mathematics

MDPI

Article

A Two-View Hierarchical Contrastive Learning-Driven Method for Community Detection

Shun Liu [1,2,3], Yuzhi Xiao [1,2,3,*], Tao Huang [1,2,3], Yuanli Zhang [1,2,3] and Yifei Wang [1,2,3]

[1] School of Computer, Qinghai Normal University, Xining 810008, China; liushun844127900@outlook.com (S.L.); taohuang658@gmail.com (T.H.)
[2] State Key Laboratory of Tibetan Intelligence, Xining 810008, China
[3] Key Laboratory of Tibetan Information Processing, Ministry of Education, Xining 810008, China
* Correspondence: 1021009@qhnu.edu.cn

Abstract

Effectively integrating graph topology and node attributes, while assigning nodes with both semantic similarity and structural closeness to the same community, remains a key challenge in attributed graph community detection. To address this challenge, this study proposes TVHCL-CD, a two-view hierarchical contrastive learning-driven method for community detection. The proposed method constructs an attribute view and a modularity view from the node attribute matrix and the modularity matrix, respectively, to model attribute semantics and high-order community structure priors. Structure-aware two-view representations are then learned in parallel through dual-view graph attention encoders incorporating multi-order neighborhood priors. Furthermore, a structure-enhanced Graph Transformer fusion module is designed to achieve node-level adaptive fusion of the two-view representations by introducing a learnable adjacency bias into global self-attention and a view-aware gating mechanism into the feed-forward network. To align the optimization objective with community semantics, a hierarchical contrastive learning strategy is further developed. Specifically, view-level consistency contrastive learning constructs modularity-guided augmented views to improve representation robustness, while community-level semantic contrastive learning incorporates partial ground-truth labels to enhance intra-community compactness and inter-community separation. Finally, clustering is performed on the fused representations to obtain community partitions. Experimental results on eight real-world attributed graphs and the generated tree-like attributed graph Tree-2500 indicate that TVHCL-CD achieves competitive performance under the semi-supervised transductive setting, while ablation results support the contributions of its main components.

Keywords: attributed graph community detection; two-view representation learning; modularity-based prior; structure-enhanced Graph Transformer; hierarchical contrastive learning

MSC: 68T07

Check for updates

Academic Editor: Raymond Lee

Received: 11 May 2026
Revised: 10 June 2026
Accepted: 11 June 2026
Published: 14 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

An attributed graph contains both topological connections and node attribute information and is widely observed in real-world scenarios such as social networks, citation networks, recommendation systems, and product association networks [1]. Attributed graph community detection aims to identify groups of nodes with dense internal connections and similar attribute semantics, thereby revealing latent organizational structures

and functional units in complex networks [2]. However, in real-world networks, structural closeness and attribute similarity are not always consistent: some sparsely connected nodes may belong to the same community owing to similar attribute semantics, whereas some densely connected nodes may belong to different communities owing to attribute differences [3]. This issue is also relevant to sparse and tree-like networks, where short-cycle evidence and local closure are limited, and community boundaries depend more on multi-hop structural relationships, modular organization, and attribute consistency [4,5]. Therefore, jointly characterizing local topological dependencies, attribute semantic correlations, and high-order community structures in a unified representation space remains a central challenge in attributed graph community detection [6].

To address this problem, existing studies have explored graph autoencoding, modularity priors, and graph contrastive learning. Graph autoencoding methods usually learn low-dimensional representations by reconstructing topological structures and node attributes. For example, DAEGC [7] characterizes node attributes and topological neighborhood relationships using a graph attention autoencoder, and improves the clustering separability of the embedding space through a self-training mechanism. On this basis, modularity, as a core criterion for measuring community structure quality, has gradually been introduced into deep community detection [8]. DDGAE [9] incorporates the modularity matrix and node attributes into the representation learning process, while DGCLUSTER [10] transforms modularity maximization into a neural-network-optimizable objective, enabling community quality criteria to be directly incorporated into representation learning. Graph contrastive learning enhances semantic discrimination by constructing multi-view sample relationships. GCA [11] constructs adaptive augmented views according to the different contributions of topological structures and attribute information, while MVGRL [12] learns local structures and global dependencies through consistency constraints between the original graph view and the diffusion graph view. MAGI [13] reveals the intrinsic consistency between modularity maximization and graph contrastive learning from a theoretical perspective, showing that the node association relationships contained in the modularity matrix can be used to generate community-aware positive and negative sample pairs.

Although the above methods have promoted the development of attributed graph community detection, three limitations remain. First, modularity priors are still insufficiently exploited for community-semantic representation. Although MAGI reveals the guiding role of modularity relationships in community-aware representation learning, it mainly constructs contrastive signals based on structural relationships and does not transform modularity priors into an independent view that can be collaboratively encoded with node attributes. Second, two-view fusion lacks a deep adaptive mechanism. Although DDGAE introduces the modularity matrix into a two-view attributed graph community detection framework, its inter-view fusion relies on shallow concatenation, making it difficult to achieve deep adaptive fusion and capture global dependencies. Third, the contrastive learning objective is not well aligned with community semantics. Existing graph contrastive learning methods usually focus on instance consistency under perturbed views [11], whereas community detection is more concerned with aggregating nodes within the same community and separating nodes in different communities [13]. Although Community-CL [14] introduces community-level constraints, its semantic signals depend on clustering pseudo-labels, which can be affected by initial bias and may fail to form a stable and reliable community-semantic alignment objective.

To overcome these limitations, we propose TVHCL-CD, a Two-View Hierarchical Contrastive Learning-Driven Community Detection method. First, the proposed method uses node attributes and the modularity matrix as two independent view inputs to characterize node attribute semantics and high-order community structure priors, respectively. The

two views are encoded in parallel by graph attention encoders incorporating multi-order neighborhood priors, thereby producing structure-aware initial representations. Second, a structure-enhanced Graph Transformer fusion module is designed. By introducing a learnable adjacency bias into global self-attention, the module captures long-range dependencies while preserving local structural constraints, and uses a view-aware gating mechanism to achieve node-level adaptive deep fusion of two-view information. Finally, a community-semantics-guided hierarchical contrastive learning strategy is constructed. In this strategy, view-level consistency contrastive learning improves representation robustness through modularity-guided view augmentation, while community-level semantic contrastive learning uses partial ground-truth community labels to model same-community and different-community node relationships, enabling the fused representations to form more discriminative community structures in the representation space. Clustering is then performed on the fused representations to obtain community partitions. The main contributions are summarized as follows:

1. A two-view collaborative encoding framework driven by community structure priors is proposed. The node attribute matrix and the modularity matrix are constructed as the attribute view and the modularity view, respectively, providing direct community-semantic guidance for the model.
2. A structure-enhanced Graph Transformer fusion module is designed. By introducing a learnable graph bias and view-aware gating, the module improves node-level adaptive deep fusion of two-view information under a global receptive field.
3. A community-semantics-guided hierarchical contrastive learning strategy is constructed. While view-level augmentation is used to improve representation robustness, partial ground-truth community labels are incorporated into the community-level contrastive objective to enhance the discriminability of the fused representation space.

2. Related Work

Community detection has evolved into a relatively systematic research field. This section reviews related studies from four perspectives: attribute–topology collaborative modeling, the incorporation of community structure priors, multi-view fusion and discriminative enhancement, and graph contrastive learning for community-discriminative optimization. A compact comparative positioning of representative methods is further provided. The purpose of this review is to summarize the technical evolution and limitations of existing methods, and to clarify the motivation and positioning of this study in terms of attribute semantic modeling, modularity prior utilization, deep cross-view fusion, and community semantic alignment.

2.1. From Topology to Attribute–Topology Collaboration

Early community detection methods mainly focused on topological structures. The GN algorithm [15], the Louvain algorithm [16], and Newman's spectral optimization method [8] achieve community partitioning through edge-betweenness-based splitting, modularity optimization, and spectral partitioning, respectively, laying an important foundation for community structure modeling. However, such methods mainly rely on network connections, making it difficult to utilize node attribute semantics or characterize community boundaries when attribute semantics and topological connections are inconsistent. With the development of graph representation learning, methods such as DeepWalk [17] have advanced community structure analysis from explicit partitioning to node representation learning by learning low-dimensional node embeddings through random-walk contexts. Beyond random-walk-based embeddings, recent studies have further attempted to improve topology-based community recovery by incorporating fine-grained pairwise structural

information. Ran et al. [18] combined machine learning with micro- and mesoscopic statistical-physics methods and used node-pair similarities to refine detected community structures. This indicates that local pairwise relations can provide useful complementary information for mesoscopic community partitioning. Nevertheless, these topology-oriented methods mainly learn from graph connectivity or structural pairwise relations, and they still lack explicit modeling of the interaction between node attributes and community structure priors in attributed graphs.

With the development of attributed graph representation learning, researchers have begun to incorporate node attributes and topological structures into a unified framework. DAEGC [7] jointly characterizes node attributes and neighborhood structures through a graph attention autoencoder, and improves the clustering separability of the embedding space with a self-training mechanism. SSAGCN [3] integrates topological and attribute information through self-supervised adaptive graph convolution to improve community representation capability in attributed networks. Beyond unsupervised and self-supervised attributed graph models, recent studies have also explored the use of supervision signals to enhance community-oriented representation learning. In addition, SDAC-DA [19] uses dual autoencoders to separately model the structural view and the attribute view, and achieves semi-supervised attributed graph clustering through pairwise constraints and collaborative regularization, indicating that limited supervised information can improve attributed graph clustering representations. TAS-Com [6] further uses full human-provided community labels and refines them with the Leiden algorithm, jointly optimizing a modularity-based similarity loss and a refined-label-based similarity loss to balance topological cohesion and label consistency. These methods effectively promote the joint modeling of attribute and topological information. However, their information sources mainly include first-order adjacency relationships, attribute features, pairwise constraints, or refined label signals, and they still do not explicitly construct the modularity matrix as an independent community structure view to model global same-community and different-community tendencies. Therefore, when attribute semantics and topological connections are inconsistent, the model may still produce blurred community boundaries.

2.2. Introduction of Community Structure Priors

To compensate for the insufficient use of community priors in attribute–topology collaborative modeling, researchers have gradually introduced modularity, a traditional metric for evaluating community quality, into deep models. DGCLUSTER [10] transforms modularity maximization into a differentiable objective, enabling community quality criteria to directly participate in neural representation learning. MAGI [13] reinterprets modularity maximization from the perspective of contrastive learning and uses the positive and negative relationships in the modularity matrix to construct community-aware positive and negative sample pairs. DDGAE [9] further uses modularity information as an independent view in a dual graph attention autoencoder, and learns community-oriented representations by reconstructing topological, attribute, and modularity information. These studies show that modularity priors can serve not only as traditional evaluation metrics for community quality, but also as learnable structural signals in deep community detection. Such a modularity-based perspective is not limited to densely connected graphs, because trees and tree-like networks can also exhibit nontrivial modularity despite sparse connectivity [4]. This indicates that sparse tree-like structures should not be treated simply as weak-community cases; instead, their community organization can still be examined through modularity-related structural separation.

However, existing methods still have limitations in their utilization of modularity priors. DGCLUSTER mainly treats modularity as an optimization objective, while MAGI

mainly uses modularity relationships to construct contrastive samples. Neither method collaboratively encodes the modularity matrix and node attributes as two semantically complementary input views. Although DDGAE elevates modularity information to an independent view, its overall framework still follows a graph-attention-autoencoder and reconstruction-driven paradigm, and inter-view interaction mainly relies on joint reconstruction and self-training. It lacks a deep cross-view fusion mechanism oriented toward community priors. Therefore, how to transform the high-order community structural relationships contained in the modularity matrix into representations that can deeply interact with attribute semantics remains a problem requiring further investigation in attributed graph community detection.

2.3. Multi-View Fusion and Discriminative Enhancement

Multi-view modeling and discriminative learning are effective techniques for alleviating the limitations of single-view representations and improving representation separability, and they can provide methodological support for the deep interaction between modularity priors and attribute semantics [20]. Existing surveys have shown that multi-view graph neural networks usually need to establish a reasonable collaborative mechanism between view-specific information and shared information [21]. In multi-view graph representation learning, AM-GCN [22] learns view-specific and shared representations from the topological graph and the feature-neighbor graph, indicating that the fusion of structure and attributes requires explicit modeling. MCGC [23] obtains clustering-friendly graph structures through graph filtering and consensus graph learning. VGMGC [24] infers task-related consensus graphs through a variational graph generator to simultaneously utilize common and view-specific information. These studies show that multi-view modeling helps characterize complementarity among different information sources. However, their views are mostly derived from diffusion graphs, feature-neighbor graphs, consensus graphs, or multimodal graph structures, and are not explicitly constructed around community structure semantics. Therefore, multi-view modeling itself is not equivalent to community-aware modeling. The key lies in whether the view semantics are consistent with the mechanism of community formation, and whether different views can be effectively aligned and fused in a unified representation space.

Relying only on multi-view construction remains insufficient for modeling long-range community dependencies between nodes [25]. In recent years, Transformers have been introduced into graph learning to enhance global interaction capability, providing a new technical path for incorporating structural priors into community detection [26]. Graphormer [27] injects graph structure into the self-attention mechanism through centrality encoding, shortest-path encoding, and edge encoding. CoGT [28] points out that global attention may dilute community semantics and enhances community preservation during global aggregation through a community-aware mechanism. FTAGC [29] applies Graph Transformer to structural-branch modeling in attributed graph clustering and improves the collaboration between structural and attribute representations through adaptive fusion. DMGC-GTN [30] combines positional encoding and path information in multimodal graph clustering to enhance the representation of complex structures. These works indicate that the effectiveness of Graph Transformer depends on positional encoding, structural bias, or task-related priors. For attributed graph community detection, the key is not only to introduce global interaction capability but also to preserve local topological constraints during global dependency modeling and to deeply and adaptively fuse the community-tendency priors contained in modularity with node attribute semantics. This consideration motivates the structure-enhanced Graph Transformer designed in this study.

2.4. Graph Contrastive Learning and Community Discrimination Optimization

In addition to multi-view modeling and Transformer enhancement, contrastive learning is also an important technique for improving the discriminability of node representations, especially for enhancing intra-community cohesion and inter-community separation. However, when existing graph contrastive learning methods are applied to community detection tasks, insufficient integration with community structure priors remains a problem [31].

General graph contrastive learning mainly serves instance-level representation learning. DGI [32] learns node embeddings by maximizing mutual information between local node representations and the global graph summary. MVGRL [12] performs cross-view contrastive learning by exploiting the complementarity between the original graph and the diffusion graph. GCA [11] generates adaptive augmented views according to the importance of topology and attributes to reduce semantic distortion caused by random perturbations. These methods can improve the robustness of node representations, but their optimization objectives mainly emphasize instance consistency under perturbed views, whereas community detection is more concerned with aggregating nodes in the same community and separating nodes in different communities. Therefore, a mismatch remains between general graph contrastive objectives and the objective of community detection.

Contrastive learning methods for graph clustering tasks further design clustering-aware or community-aware contrastive objectives. SC-CGC [33] constructs a structure-sensitive contrastive objective according to the influence of neighbors at different orders. SCAGC [34] and CLAGC [35] use pseudo-label prompts and cluster-preserving augmentation, respectively, to improve clustering consistency. CONVERT [36] and CCGC [37] alleviate semantic drift from the perspectives of reliable augmentation and clustering-guided sample construction. More recent community-aware graph contrastive learning methods further introduce community information into sample construction, augmentation, or contrastive objective design. gCooL [38] jointly learns community partitions and node representations through dense community aggregation and reweighted cross-view contrast. CSGCL [39] introduces community strength to guide attribute and edge augmentations and progressively adjusts the contrastive objective. GCLS^2 [31] models structure semantics by extracting high-level structural views from dense community patterns and optimizing structure contrastive learning. Community-CL [14] further introduces community-level consistency into the community discovery process, showing that relying only on node-level contrast is insufficient to fully adapt to the community detection objective. However, these methods usually rely on augmentation strategies, pseudo-labels, learned partitions, community strength estimates, or structural preprocessing, and inaccurate community signals may still affect representation learning. At the same time, most methods do not use modularity priors as an independent community structure view to participate in contrastive optimization. In contrast, attributed graph community detection requires more stable community semantic constraints, so that the optimization objective can directly serve intra-community cohesion and inter-community separation. Therefore, how to establish a consistent discriminative learning mechanism among attribute semantics, modularity priors, and contrastive objectives remains a problem that existing research needs to further solve. This is precisely the starting point of the community-semantics-guided hierarchical contrastive learning strategy proposed in this study.

