. In *CIKM*. 201–211.

[12] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *ICML*. 1725–1735.

[13] Yi Cheng and Xiuli Ma. 2022. scGAC: a graph attentional architecture for clustering single-cell RNA-seq data. *Bioinformatics* 38, 8 (2022), 2187–2193.

[14] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *ICLR*.

[15] Petr Chunaev. 2020. Community detection in node-attributed social networks: a survey. *Computer Science Review* 37 (2020), 100286.

[16] Fan RK Chung. 1997. *Spectral graph theory*. Vol. 92.

[17] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. 2020. Adaptive Graph Encoder for Attributed Graph Embedding. In *SIGKDD*. 976–985.

[18] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. 2016. Learning Laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing* 64, 23 (2016), 6160–6173.

[19] Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. 2022. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *WWW*. 1550–1558.

[20] Xinyu Du, Xingyi Zhang, Sibo Wang, and Zengfeng Huang. 2023. Efficient Tree-SVD for Subset Node Embedding over Large Dynamic Graphs. *Proc. ACM Manag. Data* 1, 1 (2023), 96:1–96:26.

[21] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2023. Benchmarking graph neural networks. *Journal of Machine Learning Research* 24, 43 (2023), 1–48.

[22] Amani HB Eissa, Mohamed E El-Sharkawi, and Hoda MO Mokhtar. 2018. Towards recommendation using interest-based communities in attributed social networks. In *Companion Proceedings of the The Web Conference*. 1235–1242.

[23] Ky Fan. 1949. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences* 35, 11 (1949), 652–655.

[24] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR*.

[25] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. In *NeurIPS*.

[26] Ming Gu, Gaoming Yang, Sheng Zhou, Ning Ma, Jiawei Chen, Qiaoyu Tan, Meihan Liu, and Jiajun Bu. 2023. Homophily-enhanced Structure Learning for Graph Clustering. In *CIKM*. 577–586.

[27] Lars Hagen and Andrew B Kahng. 1992. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems* 11, 9 (1992), 1074–1085.

[28] Dongxiao He, Jitao Zhao, Rui Guo, Zhiyong Feng, Di Jin, Yuxiao Huang, Zhen Wang, and Weixiong Zhang. 2023. Contrastive Learning Meets Homophily: Two Birds with One Stone. In *ICML*. 12775–12789.

[29] Mingguo He, Zhewei Wei, Hongteng Xu, et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *NeurIPS*.

[30] Xiaofei He, Deng Cai, and Partha Niyogi. 2005. Laplacian score for feature selection. In *NeurIPS*.

[31] Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

[32] Guanhao Hou, Xingguang Chen, Sibo Wang, and Zhewei Wei. 2021. Massively parallel algorithms for personalized pagerank. *PVLDB* 14, 9 (2021), 1668–1680.

[33] Guanhao Hou, Qintian Guo, Fangyuan Zhang, Sibo Wang, and Zhewei Wei. 2023. Personalized pagerank on evolving graphs with an incremental index-update scheme. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–26.

[34] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *SIGKDD*. 594–604.

[35] Glen Jeh and Jennifer Widom. 2003. Scaling personalized web search. In *WWW*. 271–279.

[36] Wei Jin, Tyler Derr, Yiqi Wang, Yao Ma, Zitao Liu, and Jiliang Tang. 2021. Node similarity preserving graph convolutional networks. In *Proceedings of the 14th ACM international conference on web search and data mining*. 148–156.

[37] Wei Jin, Xiaorui Liu, Yao Ma, Charu C. Aggarwal, and Jiliang Tang. 2022. Feature Overcorrelation in Deep Graph Neural Networks: A New Perspective. In *SIGKDD*. 709–719.

[38] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*.

[39] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *CoRR* abs/1611.07308 (2016).

[40] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017*.

[41] Kyle Kloster and David F Gleich. 2014. Heat kernel based community detection. In *SIGKDD*. 1386–1395.

[42] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM*. 2615–2623.

[43] Yiran Li, Gongyao Guo, Jieming Shi, Renchi Yang, Shiqi Shen, Qing Li, and Jun Luo. 2024. A versatile framework for attributed network clustering via K-nearest neighbor augmentation. *The VLDB Journal* 33, 6 (2024), 1913–1943.

