OPTIMAL TRANSPORT GUIDED DUAL-DISTRIBUTION CONSTRAINTS FOR MULTI-VIEW GRAPH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-view graph clustering aims to group graph nodes by jointly exploiting multi-view node attributes and topological connections. However, existing methods lack reliable global supervision for cross-view representation alignment. Under severe inter-view noise and heterogeneity, representations of each view converge to distinct local optima and yield weak cross-view consensus. Furthermore, most works only regularize node features while neglecting graph topological neighborhood distributions, decoupling graph smoothing from clustering objectives. To address these issues, we propose a novel **O**ptimal **T**ransport Guided **D**ual-**D**istribution **C**onstraints for Multi-view Graph Clustering (OT-DDC). It employs an optimal transport-based pseudo-label generation module to construct global cross-view self-supervision and alleviate weak cross-view consensus. Meanwhile, the dual-distribution constraint clustering module jointly optimizes cluster centroids in the feature space and topology-aware neighborhood distribution centers, reducing misalignment between semantic representations and graph topological structures. Extensive experiments demonstrate the effectiveness of our algorithm.

1 INTRODUCTION

Clustering, a fundamental unsupervised learning technique, aims to partition data samples into distinct groups based on their intrinsic similarities. It has been extensively applied in diverse domains including social network analysis (Sen et al., 2008) and bioinformatics (Xu et al., 2021). Traditional clustering methods, such as K-Means (MacQueen, 1967) and spectral clustering (Hubert & Arabie, 1985), perform well on Euclidean structured data, but struggle to deal with graph data with non-Euclidean structures, such as social networks and knowledge graphs. Graph data contains rich node attributes and complex topological connections, requiring models to capture both feature similarity and structural proximity simultaneously.

Graph neural networks (GNNs) (Kipf & Welling, 2017; Scarselli et al., 2009), as powerful tools specifically designed for graph data, aggregate neighborhood information through message-passing mechanisms to learn node representations effectively. Deep graph clustering methods based on GNNs directly use low-dimensional embeddings learned by encoders for clustering and have achieved significant progress. For example, Graph Auto-Encoders (GAE) and Variational Graph Auto-Encoders (VGAE) (Kipf & Welling, 2016) pioneer the integration of autoencoder frameworks and Graph Convolutional Networks (GCNs) for unsupervised representation learning. Deep Attentional Embedding Graph Clustering (DAEGC) (Wang et al., 2019) introduces attention mechanism to assign differentiated weights to neighbor contributions. Nevertheless, these methods are largely designed for single-view graphs where each node possesses only one type of feature vector. In reality, entities in the real world are usually associated with multiple heterogeneous information sources. Although such multi-view property facilitates more informative node representations, it poses significant challenges for effectively fusing and harnessing the complementary potential of diverse data streams (Wang et al., 2020).

Recent studies have therefore increasingly investigated multi-view and contrastive graph clustering. Methods such as MAGCN (Cheng et al., 2020), SGCMC (Xia et al., 2022b), and CO-GCN (Li et al., 2020) integrate information from multiple graph views or jointly learn attribute and structural representations. Contrastive graph clustering methods further construct node-level, neighborhood-level, or cross-view consistency objectives to improve representation discrimination and robustness. For

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

example, NCLA (Shen et al., 2023) introduces learnable graph augmentations for neighbor contrastive learning, while CMGEC (Wang et al., 2023b) and its related approaches exploit cross-view consistency for multi-view graph representation learning. More recently, graph Transformer (Wang et al., 2025) and multi-view graph clustering methods have been developed to capture long-range dependencies and semantic relationships in large graphs. These developments provide important foundations for combining cross-view representation learning, globally constrained pseudo-labeling, and topology-aware clustering. However, existing methods mainly rely on local pairwise contrastive learning or independently generated pseudo-labels. Without a global node-to-prototype assignment mechanism, they remain vulnerable to noisy, locally optimal, and unbalanced cluster assignments. Furthermore, most graph clustering methods learn cluster centroids only in the feature space while overlooking neighborhood distributions, which may lead to assignments that are inconsistent with the underlying graph structures.