2.5. Comparative Positioning of Representative Methods

To further clarify the positioning of TVHCL-CD with respect to representative community detection methods, Table 1 summarizes the differences among related studies in terms of input information, community signal, interaction mechanism, supervision type, and

optimization objective. In this table, A, X, B, and C denote adjacency, attribute, modularity, and pairwise constraint matrices, respectively; AE denotes autoencoder, and CD denotes community detection.

Table 1. Comparative positioning of representative community detection methods.

Method	Input	Community Signal	Interaction	Supervision	Objective
DGCLUSTER	A, X, optional labels	Modularity, attributes	GNN embedding	Unsupervised or auxiliary-supervised	Modularity and attribute losses
MAGI	A, subgraphs	Mini-batch modularity	Modularity contrast	Self-supervised	Contrastive modularity maximization
DDGAE	A, X, B	Modularity view	Dual GAT pooling	Unsupervised self-training	Reconstruction and clustering
SDAC-DA	A, X, C	Pairwise constraints	Dual AE co-regularization	Semi-supervised	Reconstruction, constraints and KL
TAS-Com	A, X, full labels	Refined full labels	Loss-level coupling	Full-label guided	Modularity and label losses
gCooL	A, X, views	Learned communities	DeCA and ReSC	Self-supervised	Node and community contrast
CSGCL	A, X, communities	Community strength	Strength-guided augmentation	Self-supervised	InfoNCE and Team-up contrast
GCLS^2	A, optional X, structural views	Structure semantics	Structure-semantic encoder	Self-supervised	Structure contrastive loss
Ran et al.	A, node-pair features	Node-pair similarity	Ensemble weighting	Unsupervised or label-informed	Similarity prediction and weighted CD
TVHCL-CD	A, X, B, partial labels	Modularity view	Transformer bias and gating	Semi-supervised transductive	View and community contrast

As shown in Table 1, existing methods usually focus on limited community signals or interaction mechanisms. DGCLUSTER and MAGI use modularity for optimization or contrastive learning, while DDGAE introduces the modularity matrix as an additional view but mainly relies on reconstruction-based interaction. Community-aware contrastive methods use pseudo communities, learned partitions, community strength, or structural semantics, but they do not jointly fuse attribute semantics and modularity priors. In contrast, TVHCL-CD constructs an independent modularity view, performs structure-enhanced Transformer fusion, and combines view-level and community-level contrastive objectives under a semi-supervised transductive setting.

3. TVHCL-CD Model

3.1. Problem Description

Attributed graph community detection aims to partition graph nodes into communities with dense internal connections and consistent attribute semantics by jointly exploiting node attributes and topological structures. Accordingly, this study formulates attributed graph community detection as a partitioning problem that should jointly consider structural cohesion, attribute-semantic consistency, and, under the semi-supervised setting, consistency with the available community labels. To address the difficulty of explicitly modeling community affiliation priors using only the original adjacency structure and

node attributes, this study introduces a modularity-based community structure prior and constructs an attribute view and a modularity view. A unified node representation is then learned through two-view collaborative encoding and adaptive fusion, and the final community partitioning results are obtained through clustering. The relevant definitions are given below.

Undirected attributed graph: Given an undirected attributed graph $G = (V, E, A, X^{raw})$, where $V = \{v_1, v_2, \ldots, v_N\}$ is the node set and $N = |V|$ is the number of nodes; $E \in V \times V$ is the edge set; $A \in \{0,1\}^{N \times N}$ is the adjacency matrix without self-loops. If $(v_i, v_j) \in E$, then $A_{ij} = 1$; otherwise, $A_{ij} = 0$. $X^{raw} \in R^{N \times F}$ is the raw node attribute matrix, and its i-th row x_i^{raw} : denotes the F-dimensional raw attribute representation of node v_i. To reduce the influence of different attribute scales across nodes, row-wise ℓ_1 normalization is applied to the raw attribute matrix, i.e., $x_{i:} = x_{i:}^{raw} / max\left(\left\|x_{i:}^{raw}\right\|_1, \epsilon\right)$, where ϵ is a small positive constant for numerical stability. After normalization, the resulting attribute matrix is denoted as X and is used consistently as the node attribute matrix in the following sections.

Modularity matrix: It is difficult to directly perceive the community affiliation tendency between nodes using only A and X. The matrix A mainly characterizes whether nodes are directly connected, while X mainly characterizes node attribute semantics; neither explicitly encodes the community structure preference of node pairs. Therefore, this study introduces the raw modularity matrix $B^{raw} \in R^{N \times N}$, whose elements are defined as follows:

$$B_{ij}^{raw} = A_{ij} - \frac{k_i k_j}{2m_E}, \tag{1}$$

where $k_i = \sum_{j=1}^{N} A_{ij}$ denotes the degree of node v_i, and $m_E = |E| = \frac{1}{2}\sum_{i=1}^{N} k_i$ is the number of edges. B_{ij}^{raw} measures the deviation of the actual connection between nodes v_i and v_j from the expected random connection. Since the modularity matrix is signed and symmetric, this study uses sign-preserving max-absolute normalization. Specifically, the raw modularity matrix is normalized as $B = B^{raw} / \left(max_{i,j}\left|B_{ij}^{raw}\right| + \epsilon\right)$, where ϵ is a small positive constant for numerical stability. In the following sections, B denotes this normalized modularity matrix. Therefore, when $B_{ij} > 0$, the two nodes have a stronger tendency to belong to the same community; when $B_{ij} < 0$, they are more likely to belong to different communities.

Attribute view: The attribute view is denoted as $G^a = (V, A, X)$, and its initial node representation is $H_a^{(0)} = X \in R^{N \times F}$, which characterizes node attribute semantics.

Modularity view: The modularity view is denoted as $G^m = (V, A, B)$, and its initial node representation is $H_m^{(0)} = B \in R^{N \times N}$, which is used to characterize high-order community structure priors.

The two views share the same topological structure A, but have different initial semantic sources. The model learns representations for the two views separately, fuses them into a unified node representation Z through the structure-enhanced Graph Transformer, and finally applies a clustering algorithm to Z to obtain the community partitioning results. The following subsections introduce the community structure pri-or-driven two-view collaborative encoding, the structure-enhanced Graph Transformer adaptive fusion, and the community semantics-guided hierarchical contrastive learning strategy. The main symbols used in the model formulation are summarized in Table 2.

Table 2. Main symbols used in this study and their meanings.

Notation	Description
G	Undirected attributed graph
N	Number of nodes
A	Adjacency matrix without self-loops
X	Node attribute matrix
B	Modularity matrix
G^a	Attribute view
G^m	Modularity view
$H_s^{(l)}$	Node representation of view s at the l-th encoder layer, where $s \in \{a, m\}$
K_h	Number of attention heads in the graph attention encoder
Z^a	Node representation learned from the attribute view
Z^m	Node representation learned from the modularity view
P	Laplacian positional encoding
$U^{(l)}$	Node representation at the l-th layer of the structure-enhanced Graph Transformer
K_t	Number of attention heads in the structure-enhanced Graph Transformer
Γ	View-aware gating matrix
Z	Final fused node representation
V_{tr}	Training node set
y_i	Ground-truth community label of node v_i
τ_v	View-level contrastive temperature
τ_c	Community-level contrastive temperature
λ_c	Weight of the community-level semantic contrastive loss
λ_v	Weight of the view-level consistency contrastive loss, where $\lambda_v = 1 - \lambda_c$
$\mathcal{L}_{view}$	View-level consistency contrastive loss
$\mathcal{L}_{comm}$	Community-level semantic contrastive loss

3.2. Overall Framework of TVHCL-CD

This study proposes a two-view hierarchical contrastive learning-driven community detection method, named TVHCL-CD. The overall framework is shown in Figure 1. The method takes an undirected attributed graph and its modularity matrix B as inputs and jointly models attribute semantics, topological dependencies, and high-order community structure priors. The model consists of three main modules: community structure prior-driven two-view collaborative encoding, structure-enhanced Graph Transformer adaptive fusion, and community semantics-guided hierarchical contrastive learning. Finally, community partitions are obtained from the fused representation Z.

First, the community structure prior-driven two-view collaborative encoding module processes the attribute view G^a and the modularity view G^m in parallel. The attribute view is used to characterize node attribute semantics, while the modularity view is used to represent potential community structure relationships between nodes. To further enhance the ability of the encoding process to characterize local topological dependencies and high-order neighborhood relationships, this study introduces the multi-order neighborhood matrix M into graph attention propagation, enabling the two-view encoder to generate structure-aware node representations while preserving the semantic differences between the two views. Second, the structure-enhanced Graph Transformer adaptive fusion module receives the two-view node representations Z^a and Z^m. To address the problem that traditional concatenation, averaging, or static weighted fusion methods cannot adaptively regulate view contributions and may weaken topological constraints, this study introduces a learnable adjacency bias into global self-attention and designs a view-aware gated feed-forward network. The adjacency bias is used to preserve local structural constraints during global dependency modeling, while the gated feed-forward network dynamically adjusts the contribution of each view according to the different dependencies of different nodes on attribute semantics and modularity structure. In this way, node-level adaptive deep fusion of two-view information is achieved.

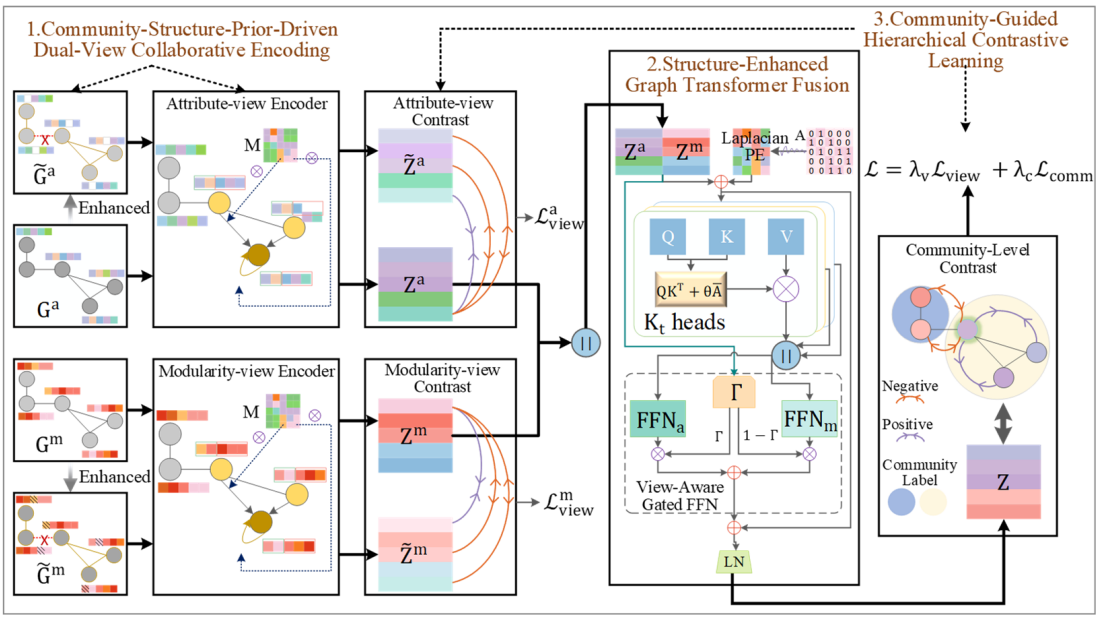


Figure 1. Overall framework of the proposed TVHCL-CD model, including two-view collaborative encoding, structure-enhanced Graph Transformer fusion, and hierarchical contrastive learning.

Finally, the community semantics-guided hierarchical contrastive learning module optimizes node representations from two levels: view consistency and community discrimination. View-level consistency contrastive learning constructs augmented views based on modularity relationships and two-view feature perturbations, and constrains the representational consistency of the same node between the original and augmented views, thereby improving the stability of representation learning. Community-level semantic contrastive learning uses partial ground-truth community labels to characterize the aggregation relationships of nodes in the same community and the separation relationships of nodes in different communities, enabling the fused representations to form more discriminative community structures in the representation space. Thus, TVHCL-CD unifies two-view structure modeling, deep cross-view fusion, and community semantic alignment within the same representation learning framework.

3.3. Community Structure Prior-Driven Two-View Collaborative Encoding

The community structure of an attributed graph is usually jointly determined by node attribute similarity and topological cohesion. If only node attributes are used, the model may ignore the connection patterns within communities; if only the original topological structure is used, the model cannot capture high-dimensional attribute semantics and has difficulty explicitly characterizing the community structure relationship between nodes relative to the expected random connection. Therefore, this study takes the node attribute matrix X and the modularity matrix B as the inputs of the attribute view G^a and the modularity view G^m, respectively, and adopts two-view graph attention encoders with independent parameters for parallel representation learning, enabling the model to preserve both attribute semantic information and high-order community structure priors during the encoding stage.

The attribute view G^a mainly takes $H_a^{(0)} = X \in R^{N \times F}$ as input, while the modularity view G^m mainly takes $H_m^{(0)} = B \in R^{N \times N}$ as input, where the i-th row represents the deviation relationship between node v_i and other nodes relative to the expected random connection. Compared with directly using the adjacency matrix A, the modularity view can explicitly reflect the potential same-community tendency between nodes, allowing the encoding process to rely not only on local edges but also on high-order community

structure priors. The two-view encoders share the same graph topological propagation constraint A, but adopt independent parameters to avoid premature mixing of different semantic sources during the encoding stage.

To enhance the encoder's ability to capture local topological dependencies and multi-hop structural relationships, this study constructs the multi-order neighborhood matrix $M \in R^{N \times N}$. First, symmetric normalization is performed on the adjacency matrix with self-loops, yielding

$$S = \overline{D}^{-\frac{1}{2}} \overline{A} \overline{D}^{-\frac{1}{2}}, \tag{2}$$

where $\overline{A} = A + I$, and $\overline{D}$ is the degree matrix of $\overline{A}$. Furthermore, the structural relationships from the first order to the T-th order are averaged to obtain the multi-order neighborhood matrix:

$$M = \frac{1}{T} \sum_{t=1}^{T} S^t. \tag{3}$$

The matrix M contains both direct neighborhood information and multi-hop neighborhood information, and can reflect smoother structural correlations between nodes. Compared with message propagation relying only on first-order adjacency, this structural prior helps suppress the interference of local noise on community representations, allowing nodes within the same community to obtain more stable structural consistency during the encoding process.