[44] Yiran Li, Renchi Yang, and Jieming Shi. 2023. Efficient and Effective Attributed Hypergraph Clustering via K-Nearest Neighbor Augmentation. *Proc. ACM Manag. Data* 1, 2 (2023), 116:1–116:23.

[45] Xiaoyang Lin, Renchi Yang, Haoran Zheng, and Xiangyu Ke. 2024. Spectral Subspace Clustering for Attributed Graphs. *arXiv preprint arXiv:2411.11074* (2024).

[46] Chengyi Liu, Wenqi Fan, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. 2023. Generative diffusion models on graphs: Methods and applications. *arXiv preprint arXiv:2302.02591* (2023).

[47] Chang Liu, Yuwen Yang, Yue Ding, Hongtao Lu, Wenqing Lin, Ziming Wu, and Wendong Bi. 2024. DAG: Deep Adaptive and Generative K-Free Community Detection on Attributed Graphs. In *SIGKDD*.

[48] Fanzhen Liu, Zhao Li, Baokun Wang, Jia Wu, Jian Yang, Jiaming Huang, Yiqing Zhang, Weiqiang Wang, Shan Xue, Surya Nepal, et al. 2022. eRiskCom: an e-commerce risky community detection platform. *VLDBJ* 31, 5 (2022), 1085–1101.

[49] Hua Liu, Haoyu Han, Wei Jin, Xiaorui Liu, and Hui Liu. 2023. Enhancing Graph Representations Learning with Decorrelated Propagation. In *SIGKDD*. 1466–1476.

[50] Huikang Liu, Weijie Wu, and Anthony Man-Cho So. 2016. Quadratic optimization with orthogonality constraints: Explicit lojasiewicz exponent and linear convergence of line-search methods. In *ICML*. 1158–1167.

[51] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. 2022. Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning. *IEEE Trans. Neural Networks Learn. Syst.* 33, 6 (2022), 2378–2392.

[52] Yue Liu, Jun Xia, Sihang Zhou, Siwei Wang, Xifeng Guo, Xihong Yang, Ke Liang, Wenxuan Tu, Stan Z. Li, and Xinwang Liu. 2022. A Survey of Deep Graph Clustering: Taxonomy, Challenge, and Application. *CoRR* abs/2211.12875 (2022).

[53] Yue Liu, Jun Xia, Sihang Zhou, Xihong Yang, Ke Liang, Chenchen Fan, Yan Zhuang, Stan Z Li, Xinwang Liu, and Kunlun He. 2022. A Survey of Deep Graph Clustering: Taxonomy, Challenge, Application, and Open Resource. *arXiv preprint arXiv:2211.12875* (2022).

[54] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Siwei Wang, Ke Liang, Wenxuan Tu, and Liang Li. 2023. Simple contrastive graph clustering. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[55] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. 2023. Hard Sample Aware Network for Contrastive Deep Graph Clustering. In *AAAI*. 8914–8922.

[56] Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. 2022. Towards Unsupervised Deep Graph Structure Learning. In *WWW*. 1392–1403.

[57] Sitao Luan, Chenqing Hua, Qincheng Lu, Liheng Ma, Lirong Wu, Xinyu Wang, Minkai Xu, Xiao-Wen Chang, Doina Precup, Rex Ying, et al. 2024. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges. *arXiv preprint arXiv:2407.09618* (2024).