For any view $s \in \{a, m\}$, its encoder consists of two graph attention network layers injected with multi-order neighborhood priors. The first layer contains K_h attention heads to capture neighborhood dependencies in multiple subspaces, whereas the second layer is a single-head graph attention layer that maps the intermediate representation after multi-head concatenation to the unified dimension d_v. In the r-th attention head of the l-th layer, the unnormalized attention score between node v_i and its neighbor node v_j is

$$e_{ij,s}^{(l,r)} = LeakyReLU\left(\left(\rho_s^{(l,r)} \right)^{\top} \left[W_s^{(l,r)} H_{i,s}^{(l-1)} \parallel W_s^{(l,r)} H_{j,s}^{(l-1)} \right] \right), \tag{4}$$

where $W_s^{(l,r)}$ is the linear mapping matrix, $\rho_s^{(l,r)}$ is the attention vector, and $\parallel$ denotes vector concatenation. To make the attention weights constrained by both feature similarity and multi-hop structural correlation, this study introduces the multi-order neighborhood matrix M into the attention calculation as a multiplicative structural bias:

$$\alpha_{ij,s}^{(l,r)} = \frac{exp\left(M_{ij} e_{ij,s}^{(l,r)} \right)}{\sum_{k \in N(i)} exp\left(M_{ik} e_{ik,s}^{(l,r)} \right)}, \tag{5}$$

where $N(i)$ denotes the neighborhood set containing node v_i itself. Thus, information propagation between nodes is no longer determined only by local feature similarity, but is also regulated by multi-hop structural relationships, which helps strengthen the representational consistency of nodes within the same community. The output representation of the r-th attention head of node v_i is

$$o_{i,s}^{(l,r)} = \sum_{j \in N(i)} \alpha_{ij,s}^{(l,r)} W_s^{(l,r)} H_{j,s}^{(l-1)}. \tag{6}$$

In the first layer, the outputs of all attention heads are concatenated and then processed by nonlinear activation and normalization to form the intermediate representation. The overall output is

$$H_s^{(1)} = LN\left(\sigma\left(o_s^{(1,1)} \parallel o_s^{(1,2)} \parallel \cdots \parallel o_s^{(1,K_h)} \right) \right) \in R^{N \times d_h}, \tag{7}$$

where $o_s^{(1,r)}$ denotes the output representation of the r-th attention head over all nodes, K_h is the number of attention heads in the first layer, d_h is the hidden-layer dimension, $\sigma(\cdot)$ is the nonlinear activation function, and $\mathrm{LN}(\cdot)$ denotes layer normalization. The second layer takes $H_s^{(1)}$ as input and obtains the view encoding result through single-head structure-aware attention:

$$Z^s = H_s^{(2)} \in R^{N \times d_v}, s \in \{a, m\}. \tag{8}$$

Thus, the attribute view encoder outputs $Z^a = H_a^{(2)}$, and the modularity view encoder outputs $Z^m = H_m^{(2)}$. The two encoders do not repeatedly encode the same information source; instead, under the same topological propagation constraint, they learn attribute semantic similarity and modularity-based community association, respectively. The multi-order neighborhood matrix M provides an alignable local structural basis for the two views, while the independent encoding paths preserve view-specific information. The two types of representations are complementary in semantic source and unified in propagation constraints, providing a structurally consistent and semantically distinct two-view basis for subsequent deep adaptive fusion.

3.4. Structure-Enhanced Graph Transformer Adaptive Fusion

The two-view encoding outputs Z^a and Z^m represent the attribute semantic representation and the modularity structural representation, respectively. If they are directly fused by concatenation, averaging, or static weighting, it is difficult to characterize the differentiated dependence of different nodes on the two types of information, and it is also difficult to capture long-range relationships across community boundaries within a global receptive field. Therefore, this study designs a structure-enhanced Graph Transformer adaptive fusion module to achieve node-level adaptive deep fusion of two-view representations while preserving local topological constraints.

First, the attribute view representation Z^a, the modularity view representation Z^m, and the Laplacian positional encoding are jointly mapped into the Transformer input. Let $P \in R^{N \times d_p}$ be the Laplacian positional encoding, where d_p is the positional encoding dimension. The initial fusion input is

$$U^{(0)} = [Z^a \parallel Z^m \parallel P]W_e, \tag{9}$$

where W_e is the linear projection matrix, $U^{(0)} \in R^{N \times d_t}$, and where d_t is the Transformer embedding dimension. The positional encoding is used to characterize the structural position of nodes in the global topology, enabling the model to distinguish nodes that are locally similar in attributes but have different overall structural roles.

In the structure-enhanced multi-head global self-attention, the number of global self-attention heads is K_t. This study adds a learnable adjacency bias to the standard Transformer attention scores. In the h-th attention head of the l-th layer, the attention score between node v_i and node v_j is

$$s_{ij}^{(l,h)} = \frac{\left(U_i^{(l-1)} W_Q^{(l,h)}\right)\left(U_j^{(l-1)} W_K^{(l,h)}\right)^\top}{\sqrt{d_r}} + \theta^{(h)} \overline{A}_{ij}, \tag{10}$$

where $W_Q^{(l,h)}$ and $W_K^{(l,h)}$ are the query matrix and key matrix, respectively; d_r is the single-head output representation dimension; and $\theta^{(h)}$ is the learnable structural bias coefficient. This design preserves the ability to model global dependencies while allowing adjacent

nodes to obtain explicit structural bias, thereby reducing the dilution of local community semantics during global interaction. Accordingly, the normalized attention weight is

$$\beta_{ij}^{(l,h)} = \frac{\exp\left(s_{ij}^{(l,h)}\right)}{\sum_{k=1}^{N} \exp\left(s_{ik}^{(l,h)}\right)}. \tag{11}$$

The output of the h-th head of node v_i is

$$\Phi_i^{(l,h)} = \sum_{j=1}^{N} \beta_{ij}^{(l,h)} U_j^{(l-1)} W_V^{(l,h)}, \tag{12}$$

where $W_V^{(l,h)}$ is the value mapping matrix, and $\Phi_i^{(l,h)} \in R^{d_r}$. After the multi-head outputs are concatenated, a linear mapping, residual connection, and layer normalization are applied to obtain the structure-enhanced attention representation of node v_i at the l-th layer:

$$\hat{\Phi}_i^{(l)} = LN\left(U_i^{(l-1)} W_u^{(l)} + \left(\Phi_i^{(l,1)} \parallel \Phi_i^{(l,2)} \parallel \cdots \parallel \Phi_i^{(l,K_t)}\right) W_o^{(l)}\right). \tag{13}$$

Thus, the structure-enhanced attention representation of all nodes at the l-th layer is $\hat{\Phi}^{(l)} \in R^{N \times d_t}$, where d_t is the Transformer embedding dimension. This process enables the model to simultaneously capture local topological constraints and global long-range dependencies, providing more sufficient contextual information for nodes at community boundaries.

To further achieve node-level adaptive fusion of two-view information, this study proposes a view-aware gated feed-forward network. The gating matrix is jointly generated by the attribute view representation Z^a and the modularity view representation Z^m:

$$\Gamma = \sigma\left([Z^a || Z^m] W_g\right), \tag{14}$$

where $\sigma(\cdot)$ is the Sigmoid function, and $\Gamma \in R^{N \times d_t}$ is the node-level gating matrix. This gating mechanism can dynamically adjust the contribution of different information channels during fusion according to the view characteristics of each node.

The structure-enhanced attention representation $\hat{\Phi}^{(l)}$ of the l-th layer is fed into two feed-forward branches:

$$F_a^{(l)} = FFN_a^{(l)}\left(\hat{\Phi}^{(l)}\right), F_m^{(l)} = FFN_m^{(l)}\left(\hat{\Phi}^{(l)}\right), \tag{15}$$

where $F_a^{(l)} \in R^{N \times d_t}$, $F_m^{(l)} \in R^{N \times d_t}$, $FFN_a(\cdot)$ and $FFN_m(\cdot)$ are both two-layer MLPs, which generate the nonlinear transformations of the attribute view branch and the modularity view branch, respectively. Then, the view-aware feed-forward output is

$$U^{(l)} = LN\left(\Gamma \odot F_a^{(l)} + (1 - \Gamma) \odot F_m^{(l)}\right), \tag{16}$$

where $\odot$ denotes element-wise multiplication, and $U^{(l)} \in R^{N \times d_t}$ is the output of the l-th layer of the structure-enhanced Graph Transformer adaptive fusion module. This design avoids using a fixed fusion ratio shared by all nodes, enabling nodes with more reliable attribute semantics to focus more on attribute information and nodes with more significant structural relationships to focus more on modularity view information.

After L layers of the structure-enhanced Graph Transformer, the output representation $U^{(L)}$ is linearly projected to obtain the final fused representation:

$$Z = U^{(L)} W_z, \tag{17}$$

where W_z is the output projection matrix, and $Z \in R^{N \times d_z}$. Thus, TVHCL-CD forms a unified representation among attribute semantics, modularity-based community relationships, multi-order neighborhood structures, and global topological dependencies, providing the basis for the subsequent community semantics-guided hierarchical contrastive learning.

3.5. Community Semantics-Guided Hierarchical Contrastive Learning

Through two-view encoding and structure-enhanced fusion, the fused representation Z integrates attribute semantics, modularity priors, and topological dependencies. To further align the optimization objective with community semantics, we propose a community semantics-guided hierarchical contrastive learning strategy that constrains the representation space at two levels: view consistency and community discrimination. Specifically, view-level consistency contrastive learning constructs augmented views based on modularity relationships and two-view feature perturbations, and maintains the representational consistency of the same node between the original and augmented views. Community-level semantic contrastive learning uses partial ground-truth community labels to define same-community positive samples and different-community negative samples, directly constraining the fused representation Z to form a discriminative structure with intra-community aggregation and inter-community separation.

To preserve the core community semantics of the augmented views, this study constructs an augmented attribute view $\tilde{G}^{a} = \left(V, \tilde{A}, \tilde{X}\right)$ and an augmented modularity view $\tilde{G}^{m} = \left(V, \tilde{A}, \tilde{B}\right)$ based on community structure priors. Specifically, the augmented attribute view obtains the augmented attribute matrix $\tilde{X} \in R^{N \times F}$ by randomly masking the attribute matrix X, thereby simulating attribute missingness and semantic noise. At the topological level, edge dropping is performed only on existing edges. For each edge $(i, j) \in E$, the modularity value B_{ij} is used as its community-association score. The dropping probability is defined according to the relative modularity value among existing edges. Let $b_{max} = \max_{(u,v)\in E} B_{uv}$, the dropping weight of edge (i, j) is defined as $w_{ij} = b_{max} - B_{ij}$ and the corresponding dropping probability is $p_{ij}^{drop} = w_{ij} / \sum_{(u,v)\in E} w_{uv}$. If all existing edges have the same modularity value, the dropping probability is set uniformly over existing edges. Given the preset edge dropping ratio, the required number of undirected edges is sampled without replacement according to p_{ij}^{drop}, and the selected edges are removed symmetrically to produce the augmented adjacency matrix $\tilde{A}$. In this way, edges with stronger positive modularity are more likely to be preserved, whereas edges with lower modularity values are more likely to be removed. For the modularity view, a small Gaussian perturbation is added to B to obtain the augmented modularity matrix $\tilde{B}$, while the signed modularity structure is retained. Thus, the augmented attribute view $\tilde{G}^{a} = \left(V, \tilde{A}, \tilde{X}\right)$ and the augmented modularity view $\tilde{G}^{m} = \left(V, \tilde{A}, \tilde{B}\right)$ introduce moderate perturbations while preserving the main community relationships, providing a reliable contrastive basis for view-level consistency contrast.

For any view $s \in \{a, m\}$, the original representation Z^s and the augmented representation $\tilde{Z}^{s}$ are obtained by the same view encoder $f_s(\cdot)$:

$$Z^s = f_s\left(H_s^{(0)}, A\right), \tilde{Z}^{s} = f_s\left(\tilde{H}_s^{(0)}, \tilde{A}\right), s \in \{a, m\}, \tag{18}$$

where $H_a^{(0)} = X$, $H_m^{(0)} = B$, $\tilde{H}_a^{(0)} = \tilde{X}$ and $\tilde{H}_m^{(0)} = \tilde{B}$. Z^s and $\tilde{Z}^s$ represent the encoding results of nodes under the original view and the augmented view, respectively. For the original graph, Z^a and Z^m are the same view-specific representations used as inputs to the structure-enhanced fusion module. In contrast, the augmented representations $\tilde{Z}^a$ and $\tilde{Z}^m$ are used only for view-level consistency contrast and are not input into the structure-enhanced fusion module.

For the set of all nodes V in the graph, Z^s and $\tilde{Z}^s$ are first L_2-normalized. The representations of the same node under the original view and the augmented view are regarded as positive sample pairs. For each contrastive direction, the representations of all other nodes in the opposite view are treated as negative samples. Then, the contrastive loss from the original representation to the augmented representation is

$$\mathcal{L}_s^{o\to e} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp\left(\left(z_i^s\right)^\top \tilde{z}_i^s/\tau_v\right)}{\sum_{j=1}^{N} exp\left(\left(z_i^s\right)^\top \tilde{z}_j^s/\tau_v\right)}, \tag{19}$$

where τ_v is the view-level contrastive temperature. To symmetrically characterize the node-level correspondence between the original representation and the augmented representation, this study further calculates the reverse contrastive loss from the augmented representation to the original representation:

$$\mathcal{L}_s^{e\to o} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp\left(\left(\tilde{z}_i^s\right)^\top z_i^s/\tau_v\right)}{\sum_{j=1}^{N} exp\left(\left(\tilde{z}_i^s\right)^\top z_j^s/\tau_v\right)}. \tag{20}$$

Therefore, the consistency contrastive loss of view s is

$$\mathcal{L}_{view}^s = \frac{1}{2}(\mathcal{L}_s^{o\to e} + \mathcal{L}_s^{e\to o}). \tag{21}$$

The consistency contrastive losses of the attribute view and the modularity view are further averaged to obtain the view-level contrastive loss:

$$\mathcal{L}_{view} = \frac{1}{2}(\mathcal{L}_{view}^a + \mathcal{L}_{view}^m). \tag{22}$$

Thus, both attribute semantic encoding and modularity prior encoding obtain perturbation stability constraints within their respective views, providing a more reliable representation basis for subsequent community-discriminative optimization of the fused representation.

Community-level semantic contrast acts on the fused representation Z and strengthens the community-discriminative structure of the representation space. Let V_{tr} be the training set. For any node $v_i \in V_{tr}$, let its ground-truth community label be y_i. The same-community positive sample set and different-community negative sample set are defined as follows:

$$P_i = \{p \in V_{tr} \setminus \{i\} | y_p = y_i\}, N_i = \{n \in V_{tr} | y_n \neq y_i\}. \tag{23}$$

Before computing the community-level semantic contrastive loss, the fused representations Z are first $\ell 2$-normalized, so that node similarity is measured by the inner product

of normalized embeddings. Based on the fused representation Z, for anchors with $P_i \neq \varnothing$, the community-level semantic contrastive loss of node v_i is

$$\mathcal{L}_i^{comm} = -\frac{1}{|P_i|}\sum_{p\in P_i} \log \frac{\exp\left(z_i^\top z_p/\tau_c\right)}{\sum_{a\in N_i\cup\{p\}} \exp\left(z_i^\top z_a/\tau_c\right)}, \tag{24}$$

where τ_c is the community temperature parameter. When $P_i = \varnothing$, the anchor node v_i is skipped in the computation of the community-level semantic contrastive loss. The community-level semantic contrastive losses of training nodes with non-empty positive sample sets are summed and averaged to obtain the community-level semantic contrastive loss of the entire model:

$$\mathcal{L}_{comm} = \frac{\sum_{i\in V_{tr}} \mathbf{1}(|P_i| > 0)\mathcal{L}_i^{comm}}{\sum_{i\in V_{tr}} \mathbf{1}(|P_i| > 0)}. \tag{25}$$

where $\mathbf{1}(\cdot)$ is the indicator function, which equals 1 when the condition is satisfied and 0 otherwise. Therefore, only training nodes with at least one same-community positive sample are used as valid anchors in the computation of $\mathcal{L}_{comm}$. This objective takes the ground-truth community labels of training nodes as semantic guidance, enhances the aggregation tendency of same-community nodes in the fused representation space, and expands the representation differences between nodes in different communities.

Finally, the overall optimization objective of TVHCL-CD is

$$\mathcal{L} = \lambda_v\mathcal{L}_{view} + \lambda_c\mathcal{L}_{comm}, \lambda_v = 1 - \lambda_c, 0 \leq \lambda_c \leq 1, \tag{26}$$

where λ_v and λ_c control the contributions of the view-level consistency contrastive loss and the community-level semantic contrastive loss, respectively. Since $\lambda_v = 1 - \lambda_c$, the two loss weights form a normalized trade-off between perturbation-stable representation learning and community-discriminative semantic optimization. $\mathcal{L}_{view}$ is averaged over nodes, contrastive directions, and views, while $\mathcal{L}_{comm}$ is first averaged over same-community positive samples for each valid anchor and then averaged over valid training anchors with non-empty positive sample sets, as shown in Equations (24) and (25).

After training, K-means clustering is performed on the fused representation Z to obtain the community partitioning results. For a specified number of communities K, the predicted community label of node v_i is defined as

$$\hat{y}_i = \arg\min_{1\leq k\leq K} \|z_i - \mu_k\|_2^2, \tag{27}$$

where $\{\mu_k\}_{k=1}^K$ is the k-th cluster center obtained by K-means clustering. In the benchmark evaluation, K is set according to the number of communities specified in the dataset statistics and is used only to determine the number of clusters. Under the semi-supervised transductive setting, clustering is conducted on the learned fused representations of all nodes, while external evaluation metrics are calculated only on the test nodes. Validation and test labels are not used in the K-means clustering step. The final community partitioning result is $\hat{Y} = \{\hat{y}_i | v_i \in V\}$. Since Z integrates attribute semantics, modularity priors, and topological dependencies, the obtained partitioning can more effectively characterize the community structure of attributed graphs in the representation space.

3.6. Complexity and Scalability Analysis

To clarify the computational characteristics of TVHCL-CD, this subsection analyzes the main cost sources of the proposed model and discusses its scalability. TVHCL-CD contains both sparse topology-related operations and full-node pairwise operations. The latter may

cause high peak memory when pairwise score matrices are directly materialized. Therefore, to preserve the full-batch transductive setting, the fixed modularity prior, and the full-node interaction range, this study adopts a memory-controlled full-batch implementation for all datasets. Specifically, the original modularity-view projection is computed by an exact operator-based form without explicitly storing the dense modularity matrix. For pairwise operations that cannot be reduced to this form, including augmented modularity perturbation, global self-attention, and cross-view contrastive similarity, the corresponding row blocks are computed against all N nodes and released after the required accumulation. Thus, the implementation changes only the memory scheduling of pairwise computation, without changing the mathematical objective or the full-node interaction range.

Let N denote the number of nodes, F denote the attribute dimension, m_E denote the number of edges, and T denote the maximum order of the multi-order neighborhood matrix M. Here, d_h denotes the hidden-layer dimension of the first graph attention encoder layer, d_v denotes the unified output dimension of the two-view encoders, $\mathrm{d_t}$ denotes the Transformer embedding dimension, d_r denotes the single-head output dimension in the structure-enhanced Graph Transformer, K_t denotes the number of attention heads in the structure-enhanced Graph Transformer, d_z denotes the dimension of the final fused representation Z, and $|V_{tr}|$ denotes the number of training nodes. In addition, K denotes the number of communities and R_{km} denotes the number of K-means iterations. For memory-controlled pairwise computation, n_{blk} denotes the maximum number of rows processed in one block and is determined by $n_{\mathrm{blk}} = \min\left(N, \max\left(1, \left\lfloor \frac{C_{\mathrm{pair}}}{K_t N} \right\rfloor\right)\right)$, where C_{pair} is the maximum number of pairwise scalar scores allowed to be materialized at one time. This definition uses the Transformer attention as the most memory-demanding row-block case. For operations without multi-head score tensors, the same n_{blk} gives a conservative memory bound. For training components, the complexity is reported as the dominant per-epoch cost under the full-batch transductive setting. Fixed structural preprocessing and post-training K-means clustering are reported as one-time costs. Backpropagation has the same asymptotic order as the corresponding forward computation and is not listed separately. Constant factors caused by fixed numbers of layers are omitted in the big-O notation, while K_t is retained for the Transformer attention cost. The main computational and memory costs are summarized in Table 3.

Table 3. Main computational cost sources of TVHCL-CD under the memory-controlled full-batch implementation.

Component	Main Operation	Dominant Time Cost	Dominant Peak Memory
Attribute-view encoding	Attribute projection and sparse neighborhood attention	$\mathrm{O}(NFd_h + Ndd_hd_v + m_E(d_h + d_v))$	$\mathrm{O}(m_E + \mathrm{N}(d_h + d_v))$
Modularity-view encoding	Operator-based modularity projection and sparse neighborhood attention	$\mathrm{O}((m_E + N)d_h + \mathrm{N}d_hd_v + m_Ed_v)$	$\mathrm{O}(m_E + N(d_h + d_v))$
Augmented modularity perturbation	Row-block perturbation during projection	$O(N^2d_h)$	$\mathrm{O}(n_{\mathrm{blk}}N + Nd_h)$
Multi-order prior M	Fixed-order sparse structural preprocessing	$\mathrm{O}(Tm_E)$	$\mathrm{O}(m_E)$
Transformer fusion	Query-blocked global self-attention	$\mathrm{O}(K_tN^2d_r)$	$\mathrm{O}(K_tn_{\mathrm{blk}}N + Nd_t)$
View-level contrast	Anchor-blocked cross-view similarity	$\mathrm{O}(N^2d_v)$	$\mathrm{O}(n_{\mathrm{blk}}N + Nd_v)$
Community-level contrast	Training-anchor semantic contrast	$\mathrm{O}(\lvert V_{tr}\rvert^2d_z)$	$\mathrm{O}(\lvert V_{tr}\rvert^2)$
K-means clustering	Final clustering on Z	$\mathrm{O}(R_{\mathrm{km}}KNd_z)$	$\mathrm{O}(Nd_z + Kd_z)$

The main scalability pressure comes from full-node pairwise operations. For the original modularity view, $B \in R^{N\times N}$ denotes the normalized modularity matrix defined in Section 3.1. Let $k = (k_1, k_2, \ldots, k_N)^T$ be the degree vector, and let W be the first-layer projection matrix of the modularity-view encoder. According to the definition and normalization of B, the projection BW can be computed as $BW = \left(AW - (1/2m_E)k\left(k^T W\right)\right) / \left(\max_{p,q}\left|B^{raw}_{pq}\right| + \epsilon\right)$, where $B^{raw}_{pq} = A_{pq} - k_p k_q/(2m_E)$), and ϵ is the numerical stability constant used in Section 3.1. This computation preserves the definition of the normalized modularity matrix B and avoids explicitly storing the full dense B during the first projection.

The row-block strategy is used for the augmented modularity perturbation, global self-attention, and view-level contrastive similarity. For each row block, the computation is still performed against all N nodes, so the global attention range and the complete negative sample set are preserved. The augmented modularity perturbation is also generated and applied by row blocks instead of being stored as another dense matrix. These strategies reduce peak memory, but they do not remove the corresponding $O\left(N^2\right)$ pairwise computation and may increase wall-clock time because multiple row blocks must be evaluated sequentially.

Overall, TVHCL-CD is suitable for small- and medium-scale attributed graphs where full-node pairwise operations can be evaluated under the available hardware environment. For larger graphs, augmented modularity perturbation, global self-attention, and full-node contrastive computation remain the main scalability bottlenecks. Possible large-scale extensions include replacing the dense modularity view with a sparse top-k modularity view, replacing global self-attention with sampled attention, and computing contrastive losses with mini-batch nodes or memory queues. These extensions can reduce memory and runtime costs, but they would introduce approximations to the current full-node formulation and are therefore left for future work.

4. Experiments

4.1. Datasets

We conducted experiments on nine attributed networks, including eight real-world attributed networks and one generated tree-like attributed benchmark. These datasets cover citation networks, author collaboration networks, social networks, product networks, and a controlled sparse tree-like network. The real-world datasets are used to evaluate the proposed model under different attribute sources and structural relationships, whereas Tree-2500 is introduced to examine community detection when the graph is sparse and local short-cycle information is limited. Each dataset is regarded as an attributed graph $G = (V, E, A, X^{raw})$, where V denotes the node set, E denotes the edge set, A denotes the adjacency matrix, and raw node attribute matrix X^{raw}. For the real-world datasets, the graph structure and raw node attributes are directly provided by the corresponding datasets. For the generated tree-like benchmark, the adjacency matrix, node attributes, and planted community labels are generated together to provide a controlled attributed graph with sparse tree-like topology. Based on the adjacency matrix A and the node degrees of each graph, the modularity matrix is calculated according to Section 3.1 and used as the modularity-view input after normalization. Table 4 summarizes the statistics of the eight real-world datasets used in this study, where N denotes the number of nodes, m_E denotes the number of edges, F denotes the attribute dimension, and K denotes the number of communities.

Table 4. Statistics of datasets used in this study.

Dataset	Type	N	m_E	F	K
Cora	Citation	2708	5429	1433	7
Citeseer	Citation	3327	4732	3703	6
ACM	Author collaboration	3025	13,128	1870	3
DBLP	Author collaboration	4057	3528	334	4
BlogCatalog	Social	5196	171,743	8189	6
Flickr	Social	7575	239,738	12,047	9
Amazon-Photo	Product	7650	119,081	745	8
Amazon-Computer	Product	13,752	245,861	767	10

The citation networks include Cora [40] and Citeseer [40]. In both datasets, papers are treated as nodes and citation relationships are treated as edges. Node attributes are represented by bag-of-words features extracted from paper texts, and community labels correspond to the research fields of the papers. These datasets have relatively clear topic-related attribute semantics and citation-based topological associations, making them suitable for evaluating whether the model can simultaneously capture paper content attributes and community consistency in citation structures.

The author collaboration networks include ACM [41] and DBLP [41]. In ACM, papers are treated as nodes, and an edge indicates that two papers were written by the same author. Node attributes are constructed from textual terms in paper titles and abstracts. In DBLP, authors are treated as nodes, edges indicate collaboration relationships between authors, and node attributes are constructed from authors' research keywords. Communities in such networks are usually related to research directions, and these datasets can be used to evaluate the model's ability to jointly encode academic semantic attributes and collaboration structures.

The social networks include BlogCatalog [42] and Flickr [42]. BlogCatalog treats blog users as nodes and friendship relationships as edges. Node attributes are constructed from keywords in user profile information, and labels indicate user interest categories. Flickr is derived from an image and video sharing platform, where users are treated as nodes and user relationships are treated as edges. Node labels are assigned according to user interest groups. Compared with citation and author collaboration networks, social networks contain denser connections and more individualized interest-related attribute semantics. Therefore, they can further evaluate the community representation capability of the model under high-dimensional user attributes and complex social topology.

The product networks include Amazon-Photo [43] and Amazon-Computer [43]. Both datasets are derived from Amazon co-purchase networks, where products are treated as nodes and edges indicate co-purchase relationships between products. Node attributes are represented by sparse bag-of-words features extracted from product review texts, and labels correspond to product categories. These networks can be used to examine whether the model can learn community-discriminative representations from product attribute semantics and purchase structures.

In addition to the eight real-world attributed networks, we construct a generated tree-like attributed benchmark, denoted as Tree-2500. This benchmark is motivated by the tree-like network setting used in Ran et al. [18], but it is generated as a larger attributed graph rather than directly reusing their original small tree network. This design is further supported by the observation that tree-like networks can maintain modular organization under sparse connectivity [4]. In such networks, limited short-cycle evidence makes local and quasi-local structural relationships important for characterizing sparse graph organization [44]. Tree-2500 is designed to evaluate whether the proposed model can still

recover community-discriminative representations on sparse graphs where short-cycle evidence is limited. It contains 2500 nodes and five planted non-overlapping communities, with 500 nodes in each community.

For graph generation, a random recursive tree is first generated inside each community, where each newly added node is connected to one existing node according to a mixture of preferential attachment and uniform selection. The preferential attachment coefficient is set to 0.7. The root nodes of adjacent communities are then connected by a chain to make the whole graph connected. To avoid an overly simple pure-tree structure, 20 long-range intra-community edges are added in each community, while candidate edges with existing connections, common neighbors, or shortest-path distance no greater than three are excluded. In addition, 25 inter-community edges are inserted to increase detection difficulty without destroying the planted community structure. The final graph contains 2624 undirected edges and remains sparse and weakly cyclic. Node attributes are generated as sparse binary features with 256 dimensions. Each community is assigned a 40-dimensional community-specific feature block; for each node, 14 active features are sampled from its own block, 5 noise features are sampled from the remaining dimensions, and a feature dropout ratio of 0.15 is applied. The generation settings of Tree-2500 are summarized in Table 5.

Table 5. Generation settings of Tree-2500.

Setting	Value
Number of nodes	2500
Number of planted communities	5
Nodes per community	500
Number of undirected edges	2624
Feature dimension	256
Community-specific feature block size	40
Preferential attachment coefficient	0.7
Long-range intra-community edges	20 per community
Inter-community edges	25
Active community features per node	14
Active noise features per node	5
Feature dropout ratio	0.15

The eight real-world datasets cover various attribute sources, including text attributes, user attributes, and product review attributes, and contain different structural relationships, including citation, collaboration, social, and co-purchase relationships. Tree-2500 further provides a controlled sparse attributed graph with planted communities and weakly cyclic tree-like topology. Therefore, this dataset collection can evaluate the collaborative representation capability of the proposed model for attribute semantics, modularity priors, and topological dependencies across both real-world attributed graph scenarios and a controlled tree-like benchmark.

4.2. Baseline Models

To comprehensively evaluate TVHCL-CD, this paper compares it with representative community detection methods covering different learning paradigms, including unsupervised representation learning, autoencoder-based learning, graph contrastive learning, modularity-oriented learning, semi-supervised learning, and statistical-physics-informed community recovery. Since these methods are designed under different supervision mechanisms, each method is implemented or reproduced according to its original setting, and the results are interpreted together with its learning paradigm. The selected comparison methods are briefly introduced as follows.

DeepWalk [15]: DeepWalk is a graph-structure-based representation learning method that regards random walk sequences as sentences and learns low-dimensional node embeddings through the Skip-gram model.

DAEGC [5]: DAEGC is a deep graph clustering model based on a graph attention autoencoder. It jointly optimizes node representations and graph reconstruction through a KL-divergence-based self-training module, and integrates graph structure and node content information for clustering.

MVGRL [10]: MVGRL is a self-supervised graph representation learning method that learns node-level and graph-level representations by contrasting two structural views generated from the adjacency matrix and the diffusion matrix. This method maximizes mutual information between node representations and graph representations across views, thereby capturing both local and global structural information.

GCA [9]: GCA is a graph contrastive learning method based on adaptive augmentation. It generates two augmented views by dropping edges according to node centrality and masking features according to feature importance, and learns robust node representations by maximizing inter-view consistency with the InfoNCE loss.

SSAGCN [3]: SSAGCN is a multi-topic community detection method based on self-supervised learning. It fuses topological and attribute information through an adaptive graph convolutional network, guides community structure learning by modularity maximization, and reconstructs network topology and node attributes through dual decoders, thereby extracting multiple topics for each community while detecting communities.

MAGI [11]: MAGI defines positive and negative sample pairs in contrastive learning through modularity maximization. It samples subcommunities using two-stage random walks, calculates mini-batch modularity matrices, and combines them with the SimCLR loss for self-supervised graph clustering.

DDGAE [7]: DDGAE is a dual-view graph attention autoencoder that integrates attribute and modularity information. It performs community detection by reconstructing graph topology, attributes, and modularity information, together with joint self-training optimization.

SDAC-DA [17]: SDAC-DA is a semi-supervised deep attributed clustering method based on dual autoencoders. It transforms an attributed graph into a structural view and an attribute view, introduces pairwise constraints into the representation learning process, and integrates the two view-specific embeddings through collaborative regularization. The clustering-friendly representation is then optimized together with a KL-divergence-based clustering objective.

TAS-Com [4]: TAS-Com is a label-guided community detection method that combines a modularity-based similarity objective with a label-consistency objective based on Leiden-refined community structures. It is included as a representative supervision-informed method to evaluate TVHCL-CD against recent community detection approaches that also exploit label-related community signals.