[58] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. 2022. Revisiting heterophily for graph neural networks. In *NeurIPS*.
[59] Sitao Luan, Mingde Zhao, Chenqing Hua, Xiao-Wen Chang, and Doina Precup. 2020. Complete the missing half: Augmenting aggregation filtering with diversification for graph convolutional networks. *arXiv preprint arXiv:2008.08844* (2020).
[60] Xuexiong Luo, Jia Wu, Amin Beheshti, Jian Yang, Xiankun Zhang, Yuan Wang, and Shan Xue. 2022. Comga: Community-aware attributed graph anomaly detection. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 657–665.
[61] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. 2021. A unified view on graph neural networks as graph signal denoising. In *CIKM*. 1202–1211.
[62] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.
[63] Nairouz Mrabah, Mohamed Bouguessa, Mohamed Fawzi Touati, and Riadh Ksantini. 2023. Rethinking Graph Auto-Encoder Models for Attributed Graph Clustering. *IEEE Trans. Knowl. Data Eng.* 35, 9 (2023), 9037–9053.
[64] Hoang Nt and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550* (2019).
[65] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
[66] Erlin Pan and Zhao Kang. 2023. Beyond Homophily: Reconstructing Structure for Graph-agnostic Clustering. In *ICML*, Vol. 202. 26868–26877.
[67] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS 2019*. 8024–8035.
[68] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *ICLR*.
[69] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 76, 3 (2007), 036106.
[70] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. 2017. *struc2vec*: Learning Node Representations from Structural Identity. In *SIGKDD*. 385–394.
[71] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-Scale attributed node embedding. *J. Complex Networks* 9, 2 (2021).
[72] Jianbo Shi and Jitendra Malik. 2000. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8 (2000), 888–905.
[73] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *SIGKDD*. 990–998.
[74] Lei Tang and Huan Liu. 2011. Leveraging social media networks for classification. *Data mining and knowledge discovery* 23 (2011), 447–478.
[75] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. 2020. Graph Clustering with Graph Neural Networks. *CoRR* abs/2006.16904 (2020).
[76] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
[77] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR 2019*.
[78] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17 (2007), 395–416.
[79] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. 2017. Mgae: Marginalized graph autoencoder for graph clustering. In *CIKM*. 889–898.
[80] Runhui Wang, Sibo Wang, and Xiaofang Zhou. 2019. Parallelizing approximate single-source personalized PageRank queries on shared memory. *VLDB J.* 28, 6 (2019), 923–940.
[81] Sibo Wang, Youze Tang, Xiaokui Xiao, Yin Yang, and Zengxiang Li. 2016. HubPPR: Effective Indexing for Approximate Personalized PageRank. *Proc. VLDB Endow.* 10, 3 (2016), 205–216.
[82] Sibo Wang, Renchi Yang, Runhui Wang, Xiaokui Xiao, Zhewei Wei, Wenqing Lin, Yin Yang, and Nan Tang. 2019. Efficient Algorithms for Approximate Single-Source Personalized PageRank Queries. *ACM Trans. Database Syst.* 44, 4 (2019), 18:1–18:37.
[83] Sibo Wang, Renchi Yang, Xiaokui Xiao, Zhewei Wei, and Yin Yang. 2017. FORA: Simple and Effective Approximate Single-Source Personalized PageRank. In *SIGKDD*. 505–514.
[84] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *ICML*. 6861–6871.
[85] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*. 3733–3742.
[86] Kun Xie, Xiangyu Dong, Yusong Zhang, Xingyi Zhang, Qintian Guo, and Sibo Wang. 2024. Learning-Based Attribute-Augmented Proximity Matrix Factorization for Attributed Network Embedding. *IEEE Trans. Knowl. Data Eng.* 36, 11 (2024), 6517–6531.
[87] Xuanting Xie, Erlin Pan, Zhao Kang, Wenyu Chen, and Bingheng Li. 2024. Provable Filter for Real-world Graph Clustering. *arXiv preprint arXiv:2403.03666* (2024).
[88] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *ICML*. PMLR, 5453–5462.
[89] Lei Xu, Lei Chen, Rong Wang, Feiping Nie, and Xuelong Li. 2024. Joint Feature and Differentiable k-NN Graph Learning using Dirichlet Energy. In *NeurIPS*.
[90] Liang Yang, Chuan Wang, Junhua Gu, Xiaochun Cao, and Bingxin Niu. 2021. Why do attributes propagate in graph convolutional neural networks?. In *AAAI*. 4590–4598.
[91] Renchi Yang and Jieming Shi. 2024. Efficient High-Quality Clustering for Large Bipartite Graphs. *Proc. ACM Manag. Data* 2, 1 (2024), 1–27.
[92] Renchi Yang, Jieming Shi, Xiaokui Xiao, Yin Yang, and Sourav S Bhowmick. 2020. Homogeneous network embedding for massive graphs via reweighted personalized PageRank. *Proceedings of the VLDB Endowment* 13, 5 (2020), 670–683.
[93] Renchi Yang, Jieming Shi, Yin Yang, Keke Huang, Shiqi Zhang, and Xiaokui Xiao. 2021. Effective and scalable clustering on massive attributed graphs. In *WWW*. 3675–3687.
[94] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. In *NeurIPS*. 4805–4815.
[95] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *NeurIPS* (2020), 5812–5823.
[96] Stella X. Yu and Jianbo Shi. 2003. Multiclass Spectral Clustering. In *ICCV*. 313–319.
[97] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S. Yu. 2021. From Canonical Correlation Analysis to Self-supervised Graph Neural Networks. In *NeurIPS 2021*. 76–89.
[98] Xingyi Zhang, Zixuan Weng, and Sibo Wang. 2024. Towards Deeper Understanding of PPR-based Embedding Approaches: A Topological Perspective. In *WWW*. 969–979.
[99] Xingyi Zhang, Kun Xie, Sibo Wang, and Zengfeng Huang. 2021. Learning Based Proximity Matrix Factorization for Node Embedding. In *SIGKDD*. 2243–2253.
[100] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. 2018. Arbitrary-order proximity preserved network embedding. In *SIGKDD*. 2778–2786.
[101] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S Yu, and Shirui Pan. 2022. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082* (2022).
[102] Zhiqiang Zhong, Guadalupe Gonzalez, Daniele Grattarola, and Jun Pang. 2022. Unsupervised Network Embedding Beyond Homophily. *Trans. Mach. Learn. Res.* 2022 (2022).
[103] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*.
[104] Meiqi Zhu, Xiao Wang, Chuan Shi, Houye Ji, and Peng Cui. 2021. Interpreting and unifying graph neural networks with an optimization framework. In *WWW*. 1215–1226.