Ran et al. [18]: Ran et al. propose a statistical-physics-informed machine-learning framework that predicts node-pair similarity from micro-structural features and performs community detection on the constructed weighted similarity network. Because this framework is topology-oriented and has been evaluated on tree-like networks, it is used here to compare community recovery on Tree-2500. Since the framework can combine different classifiers and community detection algorithms, the specific evaluated configuration is reported in Section 4.5.3.

4.3. Evaluation Metrics

To comprehensively evaluate the community detection performance of the proposed model, this study adopts external evaluation metrics and internal structural evaluation

metrics. The external evaluation metrics include Accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Macro-F1, which measure the consistency between the predicted community labels and the ground-truth labels. The internal structural evaluation metrics include modularity Q and weighted modularity Q_w, which evaluate the structural quality of the detected communities without using ground-truth labels. Let the test node set be V_{te}, the number of test nodes be $n = |V_{te}|$, the ground-truth community label of node v_i be y_i, and the predicted label obtained by clustering be $\hat{y}_i$. Since clustering labels do not have a fixed semantic order, the Hungarian algorithm, also known as the Munkres algorithm, is used to optimally match predicted labels with ground-truth labels before calculating ACC and Macro-F1 [45], so as to eliminate the influence of label permutation. NMI and ARI are calculated directly based on the ground-truth partition and the predicted partition, without label alignment. ACC, NMI, ARI, and Macro-F1 are calculated on the test nodes under the semi-supervised transductive setting, whereas Q and Q_w are calculated on the complete graph using the predicted partition, because they are label-free structural quality functions. Larger values of all metrics indicate better community detection performance. Therefore, the external metrics and the internal structural metrics are used together to evaluate the two aspects of the attributed community criterion described in Section 3.1, namely semantic consistency and structural cohesion.

ACC measures the consistency between predicted community labels and ground-truth community labels. Let $\pi(\cdot)$ denote the mapping function from predicted labels to ground-truth labels. The optimal mapping π^* is determined by the Hungarian algorithm:

$$\pi^* = arg \max_{\pi} \sum_{i=1}^{n} \mathbf{1}(y_i = \pi(\hat{y}_i)). \tag{28}$$

Then, ACC is defined as

$$ACC = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(y_i = \pi^*(\hat{y}_i)), \tag{29}$$

where $\mathbf{1}(\cdot)$ is the indicator function, which equals 1 when the condition inside the parentheses is satisfied and 0 otherwise. ACC ranges from 0 to 1, and a larger value indicates that the predicted community partition is more consistent with the ground-truth community partition.

NMI measures the shared information between the ground-truth community partition and the predicted community partition from an information-theoretic perspective. Let the ground-truth partition be Y, the predicted partition be $\hat{Y}$, the mutual information between them be $I(Y;\hat{Y})$, and their entropies be $H(Y)$ and $H(\hat{Y})$, respectively. NMI is defined as

$$NMI = \frac{2I(Y;\hat{Y})}{H(Y) + H(\hat{Y})}. \tag{30}$$

The mutual information and entropy are calculated as follows:

$$I(Y;\hat{Y}) = \sum_{p=1}^{K} \sum_{q=1}^{K} \frac{n_{pq}}{n} \log \frac{n \cdot n_{pq}}{a_p b_q}, \tag{31}$$

$$H(Y) = -\sum_{p=1}^{K} \frac{a_p}{n} \log \frac{a_p}{n}, \tag{32}$$

$$H(\hat{Y}) = -\sum_{q=1}^{K} \frac{b_q}{n} \log \frac{b_q}{n}, \tag{33}$$

where n_{pq} denotes the number of nodes shared by ground-truth community p and predicted community q, $a_p = \sum_{q=1}^{K} n_{pq}$, and $b_q = \sum_{p=1}^{K} n_{pq}$. NMI usually ranges from 0 to 1, and a

larger value indicates higher information consistency between the predicted partition and the ground-truth partition.

ARI measures the consistency between the ground-truth partition and the predicted partition at the node-pair level, and corrects for chance agreement caused by random partitioning. Based on the contingency table n_{pq}, where $p = 1, \ldots, r$ indexes the ground-truth communities and $q = 1, \ldots, s$ indexes the predicted communities, let $a_p = \sum_{q=1}^{s} n_{pq}$, $b_q = \sum_{p=1}^{r} n_{pq}$, and $n = \sum_{p=1}^{r} \sum_{q=1}^{s} n_{pq}$. ARI is defined as

$$ARI = \frac{\sum_{p=1}^{r} \sum_{q=1}^{s} \binom{n_{pq}}{2} - \frac{\sum_{p=1}^{r} \binom{a_p}{2} \sum_{q=1}^{s} \binom{b_q}{2}}{\binom{n}{2}}}{\frac{1}{2}\left[\sum_{p=1}^{r} \binom{a_p}{2} + \sum_{q=1}^{s} \binom{b_q}{2}\right] - \frac{\sum_{p=1}^{r} \binom{a_p}{2} \sum_{q=1}^{s} \binom{b_q}{2}}{\binom{n}{2}}}, \tag{34}$$

where for any non-negative integer x, $\binom{x}{2} = x(x-1)/2$ denotes the number of unordered node pairs among x nodes. ARI generally ranges from -1 to 1, and the closer its value is to 1, the higher the consistency between the predicted community partition and the ground-truth community partition.

Macro-F1 comprehensively measures precision and recall for each community category. Before calculation, the optimal mapping π^* is also used to align the predicted labels. For the k-th community, its precision P_k, recall R_k, and F1 value are defined as

$$P_k = \frac{TP_k}{TP_k + FP_k}, R_k = \frac{TP_k}{TP_k + FN_k}, F1_k = \frac{2P_k R_k}{P_k + R_k}, \tag{35}$$

where P_k, R_k, and $F1_k$ denote the precision, recall, and F1 value of the k-th community, respectively, and TP_k, FP_k, and FN_k denote true positives, false positives, and false negatives corresponding to the k-th community, respectively. Macro-F1 takes the arithmetic mean of the F1 values of all communities and is defined as

$$Macro - F1 = \frac{1}{K} \sum_{k=1}^{K} F1_k, \tag{36}$$

Macro-F1 ranges from 0 to 1, and a larger value indicates better balanced recognition performance of the model across different community categories. Compared with ACC, Macro-F1 more fully reflects the average performance across community categories, and is therefore suitable for evaluating community detection results with imbalanced category sizes.

Modularity Q is a classical internal metric for evaluating the structural quality of a community partition on an unweighted graph. It measures whether the number of edges within detected communities is larger than the expected number of such edges under a random graph with the same degree distribution. For an unweighted graph, modularity is defined as

$$Q = \frac{1}{2m} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \mathbf{1}(\hat{y}_i = \hat{y}_j), \tag{37}$$

where m denotes the number of edges, d_i and d_j denote the degrees of nodes i and j, respectively, and A_{ij} denotes the element of the adjacency matrix. A larger Q indicates that the detected communities have stronger internal connectivity than expected at random. Unlike ACC, NMI, ARI, and Macro-F1, modularity does not require ground-truth labels and therefore provides a complementary evaluation of community structure from the perspective of graph topology.

For a weighted similarity network, modularity can be extended to weighted modularity Q_w. Let A^w denote the weighted adjacency matrix, $s_i = \sum_j A_{ij}^w$ denote the weighted

degree, or strength, of node i, and $m_w = \frac{1}{2}\sum_{i,j} A^w_{ij}$ denote the total edge weight. Then, weighted modularity is defined as

$$Q_w = \frac{1}{2m_w}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(A^w_{ij} - \frac{s_i s_j}{2m_w}\right)\mathbf{1}(\hat{y}_i = \hat{y}_j). \tag{38}$$

When all edge weights are equal to 1, the weighted modularity Q_w is numerically equivalent to the standard unweighted modularity Q under the same community partition. Therefore, Q and Q_w provide consistent structural-quality evaluation principles for unweighted and weighted graph representations, respectively.

In implementation, ACC is calculated after label alignment using the Hungarian algorithm implemented by scipy.optimize.linear_sum_assignment. NMI and ARI are calculated using normalized_mutual_info_score and adjusted_rand_score in scikit-learn, respectively. Macro-F1 is calculated using f1_score with average = "macro" after Hungarian label alignment. Modularity Q is computed according to Equation (37) when an unweighted adjacency matrix is used, and weighted modularity Q_w is computed according to Equation (38) when a weighted adjacency matrix or weighted similarity network is used.

4.4. Experimental Settings

We adopt a semi-supervised transductive evaluation setting. For TVHCL-CD, forward propagation is performed on the complete graph, while node labels are divided into training, validation, and test sets at a ratio of 4:2:4. For each random seed, nodes in each ground-truth community are shuffled and then split according to this ratio, so that the split is stratified within each community. Training labels are used only in the community-level semantic contrastive loss, validation labels are used for hyperparameter selection and checkpoint selection, and test labels are used only for final evaluation. The model is trained for a fixed maximum of 150 epochs, and the checkpoint with the highest validation ACC is selected.

For consistency in evaluation, all compared methods are run on the same graph data, and ACC, NMI, ARI, and Macro-F1 are calculated on the same test nodes according to Section 4.3. The split is used to define label usage and external evaluation, but it does not imply that all methods share the same training procedure. Unsupervised and self-supervised methods do not use labels during representation learning or clustering optimization. For methods that use supervised information or supervision-derived constraints, their learning procedures follow the supervision mechanisms specified in the corresponding original methods, while their output partitions are evaluated on the same test nodes under the same metric protocol. For methods with public implementations, the original settings or released code are followed when applicable. For methods without directly available implementations, reproduction follows the algorithmic descriptions and experimental settings reported in the corresponding studies.

All experiments are repeated over 10 fixed random seeds, namely {0,1,2,3,4,5,6,7,8,9}. Unless otherwise specified, reported results are averaged over the 10 runs. Tables with "±" report mean ± standard deviation. For TVHCL-CD, the random seed controls model initialization, data splitting, feature masking, edge dropping, modularity perturbation, and the initialization of the final clustering step.

The final community partition is obtained according to the output form of each method. For TVHCL-CD, K-means is applied to the fused representation Z, with K fixed to the number of communities listed in the dataset statistics. K-means uses k-means++ initialization, n_init = 20, a maximum of 300 iterations, a tolerance of 10^{-4}, and a random state corresponding to the current seed. For methods with their own clustering module or direct

community-assignment output, the predicted partitions produced by the corresponding method are used for evaluation.

The experimental environment is Ubuntu 22.04, Python 3.9, PyTorch 1.13.1, PyTorch Geometric 2.2.0, and CUDA 11.7. All experiments are conducted on a single NVIDIA RTX 4090 GPU with 24 GB memory. For Tree-2500, the generated graph is fixed across all compared methods.

For TVHCL-CD, dataset-specific hyperparameters are selected on the validation set through a limited group-wise search around the default configuration. The search mainly considers parameters that are sensitive to dataset scale, sparsity, and feature dimensionality, including the learning rate, model capacity, dropout rate, and augmentation strength. Specifically, the candidate learning rates are $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}\}$, the Transformer embedding dimensions are {128,192,256}, the GAT dropout rates are {0.4,0.5,0.6,0.7}, and the Transformer dropout rates are {0,0.1,0.2,0.3,0.4}. The candidate feature masking ratios are {0.03,0.05,0.08,0.10,0.15}, the edge dropping ratios are {0.03,0.05,0.08,0.10,0.15}, and the modularity noise standard deviations are {0.01,0.03,0.05,0.08,0.10}. Parameters are adjusted by validation ACC while the remaining settings are kept fixed to the current best configuration. After the final configuration is selected for each dataset, it is fixed for all test runs.

By default, TVHCL-CD uses a two-layer graph attention encoder injected with the multi-order neighborhood prior M. The hidden dimensions are 128 and 64, and the number of attention heads in the first layer is 8. The structure-enhanced Graph Transformer has an embedding dimension of 256, 2 layers, and 4 attention heads. The Laplacian positional encoding dimension is 10, and the maximum order T of the multi-order neighborhood matrix is 2. The dropout rates of the graph attention encoder and the structure-enhanced Graph Transformer are 0.6 and 0.1, respectively. The view-level contrastive temperature, community-level contrastive temperature, and community-level contrastive weight λ_c are 0.7, 0.5, and 0.6, respectively. The learning rate is 0.001, and the feature masking ratio, edge dropping ratio, and modularity noise standard deviation are 0.08, 0.08, and 0.05, respectively.

Cora and ACM use the default configuration. For Citeseer, the edge dropping ratio is 0.05 and the modularity noise standard deviation is 0.03. For DBLP, the feature masking ratio, edge dropping ratio, and modularity noise standard deviation are 0.05, 0.04, and 0.03, respectively. For BlogCatalog and Flickr, the GAT dropout rate is 0.7, the feature masking ratio is 0.10, the edge dropping ratio is 0.10, and the modularity noise standard deviation is 0.03. For Amazon-Photo, the feature masking ratio is 0.05. For Amazon-Computer, the learning rate is 5×10^{-4}, the Transformer embedding dimension is 192, and the feature masking ratio is 0.05. All other parameters follow the default configuration.

4.5. Comparative Experimental Results

This section evaluates TVHCL-CD from three perspectives: real-world attributed graph benchmarks, comparison with supervision-informed methods, and the generated Tree-2500 benchmark. Since the compared methods differ in supervision form, label usage, and community-generation procedure, the results are interpreted under the unified test-node evaluation protocol described in Section 4.4, rather than as comparisons under identical supervision budgets.

4.5.1. Results on Real-World Attributed Graph Benchmarks

This subsection compares TVHCL-CD with DeepWalk, DAEGC, MVGRL, GCA, SSAGCN, MAGI, DDGAE, and SDAC-DA on eight real-world attributed graphs. The results are reported in Table 6. TVHCL-CD is evaluated under the semi-supervised transductive setting, whereas several compared methods are unsupervised or self-supervised and

do not use node labels during representation learning. Therefore, the results should be interpreted as a common benchmark comparison under the same test-node evaluation protocol.

Table 6. Comparison on eight real-world attributed graph benchmarks. Boldface indicates the best result among the methods in this table.