A Related work

Attributed Graph Clustering. *Attributed graph clustering* (AGC) is extensively studied in the literature [7, 15, 43–45, 53, 93]. In recent years, GNNs achieve notable success on graph-related tasks and have been adopted as an effective methodology for AGC. VGAE [39] employs GCN [40] as the encoder and an inner product decoder to generate interpretable graph representations. Subsequently, MGAE [79] utilizes a graph auto-encoder to obtain graph embeddings, followed by spectral clustering on the learned embeddings for final clustering results. Following this, AGE [17] proposes using the disentangled GNN with the optimal low-pass filter to encode the input graph. Additionally, DiffPool [94] generates soft clustering assignments by applying the softmax function to the output of the GNN. MinCutPool [5] adopts a similar formulation and optimizes the model

Table 7: Links to code of baseline methods.

Method	Link
AGE	https://github.com/thunlp/AGE
MinCutPool	https://github.com/FilippoMB/ Spectral-Clustering-with-Graph-Neural-Networks-for-Graph-Pooling
SCGC	https://github.com/yueliu1999/SCGC
DMoN	https://github.com/google-research/google-research/tree/master/graph_embedding/dmon
DGCluster	https://github.com/pyrobits/DGCluster
VGAE	https://github.com/DaehanKim/vgae_pytorch
DGI	https://github.com/PetarV-/DGI
CCA-SSG	https://github.com/hengruizhang98/CCA-SSG
DGCN	https://github.com/Panern/DGCN
HoLe	https://github.com/galogm/HoLe
HGRL	https://github.com/yifanQi98/HGRL
PolyGCL	https://github.com/ChenJY-Count/PolyGCL

towards minimizing the normalized cut objective [72] and the orthogonality objective. Subsequently, DMoN [75] and JBGNN [4] are proposed to enhance the clustering-based and orthogonality-based objectives in MinCutPool, respectively. Contrastive learning has also been extensively studied in GNN-based graph clustering. SCGC [54] constructs different views with parameter un-shared siamese encoders and embedding perturbation, while HSAN [55] focuses on hard positive and negative samples during contrastive learning. However, these methods primarily rely on typical GNNs, which operate under the homophily assumption, leading to an unsatisfied performance on heterophilic graph clustering as shown in our experiments.

Heterophilic Graph Neural Networks. Nt and Maehara [64] identify that most GNNs act as low-pass filters, smoothing representations of connected nodes. This approach struggles with heterophilic graphs, prompting subsequent research to address this limitation. Specifically, GPR-GNN [14] and ASGC [9] employ generalized polynomial graph filtering with learnable filter weights to adapt to varying degrees of homophily. BernNet [29] utilizes Bernstein polynomials to approximate complex filters. Meanwhile, FB-GNN [59] and ACM-GNN [58] integrate multiple graph filters with learnable inter-filter weights to capture a broader frequency spectrum. Other methods mitigate the impact of heterophilic neighbors by leveraging higher-order relationships within the graph structure. Mixhop [2] and H_2GCN [103] utilize multi-hop message passing from a broader view. SimP-GCN [36] and Geom-GCN [68] extend local neighborhoods based on feature or structural similarity. Additionally, discriminative message-passing mechanisms have been proposed to avoid noisy heterophilic edges, including signed message passing in FAGCN [6] and gating strategy in GBK-GNN [19]. Most of the existing heterophily GNNs rely on node labels for supervision, limiting their applicability in unsupervised tasks, such as graph clustering.