Dataset	Metric	DeepWalk	DAEGC	MVGRL	GCA	SSAGCN	MAGI	DDGAE	SDAC-DA	Ours
Cora	ACC	0.629	0.704	0.705	0.536	0.741	0.760	0.737	0.781	**0.832**
	NMI	0.446	0.529	0.552	0.469	0.565	0.597	0.554	**0.740**	0.716
	ARI	0.359	0.496	0.487	0.303	0.515	0.573	0.525	**0.790**	0.725
	F1	0.631	0.683	0.672	0.457	0.704	0.739	0.714	0.794	**0.824**
Citeseer	ACC	0.424	0.645	0.687	0.605	0.629	0.706	0.716	0.709	**0.738**
	NMI	0.217	0.364	0.437	0.362	0.391	0.452	0.466	0.421	**0.472**
	ARI	0.139	0.378	0.443	0.352	0.379	0.468	0.479	0.385	**0.481**
	F1	0.412	0.622	0.637	0.564	0.581	0.648	0.659	0.639	**0.667**
ACM	ACC	0.729	0.869	0.867	0.705	0.902	0.923	0.920	0.926	**0.936**
	NMI	0.354	0.562	0.609	0.507	0.622	0.630	0.722	0.675	**0.742**
	ARI	0.338	0.595	0.651	0.411	0.679	0.672	0.779	0.783	**0.794**
	F1	0.733	0.847	0.869	0.625	0.895	0.905	0.911	0.919	**0.930**
DBLP	ACC	0.670	0.621	0.718	0.579	0.723	0.802	0.805	0.825	**0.843**
	NMI	0.417	0.325	0.453	0.446	0.472	0.525	0.499	0.557	**0.565**
	ARI	0.461	0.210	0.522	0.427	0.526	0.607	0.557	0.618	**0.631**
	F1	0.674	0.618	0.727	0.501	0.809	0.782	0.798	0.835	**0.848**
BlogCatalog	ACC	0.546	0.625	0.625	0.462	0.515	0.679	0.658	0.771	**0.788**
	NMI	0.333	0.416	0.439	0.355	0.393	0.481	0.455	0.579	**0.602**
	ARI	0.238	0.371	0.360	0.216	0.259	0.445	0.406	0.580	**0.597**
	F1	0.554	0.603	0.594	0.379	0.504	0.662	0.634	0.769	**0.782**
Flickr	ACC	0.305	0.524	0.567	0.486	0.448	0.588	0.595	0.587	**0.618**
	NMI	0.226	0.322	0.395	0.319	0.263	0.409	0.421	0.379	**0.429**
	ARI	0.147	0.257	0.372	0.283	0.235	0.394	0.403	0.311	**0.408**
	F1	0.275	0.483	0.501	0.427	0.371	0.510	0.519	0.501	**0.530**
Amazon-Computer	ACC	0.664	0.712	0.770	0.644	0.796	0.829	0.817	0.832	**0.856**
	NMI	0.478	0.534	0.582	0.511	0.609	0.657	0.621	0.644	**0.668**
	ARI	0.396	0.503	0.609	0.495	0.633	0.696	0.657	0.685	**0.710**
	F1	0.611	0.665	0.727	0.607	0.751	0.778	0.761	0.797	**0.817**
Amazon-Photo	ACC	0.723	0.799	0.811	0.731	0.847	0.925	0.886	0.909	**0.946**
	NMI	0.557	0.640	0.676	0.617	0.707	**0.831**	0.772	0.799	0.829
	ARI	0.565	0.655	0.694	0.624	0.728	**0.853**	0.786	0.810	0.846
	F1	0.711	0.784	0.801	0.708	0.832	0.919	0.873	0.901	**0.939**

As shown in Table 6, TVHCL-CD obtains the best ACC and Macro-F1 on all eight real-world datasets, and obtains the best NMI and ARI on six datasets. These results indicate that, under the adopted semi-supervised transductive setting, the learned representations provide effective node-level community recognition. However, the results are not uniformly dominant across all metrics. On Cora, SDAC-DA obtains higher NMI and ARI, suggesting stronger global partition-level consistency on this citation network. On Amazon-Photo, MAGI slightly outperforms TVHCL-CD in NMI and ARI, showing that modularity-oriented contrastive learning remains competitive in preserving global community structure on this product network.

The results also reflect differences among method types. Label-free representation methods such as DeepWalk, MVGRL, GCA, and MAGI provide useful topology- or contrast-based embeddings, but they do not use training labels during representation learning. Therefore, the performance gains of TVHCL-CD over these methods should be understood within its semi-supervised setting. Compared with DDGAE, which also introduces modularity information, TVHCL-CD generally achieves higher external metrics, suggesting that adaptive fusion and community-level semantic constraints help improve test-node label agreement. At the same time, the strong NMI/ARI of SDAC-DA on Cora and MAGI on Amazon-Photo indicate that different optimization strategies may favor different aspects of community detection.

Overall, the real-world dataset results show that TVHCL-CD performs competitively under the common test-node evaluation protocol, particularly in ACC and Macro-F1. The

exceptions on Cora and Amazon-Photo also indicate that node-level label agreement and global partition consistency are not always optimized by the same method.

4.5.2. Results with Supervision-Informed Methods

This subsection compares TVHCL-CD with SDAC-DA and TAS-Com on three representative real-world attributed graphs, namely Cora, BlogCatalog, and Amazon-Computer. These datasets cover a citation network, a high-dimensional social network, and the largest real-world attributed graph used in this study. The purpose is to examine the behavior of TVHCL-CD when compared with methods that also use supervised information or supervision-derived constraints.

The compared methods use supervised information in different forms. TVHCL-CD uses partial training labels in the community-level semantic contrastive loss, SDAC-DA uses pairwise constraints in a semi-supervised dual-autoencoder framework, and TAS-Com uses label-derived community information with Leiden refinement. Therefore, Table 7 is interpreted as a supervision-informed comparison under the same reported metrics, rather than as a strictly identical-label-budget comparison.

Table 7. Comparison with supervision-informed methods on representative real-world attributed graphs. Values are reported as mean ± standard deviation over 10 seeds, and Q denotes modularity computed on the unweighted graph. Boldface indicates the best result among the methods in this table.

Dataset	Model	ACC	NMI	ARI	Macro-F1	Q
Cora	TVHCL-CD	**0.832 ± 0.0187**	0.716 ± 0.0142	0.725 ± 0.0206	**0.824 ± 0.0179**	0.801 ± 0.0061
	TAS-Com	0.772 ± 0.0256	0.651 ± 0.0065	0.656 ± 0.0235	0.762 ± 0.0167	**0.817 ± 0.0048**
	SDAC-DA	0.781 ± 0.0214	**0.740 ± 0.0107**	**0.790 ± 0.0158**	0.794 ± 0.0196	0.776 ± 0.0074
BlogCatalog	TVHCL-CD	**0.788 ± 0.0286**	**0.602 ± 0.0219**	**0.597 ± 0.0304**	**0.782 ± 0.0267**	0.640 ± 0.0118
	TAS-Com	0.742 ± 0.0180	0.548 ± 0.0090	0.537 ± 0.0200	0.735 ± 0.0120	**0.651 ± 0.0050**
	SDAC-DA	0.771 ± 0.0248	0.579 ± 0.0256	0.580 ± 0.0279	0.769 ± 0.0235	0.621 ± 0.0136
Amazon-Computer	TVHCL-CD	**0.856 ± 0.0135**	**0.668 ± 0.0108**	**0.710 ± 0.0157**	**0.817 ± 0.0146**	0.609 ± 0.0058
	TAS-Com	0.803 ± 0.0110	0.612 ± 0.0060	0.640 ± 0.0140	0.758 ± 0.0100	**0.615 ± 0.0040**
	SDAC-DA	0.832 ± 0.0162	0.644 ± 0.0121	0.685 ± 0.0139	0.797 ± 0.0158	0.592 ± 0.0064

As shown in Table 7, TVHCL-CD obtains the best ACC and Macro-F1 on all three datasets, and obtains the best NMI and ARI on BlogCatalog and Amazon-Computer. On Cora, SDAC-DA achieves higher NMI and ARI, which indicates that its pairwise-constraint-based dual-autoencoder framework can produce strong global partition consistency on this dataset. In contrast, TVHCL-CD obtains higher ACC and Macro-F1 on Cora, suggesting stronger test-node label agreement under the present evaluation setting.

The modularity results show a different tendency. TAS-Com obtains slightly higher Q on the three datasets. This is consistent with its modularity-related optimization and Leiden-based refinement, which place greater emphasis on topological cohesion. TVHCL-CD obtains higher external metrics on most cases in Table 7, but it does not produce the highest modularity. This result indicates that external label agreement and internal structural modularity provide complementary perspectives and may favor different community partitions.

Overall, the supervision-informed comparison shows that TVHCL-CD performs competitively among methods that also introduce supervised information or supervision-derived constraints. At the same time, the results do not indicate uniform dominance: SDAC-DA is stronger in NMI/ARI on Cora, and TAS-Com obtains higher modularity Q.

4.5.3. Results on the Tree-2500

The Tree-2500 experiment is designed to examine the behavior of TVHCL-CD on a controlled sparse tree-like attributed graph. Unlike the real-world datasets in Tables 6 and 7, Tree-2500 has planted communities, sparse topology, weak cyclic structure, and synthetic node attributes. Therefore, this experiment provides an additional test of whether the proposed model can still learn community-discriminative representations when the graph structure is closer to a tree-like network.

Table 8 compares TVHCL-CD with TAS-Com and Ran et al. (GT DT-Infomap). In the framework of Ran et al. [18], different machine-learning classifiers can be combined with different community detection algorithms after constructing a weighted similarity network. GT DT-Infomap denotes the ground-truth-mode variant that uses the planted labels of Tree-2500 to construct node-pair training targets, uses Decision Tree to estimate node-pair similarity, and then applies Infomap to the resulting weighted similarity network for community detection. This variant is selected because it gives strong and balanced external-metric performance among the corresponding weighted Infomap variants on Tree-2500, while keeping the table focused rather than listing all classifier–detector combinations. Because Ran et al. in this setting uses stronger planted-label information than TVHCL-CD, the results are interpreted as a label-informed community-structure recovery reference rather than as a comparison under identical supervision budgets.

Table 8. Comparison on the generated Tree-2500 benchmark. Values are reported as mean ± standard deviation over 10 seeds, and Q_w denotes weighted modularity. Boldface indicates the best result among the methods in this table.

Dataset	Model	ACC	NMI	ARI	Macro-F1	Q_w
Tree-2500	Ran et al. (GT DT-Infomap)	0.9342 ± 0.0138	**0.8276 ± 0.0253**	**0.8444 ± 0.0302**	0.9343 ± 0.0139	0.7864 ± 0.0014
Tree-2500	TAS-Com	0.8660 ± 0.0236	0.7940 ± 0.0274	0.7510 ± 0.0258	0.8520 ± 0.0243	**0.8660 ± 0.0149**
Tree-2500	TVHCL-CD	**0.9410 ± 0.0177**	0.8160 ± 0.0346	0.8380 ± 0.0273	**0.9390 ± 0.0228**	0.7780 ± 0.0084

The last column in Table 8 reports weighted modularity Q_w. For weighted similarity networks, Q_w is calculated using the corresponding edge weights. For methods producing partitions on the original unweighted Tree-2500 graph, unweighted edges are treated as unit-weight edges when calculating Q_w.

As shown in Table 8, TVHCL-CD obtains the best ACC and Macro-F1 on Tree-2500, indicating effective node-level community recovery on this sparse tree-like attributed benchmark. Ran et al. obtains the best NMI and ARI under its stronger label-informed node-pair similarity setting, showing that the statistical-physics-informed framework is competitive in global partition consistency. TAS-Com obtains the highest Q_w, indicating stronger modularity-oriented structural cohesion on this generated graph.

The Tree-2500 results again show that different metrics emphasize different aspects of community detection. TVHCL-CD performs well in test-node label agreement, Ran et al. shows advantages in global partition consistency under a stronger label-informed setting, and TAS-Com produces the highest weighted modularity. Therefore, this benchmark supports the competitiveness of TVHCL-CD on sparse tree-like attributed graphs, while also showing that external agreement and structural modularity are not always optimized by the same method.

In summary, the comparative results across real-world attributed graphs, supervision-informed methods, and Tree-2500 show that TVHCL-CD provides competitive ACC and Macro-F1 under the semi-supervised transductive setting. Its advantages are metric-dependent when NMI, ARI, and modularity are considered. These results support the

effectiveness of TVHCL-CD, but they do not imply uniform dominance over all methods, supervision settings, and evaluation criteria.

4.6. Parameter Sensitivity Analysis

To analyze the influence of key hyperparameters on the performance of TVHCL-CD, this study selects the Cora dataset for parameter sensitivity experiments, focusing mainly on three parameters: the view-level contrastive temperature, the community-level contrastive temperature, and the community-level contrastive weight. The view-level contrastive temperature adjusts the strength of the consistency constraint between the original view and the augmented view. The community-level contrastive temperature controls the similarity distribution between nodes in the same community and nodes in different communities in the fused representation. The community-level contrastive weight balances the contributions of the view-level consistency constraint and the community-level discriminative constraint to the overall optimization objective. In the experiments, all other parameters are kept unchanged, only one parameter is adjusted at a time, and ACC, NMI, ARI, and F1 are used to evaluate model performance. Figure 2 reports the sensitivity results for the two temperature parameters: Figure 2a shows the effect of the view-level contrastive temperature, and Figure 2b shows the effect of the community-level contrastive temperature.

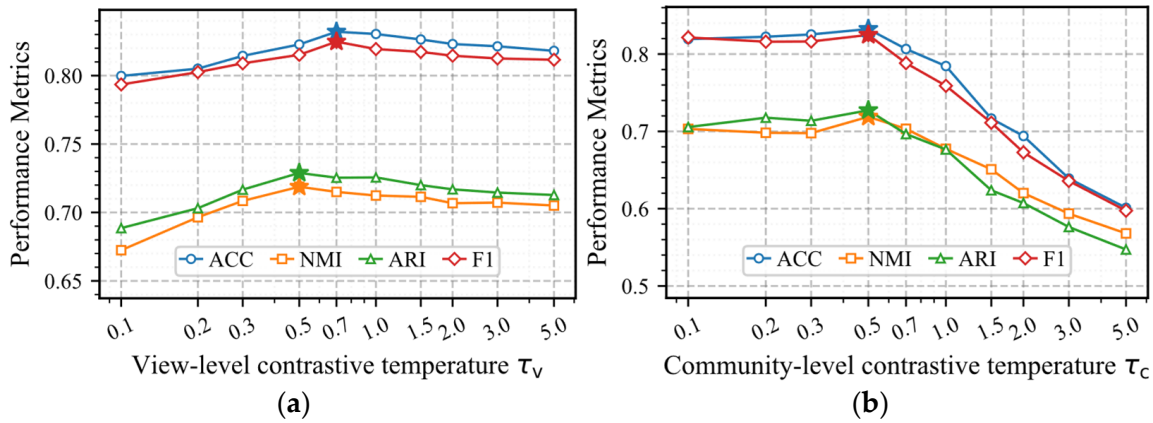


Figure 2. Influence of temperature parameters on the performance of the TVHCL-CD model. (**a**) Effect of the view-level contrastive temperature; (**b**) effect of the community-level contrastive temperature. Star markers indicate the highest values of the corresponding performance metrics.

As shown in Figure 2a, as the view-level contrastive temperature increases from 0.1, ACC and F1 reach their optimum near 0.7, while NMI and ARI reach their optimum at 0.5. Overall, the range of 0.5–0.7 yields relatively better performance. When the temperature continues to increase to 1.0 or above, all metrics show a slow downward trend, with a limited decrease. This result indicates that view-level consistency contrast is relatively stable with respect to temperature changes. A smaller temperature makes the contrastive objective overemphasize locally similar samples, weakening the stable alignment between attribute semantic encoding and modularity prior encoding. An overly large temperature reduces the discriminative ability for negative samples, making the perturbation-robustness constraint provided by the augmented view insufficient. The temperature range from 0.5 to 0.7 can better maintain intra-view semantic consistency, enabling the attribute view and modularity view to obtain more reliable structure-aware representations during their respective encoding processes.

As shown in Figure 2b, the community-level contrastive temperature has a more significant influence on model performance. When the temperature is 0.5, ACC, NMI,

ARI, and F1 reach 0.832, 0.718, 0.727, and 0.824, respectively, and all four metrics achieve their highest values. When the temperature exceeds 0.5, the performance shows a clear downward trend. When the temperature increases to 5.0, ACC, NMI, ARI, and F1 decrease to 0.601, 0.568, 0.547, and 0.597, respectively. This result indicates that community-level semantic contrast directly acts on the fused representation Z, and its temperature setting significantly affects the formation of the community-discriminative representation space. A moderate community-level contrastive temperature can effectively pull nodes in the same community closer and push nodes in different communities farther apart, making the fused representation more consistent with the objective of community detection, namely compactness within communities and separation between communities. An overly large temperature makes the similarity distribution too smooth and weakens the constraint effect of ground-truth community labels on the representation space, thereby reducing the clarity of community boundaries.

Figure 3 shows the experimental results for the community-level contrastive weight. When the community-level contrastive weight gradually increases from 0 to 0.6, model performance continues to improve, and ACC, NMI, ARI, and F1 increase from 0.769, 0.646, 0.655, and 0.761 to 0.832, 0.716, 0.725, and 0.824, respectively. When the weight continues to increase, all metrics decline. When the weight is 1.0, ACC, NMI, ARI, and F1 decrease to 0.808, 0.692, 0.701, and 0.800, respectively.