B Additional Experimental Details

B.1 Datasets

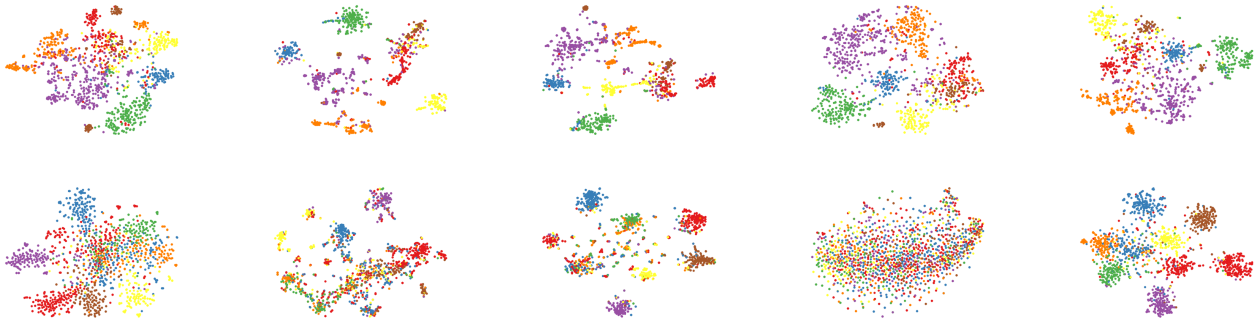
We conduct the graph clustering experiment on 6 heterophily graphs. Specifically, *Texas*, *Wisconsin*, and *Cornell* [68] are WebKB datasets, with nodes representing web pages collected from the corresponding university and edges denoting hyperlinks between web pages. Nodes are associated with the bag-of-words features and are classified into five categories: student, project, course, staff, and faculty. *Squirrel* and *Chamaleon* [68, 71] are Wikipedia page datasets, where nodes denote articles on Wikipedia about a specific topic and edges denote the mutual links between web pages. Node features indicate the presence and absence of several particular nouns in the articles and node labels are the monthly traffic level of the pages. *Flickr* [51] is a social network graph dataset where nodes represent users on the image share website Flickr, and edges denote the following relationship between users. The texts users post on the website are collected as node features and labels are the user interest groups.

We also evaluate our method on 6 homophily datasets. *Cora*, *Citeseer*, and *Pubmed* [37] are citation networks with nodes denoting papers and edges denoting the citation relationship between papers. Each node is associated with a bag-of-word vector as the node feature and the labels are the topics of papers. *BlogCatalog* [51] is another social network dataset. The nodes are website users, with the contents of post blogs as node features, and edges represent the relationship between users. The labels are the blog topics. *BAT* and *UAT* [63, 70] are air-traffic networks with nodes denoting airports and edges denoting commercial flights between airports. Labels are level of airport activity measured in a specific period.

B.2 Baselines

We compare our DGAC against 12 baseline methods:

- **DGCN** [66] constructs two new graphs—one highly homophilic and the other heterophilic—and applies a mixed filter to facilitate clustering.
- **HoLe** [26] improves clustering by refining the graph topology based on high-confidence clustering results.
- **HGRL** [11] learns representations by preserving original features and extracting information from the generalized neighborhood.
- **PolyGCL** [10] employs polynomial filters to conduct contrastive learning between low-pass and high-pass views.
- **AGE** [17] proposes the use of an optimal low-pass filter and iteratively strengthens the filtered features.
- **MinCutPool** [5] optimizes the model to minimize the normalized cut objective.
- **SCGC** [54] constructs different views using un-shared Siamese encoders and embedding perturbation and then conducts contrastive learning between views.
- **DMoN** [75] aims to optimize a modularity-based objective with an orthogonal regularization term.
- **DGCluster** [3] parameterizes the modularity with similarity between nodes and optimizes the modularity-based clustering objective.
- **VGAE** [39] is a graph auto-encoder that uses a GCN model as the encoder and an inner product as the decoder.
- **DGI** [77] learns representations by maximizing the mutual information between patch representations and graph summaries.
- **CCA-SSG** [97] optimizes a feature-level objective inspired by classical Canonical Correlation Analysis.