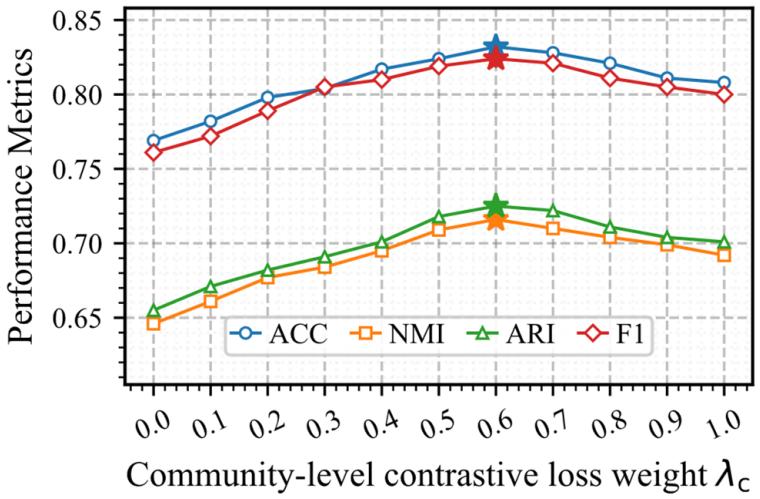


Figure 3. Influence of the community-level contrastive weight on the performance of the TVHCL-CD model. Star markers indicate the highest values of the corresponding performance metrics.

This result indicates that community-level semantic contrast is a key constraint for improving the community discriminability of fused representations. When the weight is too small, the training objective mainly relies on view-level consistency contrast, and the model can learn a certain degree of perturbation stability, but the aggregation of nodes in the same community and the separation of nodes in different communities are insufficient. When the weight is too large, the optimization process is overly biased toward the community-level semantic constraint, weakening the collaborative representation among attribute semantics, modularity priors, and topological dependencies. When the weight is set to 0.6, the view-level consistency constraint and community-level semantic contrast form a better balance, allowing the fused representation to maintain both view stability and community discriminability.

In summary, TVHCL-CD shows good stability with respect to the view-level contrastive temperature, while it is more sensitive to the community-level contrastive temperature and the community-level contrastive weight. When the community-level contrastive

temperature is 0.5, the view-level contrastive temperature lies between 0.5 and 0.7, and the community-level semantic contrastive weight is 0.6, the model can achieve relatively better performance. The above results indicate that hierarchical contrastive learning does not simply rely on the consistency constraint of view augmentation, but enables the fused representation Z to more effectively characterize the community structure in attributed graphs through the synergy between view-level stable representation learning and community-level semantic discriminative optimization.

4.7. Ablation and Label-Availability Analysis

To further analyze the contribution of different components in TVHCL-CD, this section reports three groups of experiments. First, module-level ablation is conducted to examine the effects of the modularity view, the structure-enhanced fusion module, and the two contrastive objectives. Second, fine-grained ablation is conducted on Cora to further evaluate several internal designs in the fusion and augmentation mechanisms. Third, a training label ratio analysis is performed to study the influence of the amount of supervised community information used by the community-level semantic contrastive loss. For fairness, each ablation variant changes only the component under investigation, while the remaining model configuration, data split, training protocol, clustering protocol, and evaluation metrics are kept consistent with those described in Section 4.4.

4.7.1. Module-Level Ablation

To verify the effectiveness of each core module in TVHCL-CD, this study conducts ablation experiments on three datasets, namely Cora, BlogCatalog, and Amazon-Computer, and constructs four variant models. The variant w/o MV removes the modularity view and retains only the attribute view for encoding. The variant w/o DF removes the structure-enhanced Graph Transformer adaptive fusion module and replaces deep cross-view interaction with simple concatenation. The variant w/o View-CL removes the view-level consistency contrastive loss. The variant w/o Comm-SCL removes the community-level semantic contrastive loss. All variants are evaluated under the same semi-supervised transductive setting as the complete model. Therefore, the performance differences mainly reflect the influence of the removed modules rather than differences in data partitioning or evaluation protocol. The ablation results are shown in Figure 4, where Figure 4a, Figure 4b, Figure 4c, and Figure 4d report ACC, NMI, ARI, and Macro-F1, respectively. The complete TVHCL-CD model obtains the best results on ACC, NMI, ARI, and F1 across the three datasets, indicating that the proposed dual-view collaborative modeling, deep cross-view fusion, and hierarchical contrastive learning all make positive contributions to community detection performance.

First, after removing the modularity view, model performance decreases significantly on all three datasets. Taking ACC as an example, w/o MV decreases from 0.832, 0.788, and 0.856 in the complete model to 0.794, 0.731, and 0.814 on Cora, BlogCatalog, and Amazon-Computer, respectively. NMI also decreases from 0.716, 0.602, and 0.668 in the complete model to 0.675, 0.542, and 0.617, respectively. This result indicates that although node attributes and the original topology can provide basic semantic and neighborhood information, they are insufficient to fully characterize community associations between nodes relative to random connection expectations. The modularity view explicitly introduces high-order community structure priors into the representation learning process, helping to enhance structural consistency among nodes in the same community and thereby improving the accuracy of the final clustering partition.

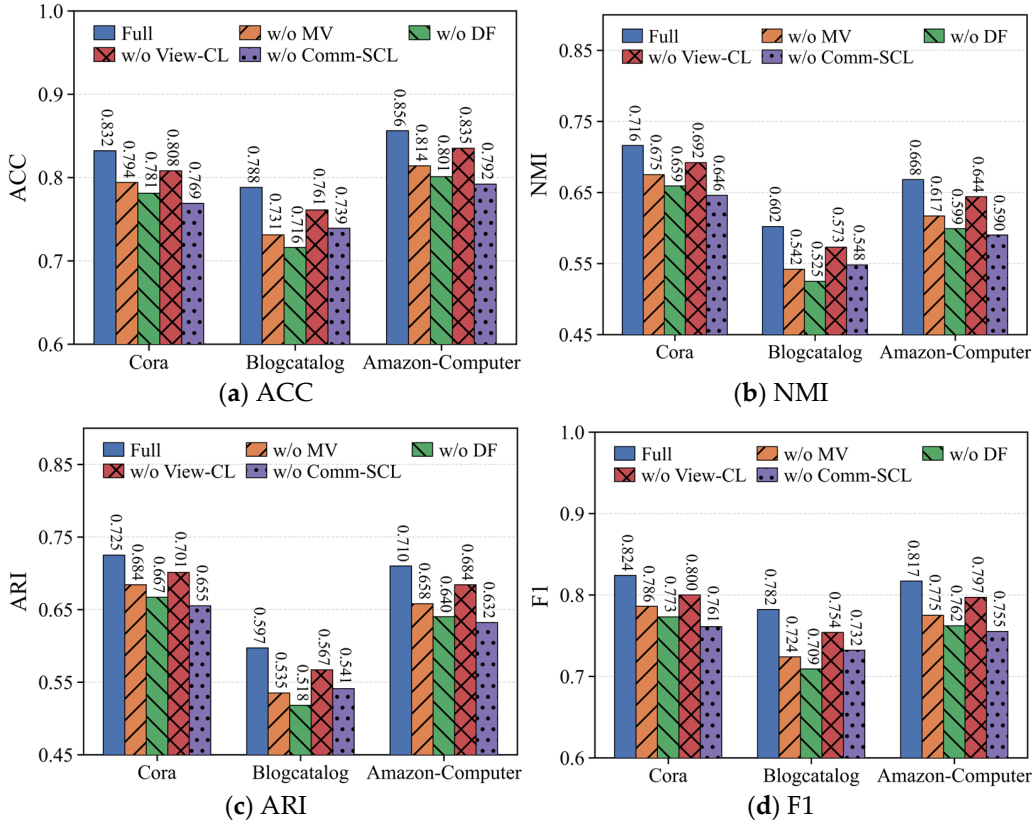


Figure 4. Module-level ablation results of TVHCL-CD on Cora, BlogCatalog, and Amazon-Computer. (**a**) ACC; (**b**) NMI; (**c**) ARI; (**d**) Macro-F1.

Second, *w/o* DF shows a clear performance decrease on all three datasets, and it produces the largest decline among the four module-level ablation variants on BlogCatalog. In particular, on BlogCatalog, ACC, NMI, ARI, and F1 decrease by 7.2, 7.7, 7.9, and 7.3 percentage points, respectively. This shows that relying only on shallow concatenation is insufficient to coordinate the complementary relationship between attribute semantics and modularity structures. The structure-enhanced Graph Transformer preserves local topological constraints through adjacency bias and adaptively adjusts the dependence of different nodes on the two types of view information through the view-aware gating mechanism, enabling the model to avoid the dilution of community semantics during global dependency modeling. Therefore, this module is an important component for improving community discriminability in complex networks.

Finally, the ablation results related to contrastive learning show that view-level consistency contrast and community-level semantic contrast optimize the representation space from the perspectives of stability and discriminability, respectively. After removing View-CL, the model still maintains stable performance, but all metrics are lower than those of the complete model, indicating that the consistency constraint under augmented views can alleviate the influence of attribute noise and structural perturbations. In contrast, after removing Comm-SCL, the performance drops more significantly than removing View-CL, and it produces the largest decreases on Cora and Amazon-Computer among the four module-level ablation variants. For example, ACC decreases by 6.3 and 6.4 percentage points on Cora and Amazon-Computer, respectively, and ARI decreases by 7.0 and 7.8 percentage points, respectively. This indicates that community-level semantic contrast can directly pull nodes in the same community closer and push nodes in different communities farther apart, making the fused representation more consistent with the community detection objective of "intra-community compactness and inter-community separation."

In summary, the performance improvement of TVHCL-CD is not attributable to a single module, but to the joint effect of the modularity view, structure-enhanced fusion, and hierarchical contrastive objectives.

4.7.2. Fine-Grained Ablation

The module-level ablation above verifies the contribution of the main components. To further examine the necessity of several internal designs, this study conducts a fine-grained ablation experiment on Cora. Cora is used because it is a commonly adopted citation-network benchmark and allows controlled comparison under the same model configuration. The evaluated variants include *w/o* adjacency bias, *w/o* view-aware gating, *w/o* Laplacian PE, *w/o* multi-order matrix M, and Random edge dropping. The first three variants analyze the structure-enhanced Graph Transformer fusion module, *w/o* multi-order matrix M analyzes the structural prior used in the two-view encoders, and Random edge dropping replaces the modularity-guided edge dropping strategy with random edge dropping while keeping the edge dropping ratio unchanged. The results are shown in Table 9.

Table 9. Fine-grained ablation results on Cora. Boldface indicates the best result in each column.

Variant	ACC	NMI	ARI	Macro-F1
TVHCL-CD	**0.832**	**0.716**	**0.725**	**0.824**
w/o adjacency bias	0.817	0.700	0.708	0.809
w/o view-aware gating	0.804	0.684	0.696	0.796
w/o Laplacian PE	0.822	0.706	0.713	0.813
w/o multi-order matrix M	0.812	0.693	0.703	0.803
Random edge dropping	0.820	0.701	0.711	0.812

As shown in Table 9, removing any fine-grained design leads to a decrease in all four metrics, indicating that these designs all contribute to the final representation quality. Among the fusion-related variants, *w/o* view-aware gating produces the largest decrease, with ACC, NMI, ARI, and Macro-F1 decreasing from 0.832, 0.716, 0.725, and 0.824 to 0.804, 0.684, 0.696, and 0.796, respectively. This suggests that node-level adaptive regulation of attribute-view and modularity-view information is important for learning community-discriminative fused representations.

The variants *w/o* adjacency bias and *w/o* Laplacian PE also show consistent degradation. Removing the adjacency bias weakens the explicit preservation of local topological constraints during global self-attention, whereas removing Laplacian positional encoding reduces the model's ability to distinguish nodes with similar local features but different structural positions. In comparison, the effect of Laplacian PE is smaller, but its contribution remains positive.

The variant *w/o* multi-order matrix M decreases ACC from 0.832 to 0.812 and NMI from 0.716 to 0.693, showing that multi-hop structural information helps the two-view encoders capture community-related structural consistency beyond first-order adjacency. The Random edge dropping variant also performs worse than the complete model, indicating that modularity-guided edge dropping provides more community-preserving augmented views than purely random perturbation. However, the moderate difference also suggests that random perturbation still provides useful contrastive signals.

Overall, the fine-grained ablation results further support the contributions of the structure-enhanced fusion module, the multi-order structural prior, and the modularity-guided augmentation strategy. These results indicate that the performance of TVHCL-CD comes from coordinated design choices rather than from a single isolated operation.

4.7.3. Training Label Ratio Analysis

Since TVHCL-CD uses partial ground-truth labels in the community-level semantic contrastive loss, it is necessary to analyze how the amount of labeled training nodes affects the model. This experiment fixes the model configuration and changes only the proportion of training labels available to the community-level semantic contrastive loss. The validation and test evaluation protocol remains unchanged. The setting 0%/*w*/*o* Comm-SCL means that no training labels are used in the community-level semantic contrastive loss, so the model relies on the attribute view, modularity view, structure-enhanced fusion, and view-level consistency contrast. The results on Cora are shown in Table 10.

Table 10. Influence of the training label ratio on Cora. Boldface indicates the best result in each column.

Training Label Ratio	ACC	NMI	ARI	Macro-F1
0% (*w*/*o* Comm-SCL)	0.769	0.646	0.655	0.761
5%	0.790	0.668	0.676	0.781
10%	0.807	0.686	0.697	0.798
20%	0.817	0.704	0.711	0.809
30%	0.826	0.710	0.723	0.817
40%	**0.832**	**0.716**	**0.725**	**0.824**

As shown in Table 10, the performance of TVHCL-CD improves as the training label ratio increases from 0% to 40%. When no training labels are used by the community-level semantic contrastive loss, the model still obtains 0.769 ACC, 0.646 NMI, 0.655 ARI, and 0.761 Macro-F1. This result indicates that the unsupervised parts of TVHCL-CD, including the attribute view, modularity view, structure-enhanced fusion, and view-level consistency contrast, can already learn useful community-related representations. However, compared with the 40% label-ratio setting, the 0% setting is clearly lower on all metrics, showing that community-level semantic supervision provides additional discriminative information.

When the training label ratio increases from 0% to 5%, ACC improves from 0.769 to 0.790, and Macro-F1 improves from 0.761 to 0.781. This indicates that even a small amount of labeled training nodes can provide useful same-community and different-community constraints for the fused representation. As the ratio further increases to 10%, 20%, and 30%, the four metrics continue to improve, which suggests that more labeled anchors can help the model form clearer community boundaries in the representation space.

When the training label ratio increases from 30% to 40%, the improvement becomes smaller. ACC increases from 0.826 to 0.832, NMI increases from 0.710 to 0.716, ARI increases from 0.723 to 0.725, and Macro-F1 increases from 0.817 to 0.824. This trend suggests that the benefit of additional labels is not strictly linear. After the model has obtained sufficient community-level semantic constraints, additional labeled nodes still help, but the marginal gain becomes more limited. Therefore, the model benefits from partial supervision, while the results do not indicate that its performance improvement depends only on a large number of labels.

Overall, the label-ratio analysis shows that community-level semantic contrast is useful for improving community-discriminative representations, but TVHCL-CD still retains a certain level of effectiveness without training labels in this loss term. This supports the semi-supervised design of the model: the modularity view and view-level consistency contrast provide a structural and perturbation-stable representation basis, while partial labels further refine the representation space toward intra-community compactness and inter-community separation. Combined with the module-level and fine-grained ablation results, these findings show that the performance of TVHCL-CD comes from the coordi-

nated contribution of structural priors, adaptive fusion, contrastive learning, and limited supervised community information.