(a) HGRL. (b) AGE. (c) HoLe. (d) DGI. (e) DGAC.

Figure 3: Visualization on Cora in the first row, and Flickr in the second row.

Table 9: Ablation study results on homophilic graphs.

Methods	Cora		BAT	
	Acc.	NMI	Acc.	NMI
DGAC	75.91	56.18	77.56	52.62
DGAC-DGDN	71.29	54.98	53.89	36.63
DGAC-GDC	74.04	55.96	75.57	52.30
DGAC-$\mathcal{L}_{cont}$	70.84	54.48	52.37	34.38

Table 8: Hyper-parameter setting on each dataset.

Dataset	L	α	L_C	γ	w_{dec}	w_{cont}	β
Texas	5	0.2	1	1	0.0001	0.006	0.5
Wisconsin	5	0.2	6	0.2	0.001	0.006	0.3
Cornell	4	0.2	2	0.2	0.001	0.006	0.5
Squirrel	1	0.2	1	0.1	0.0001	0.0001	0.9
Chameleon	1	0.2	1	0.2	0.0001	0.0001	0.8
Flickr	1	0.8	1	0.1	0.0001	0.0001	0.1
Cora	8	1	8	1	0.001	0.00001	0.9
Citeseer	5	0.8	5	0.8	0.001	0.00001	0.5
Pubmed	1	0.8	1	0.8	0	0.0001	0.5
BlogCatalog	3	0.8	3	0.1	0.0001	0.0001	0.1
BAT	3	1	3	1	0	0.001	1
UAT	3	1	3	1	0	0.0001	1

The code of baseline methods is obtained from the repository provided by the authors. Table 7 summarizes the links to the baseline methods code.

B.3 Detailed Settings

For baselines, we employ the recommended hyper-parameter settings provided by the authors. For the dataset without recommended configuration, we set the values of hyper-parameters by grid search following the search guidance given in the paper. To ensure the fairness of comparison, we fix the embedding dimension and maximum training epoch across all competitors. Following previous work [52, 54, 55], we set the maximum training epoch to 500. For our method, we implement DGAC using PyTorch [67]. We utilize the Adam optimizer [38] with a learning rate of 10^{-3} and a weight decay of 5×10^{-4}. We set the weight of $\mathcal{L}_{cluster}$ to 0.02 and the ϵ in $\mathcal{L}_{recons}$ to 1. We grid search w_{dec} and w_{cont} in Eq. (18) from set $\{10^{-5}, 10^{-4}, 10^{-3}, 6 \times 10^{-3}\}$. In Eq. (12) and Eq. (13), we set $L_t = L_a = L$, and search L and α from range $[1, 10]$ and $[0, 1]$, respectively. In Eq. (16), we set the search range of L_C and γ to $[1, 10]$ and $[0, 1]$. For the trade-off parameter β in Eq. (14), we search from $[0, 1]$. The values of hyper-parameters of each dataset are presented in Table 8.

B.4 Additional Ablation Results

Table 9 presents the ablation study results on two homophilic graphs, Cora and BAT. The results indicate that the removal of any of the three newly proposed components results in a decline in performance. Notably, the absence of the hierarchical contrastive loss causes the most significant performance drop on both graphs. Specifically, removing the hierarchical contrastive loss reduces clustering accuracy on Cora by 5.07%, highlighting the effectiveness of our proposed contrastive learning framework for homophilic graphs. The impact is even more pronounced on BAT, with a performance reduction of 25.19%, suggesting a training collapse in the absence of the hierarchical contrastive loss.

B.5 Visualization

To deepen the understanding of the superior performance of DGAC, we visualize the learned node embedding of the top-five methods, ranked by the average performance across all datasets, which are: *1.* DGAC; *2.* HGRL [11]; *3.* AGE [17]; *4.* HoLe [26]; *5.* DGI [77]. Figure 3 shows the t-SNE [76] visualization of the top-five methods on Cora and Flickr datasets. Nodes are colored according to the ground-truth clusters. As observed, the embedding generated by DGAC demonstrates clear discrimination among different clusters, indicating the potential for accurate clustering.