4.8. Visualization of the Fused Representation Z

To intuitively evaluate the quality of the fused node representation Z during model training, t-SNE visualization is performed at three representative training stages: epoch 0, epoch 70, and epoch 150. The visualization results are shown in Figure 5 for three representative datasets, where Figure 5a, Figure 5b, and Figure 5c correspond to Cora, ACM, and Citeseer, respectively. In each subfigure, the three panels from left to right represent the distributions of node representations at epoch 0, epoch 70, and epoch 150.

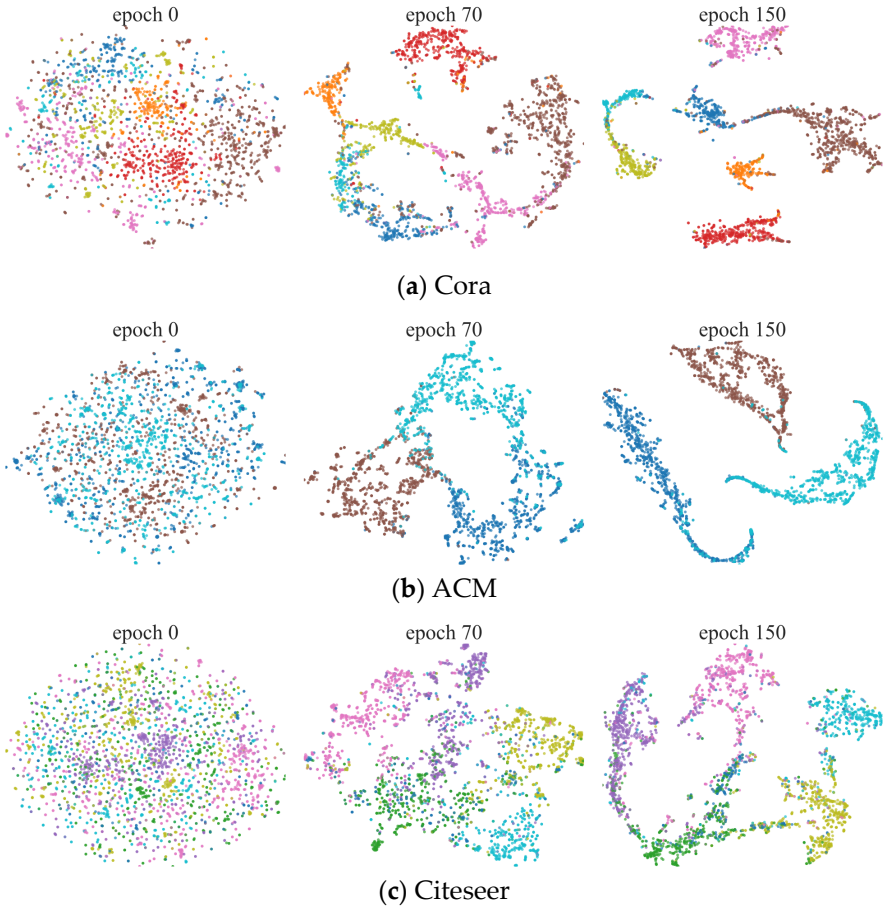


Figure 5. t-SNE visualization of the fused representation Z on three representative datasets at different training stages. (**a**) Cora. (**b**) ACM. (**c**) Citeseer. In each subfigure, the panels from left to right correspond to epoch 0, epoch 70, and epoch 150, respectively, and different colors indicate different ground-truth communities.

At the initial stage of training, the node distribution is relatively scattered and different communities overlap considerably, indicating that the original representations have not yet formed a clear community semantic structure. As training proceeds, nodes from the same community gradually gather into more compact regions, while the boundaries between different communities become clearer. At the final stage, each dataset exhibits a more distinct cluster-like distribution in the low-dimensional space. These observations indicate that view-level consistency contrast improves representation stability, while community-level semantic contrast further enhances intra-community compactness and inter-community separation.

5. Conclusions

To address the difficulty of collaboratively modeling topological structure, node attributes, and high-order community structure priors in attributed graph community detection, this study proposes TVHCL-CD, a two-view hierarchical contrastive learning-driven community detection method. The proposed method constructs an attribute view and a modularity view from the node attribute matrix and the modularity matrix, respectively, to characterize attribute semantics and community structure priors. Graph attention encoders incorporating multi-order neighborhood priors are then used to learn structure-aware two-view representations, thereby helping the model capture local topology and multi-hop relationships. Subsequently, the structure-enhanced Graph Transformer introduces a learnable adjacency bias into global self-attention and combines it with a view-aware gating mechanism to support node-level adaptive fusion while taking both long-range dependencies and local structural constraints into account. A hierarchical contrastive learning strategy is further constructed, in which view-level consistency contrastive learning helps improve perturbation stability, while community-level semantic contrastive learning helps enhance intra-community aggregation and inter-community separation. Experiments on eight real-world attributed graphs and the generated Tree-2500 benchmark show that TVHCL-CD obtains competitive performance under the semi-supervised transductive setting. Specifically, in our experiments, TVHCL-CD obtains the best ACC and Macro-F1 on the tested real-world attributed graphs and competitive NMI/ARI on most datasets, while the Tree-2500 results further show that external label agreement and internal structural modularity may favor different methods. Ablation experiments support the contributions of the modularity view, structure-enhanced fusion, hierarchical contrastive learning, and label-guided community semantic constraints. In future work, we aim to improve computational efficiency on large-scale attributed graphs and explore community semantic constraints under weak-label or label-free conditions.

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

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62566051.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets analyzed in this study are publicly available benchmark datasets. The Cora and Citeseer citation networks were loaded using the Planetoid dataset interface in PyTorch Geometric, and the corresponding Planetoid raw files are publicly available at https://github.com/kimiyoung/planetoid/tree/master/data. The processed benchmark versions of ACM and DBLP are available from the Awesome Deep Graph Clustering repository at https://github.com/yueliu1999/Awesome-Deep-Graph-Clustering/tree/main/dataset. The processed BlogCatalog and Flickr attributed social network datasets are available from the Attributed-Graph-Data repository at https://github.com/he-tiantian/Attributed-Graph-Data. The Amazon-Photo and Amazon-Computer co-purchase networks are available from the GNN benchmark repository used by PyTorch Geometric at https://github.com/shchur/gnn-benchmark/tree/master/data/npz. The above public dataset links were accessed on 6 May 2026. In addition to these public datasets, the Tree-2500 benchmark used in this study was generated according to the procedure described in Section 4.1. The source code for generating Tree-2500, together with the corresponding generation settings and random seed

configuration, is publicly available at https://github.com/ls-aa/tree-2500/blob/main/tree-2500.py. The Tree-2500 link was accessed on 4 June 2026. The modularity matrices, augmented views, processed inputs, train/validation/test split files, random seed settings, and experimental result records were generated from the above public datasets according to the procedures described in the manuscript and are available from the corresponding author upon reasonable request.

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

References

1. Li, M.; Yang, Z.; Zhou, X.; Fang, Y.; Li, K.; Li, K. Clustering on attributed graphs: From single-view to multi-view. *ACM Comput. Surv.* **2025**, *57*, 1–36. [CrossRef]
2. Liang, Y.; Shu, J.; Liu, L. Attributed network community detection based on graph contrastive learning and multi-objective evolutionary algorithm. *Neurocomputing* **2025**, *636*, 130029. [CrossRef]
3. He, C.; Cheng, J.; Chen, G.; Guan, Q.; Fei, X.; Tang, Y. Detecting communities with multiple topics in attributed networks via self-supervised adaptive graph convolutional network. *Inf. Fusion* **2024**, *105*, 102254. [CrossRef]
4. Bagrow, J.P. Communities and bottlenecks: Trees and treelike networks have high modularity. *Phys. Rev. E* **2012**, *85*, 066118. [CrossRef]
5. Jia, C.; Li, Y.; Carson, M.B.; Wang, X.; Yu, J. Node attribute-enhanced community detection in complex networks. *Sci. Rep.* **2017**, *7*, 2626. [CrossRef] [PubMed]
6. de Silva, A.; Chen, G.; Ma, H.; Nekooei, S.M.; Zuo, X. Advancing community detection with graph convolutional neural networks: Bridging topological and attributive cohesion. In Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25, Montreal, QC, Canada, 16–22 August 2025; pp. 7383–7391. [CrossRef]
7. Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; Zhang, C. Attributed graph clustering: A deep attentional embedding approach. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, Macao, China, 10–16 August 2019; pp. 3670–3676. [CrossRef]
8. Newman, M.E.J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [CrossRef]
9. Wu, X.; Lu, W.; Quan, Y.; Miao, Q.; Sun, P.G. Deep dual graph attention auto-encoder for community detection. *Expert Syst. Appl.* **2024**, *238*, 122182. [CrossRef]
10. Bhowmick, A.; Kosan, M.; Huang, Z.; Singh, A.; Medya, S. DGCLUSTER: A neural framework for attributed graph clustering via modularity maximization. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 11069–11077. [CrossRef]
11. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Graph contrastive learning with adaptive augmentation. In Proceedings of the Web Conference 2021, WWW '21, Ljubljana, Slovenia, 19–23 April 2021; pp. 2069–2080. [CrossRef]
12. Hassani, K.; Khasahmadi, A.H. Contrastive multi-view representation learning on graphs. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Virtual Event, 13–18 July 2020.
13. Liu, Y.; Li, J.; Chen, Y.; Wu, R.; Wang, E.; Zhou, J.; Tian, S.; Shen, S.; Fu, X.; Meng, C.; et al. Revisiting modularity maximization for graph clustering: A contrastive learning perspective. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, Barcelona, Spain, 25–29 August 2024; pp. 1968–1979. [CrossRef]
14. Huang, Z.; Xu, W.; Zhuo, X. Community-CL: An enhanced community detection algorithm based on contrastive learning. *Entropy* **2023**, *25*, 864. [CrossRef]
15. Newman, M.E.J.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. [CrossRef] [PubMed]
16. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
17. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA, 24–27 August 2014; pp. 701–710. [CrossRef]
18. Ran, Y.; Yi, J.; Si, W.; Small, M.; Shang, K.-K. Machine Learning Informed by Micro- and Mesoscopic Statistical Physics Methods for Community Detection. *Chaos* **2025**, *35*, 073103. [CrossRef]
19. Berahmand, K.; Bahadori, S.; Abadeh, M.N.; Li, Y.; Xu, Y. SDAC-DA: Semi-supervised deep attributed clustering using dual autoencoder. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 6989–7002. [CrossRef]
20. Zhou, L.; Du, G.; Lü, K.; Wang, L.; Du, J. A survey and an empirical evaluation of multi-view clustering approaches. *ACM Comput. Surv.* **2024**, *56*, 1–38. [CrossRef]
21. Xiao, S.; Li, J.; Lu, J.; Huang, S.; Zeng, B.; Wang, S. Graph neural networks for multi-view learning: A taxonomic review. *Artif. Intell. Rev.* **2024**, *57*, 341. [CrossRef]
22. Wang, X.; Zhu, M.; Bo, D.; Cui, P.; Shi, C.; Pei, J. AM-GCN: Adaptive multi-channel graph convolutional networks. In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '20, Virtual Event, CA, USA, 23–27 August 2020; pp. 1243–1253. [CrossRef]

23. Pan, E.; Kang, Z. Multi-view contrastive graph clustering. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 2148–2159.
24. Chen, J.; Ling, Y.; Xu, J.; Ren, Y.; Huang, S.; Pu, X.; Hao, Z.; Yu, P.S.; He, L. Variational graph generator for multiview graph clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 11078–11091. [CrossRef]
25. He, L.; Cheng, D.; Zhang, G.; Zhang, X. Leveraging long-range nodes in multi-view graph contrastive learning. *Inf. Fusion* **2025**, *122*, 103186. [CrossRef]
26. Wu, Q.; Zhao, W.; Yang, C.; Zhang, H.; Nie, F.; Jiang, H.; Bian, Y.; Yan, J. SGFormer: Simplifying and empowering transformers for large-graph representations. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 64753–64773.
27. Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.Y. Do transformers really perform badly for graph representation? *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 28877–28888.
28. Duan, Y.; Liu, J.; Wu, J.; Liu, J. Community-aware graph Transformer: Preserving community semantics for effective global aggregation. In *Machine Learning and Knowledge Discovery in Databases. Research Track, ECML PKDD 2025*; Springer: Cham, Switzerland, 2025; pp. 108–124. [CrossRef]
29. Zhang, W.; Zhu, X.; Shang, R.; Ren, J.; Jiao, L. Adaptive graph clustering based on feature adversarial and graph Transformer. *Expert Syst. Appl.* **2025**, *281*, 127541. [CrossRef]
30. Wang, Q.; Xu, H.; Zhang, Z.; Feng, W.; Gao, Q. Deep multi-modal graph clustering via graph Transformer network. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 7835–7843. [CrossRef]
31. Wen, Q.; Zhang, Y.; Ye, Y.; Zhou, Y.; Zhang, N.; Lian, X.; Chen, M. GCLS2: Towards Efficient Community Detection Using Graph Contrastive Learning with Structure Semantics. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management, Seoul, Republic of Korea, 10–14 November 2025*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 3292–3301. [CrossRef]
32. Veličković, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep Graph Infomax. In Proceedings of the International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019. Available online: https://openreview.net/forum?id=rklz9iAcKQ (accessed on 7 May 2026).
33. Yan, X.; Deng, K.; Zou, Q.; Tian, Z.; Yu, H. Self-cumulative contrastive graph clustering. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 1194–1208. [CrossRef]
34. Xia, W.; Wang, Q.; Gao, Q.; Yang, M.; Gao, X. Self-consistent contrastive attributed graph clustering with pseudo-label prompt. *IEEE Trans. Multimed.* **2023**, *25*, 6665–6677. [CrossRef]
35. Zheng, Y.; Jia, C.; Yu, J. Attributed graph clustering under the contrastive mechanism with cluster-preserving augmentation. *Inf. Sci.* **2024**, *681*, 121225. [CrossRef]
36. Yang, X.; Tan, C.; Liu, Y.; Liang, K.; Wang, S.; Zhou, S.; Xia, J.; Li, S.Z.; Liu, X.; Zhu, E. CONVERT: Contrastive graph clustering with reliable augmentation. In Proceedings of the 31st ACM International Conference on Multimedia, MM '23, Ottawa, ON, Canada, 29 October–3 November 2023; pp. 319–327. [CrossRef]
37. Yang, X.; Liu, Y.; Zhou, S.; Wang, S.; Tu, W.; Zheng, Q.; Liu, X.; Fang, L.; Zhu, E. Cluster-guided contrastive graph clustering network. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 10834–10842. [CrossRef]
38. Li, B.; Jing, B.; Tong, H. Graph Communal Contrastive Learning. In Proceedings of the ACM Web Conference 2022, Virtual Event, Lyon, France, 25–29 April 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 1203–1213. [CrossRef]
39. Chen, H.; Zhao, Z.; Li, Y.; Zou, Y.; Li, R.; Zhang, R. CSGCL: Community-Strength-Enhanced Graph Contrastive Learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, Macao, China, 19–25 August 2023*; International Joint Conferences on Artificial Intelligence Organization: Montpellier, France, 2023; pp. 2059–2067. [CrossRef]
40. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017. Available online: https://openreview.net/forum?id=SJU4ayYgl (accessed on 7 May 2026).
41. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous graph attention network. In Proceedings of the World Wide Web Conference, WWW '19, San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032. [CrossRef]
42. Meng, Z.; Liang, S.; Bao, H.; Zhang, X. Co-embedding attributed networks. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM '19, Melbourne, VIC, Australia, 11–15 February 2019; pp. 393–401. [CrossRef]
43. Shchur, O.; Mumme, M.; Bojchevski, A.; Günnemann, S. Pitfalls of graph neural network evaluation. *arXiv* **2018**, arXiv:1811.05868. [CrossRef]
44. Shang, K.-K.; Li, T.-C.; Small, M.; Burton, D.; Wang, Y. Link prediction for tree-like networks. *Chaos* **2019**, *29*, 061103. [CrossRef] [PubMed]
45. Munkres, J. Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32–38. [CrossRef]

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