To address these challenges, we propose a novel framework called Optimal Transport Guided Dual-Distribution Constraints for Multi-view Graph Clustering, named OT-DDC, whose framework is shown in Figure 1. The framework constructs two complementary graph views via Laplacian smoothing and frequency-domain transformation to capture both local smoothness and global structural patterns. We then employ cross-view consistency learning to align node representations across views in a shared semantic space. An entropy-regularized optimal transport module generates globally balanced cross-view soft pseudo-labels, and the dual-distribution constraint module jointly exploits feature-space constraint and neighborhood-distribution constraint. The main contributions of our work are summarized as follows:

- We propose Optimal Transport Guided Dual-Distribution Constraints for Multi-view Graph Clustering, termed OT-DDC, which integrates cross-view optimal transport pseudo-labeling with dual-distribution constraints.
- We develop an entropy-regularized optimal transport module to establish globally balanced node-to-prototype assignments and generate reliable soft pseudo-labels to address weak cross-view consensus.
- Through the dual-distribution constraint mechanism, our approach effectively bridges representation learning with explicit cluster structure modeling, jointly optimizing cluster centroids and topology-aware neighborhood distribution centers.
- Extensive experiments on four datasets have demonstrated the superiority of the proposed method.

2 Related Work

2.1 Multi-View Graph Clustering

Multi-view graph clustering aims to exploit complementary information from both node attributes and graph topology for improved community detection. Traditional methods typically adopt spectral clustering frameworks (Zhu et al., 2019) or matrix factorization techniques (MacQueen, 1967), focusing on designing fusion strategies for multiple similarity matrices. Wang et al. (2023a) perform joint decomposition of affinity matrices derived from different views. Nie et al. (2017) learn adaptive weights to balance their contributions. Despite their theoretical soundness, these approaches predominantly rely on handcrafted features and shallow linear models, limiting their capacity to capture complex nonlinear relationships inherent in real-world graphs.

2.2 Deep Multi-View Graph Clustering

Deep graph clustering methods typically combine Graph Neural Networks (GNNs) with clustering algorithms to learn clustering-friendly embeddings in an end-to-end manner. Based on the source of supervision signals, they can be categorized into reconstruction-based, self-supervised, and jointly optimized methods. Specifically, reconstruction-based methods learn node representations through graph autoencoders and preserve topological and attribute information using reconstruction loss (Kipf & Welling, 2016). Self-supervised approaches construct supervisory signals at the node or subgraph level via contrastive learning (Xia et al., 2022a). Jointly optimized methods dynamically update cluster centers during training and align node representations with cluster assignments using

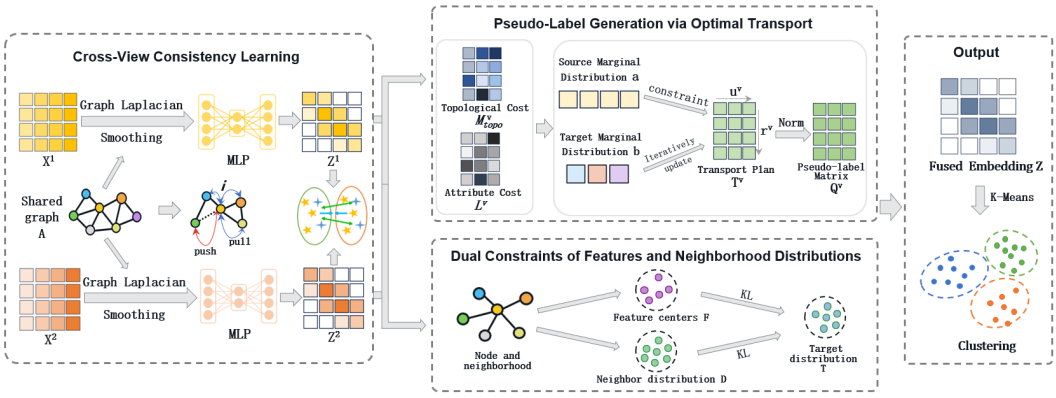


Figure 1: Overview of the proposed OT-DDC framework. Given two complementary graph views $\mathbf{X}^1$ and $\mathbf{X}^2$ sharing the graph structure $\mathbf{A}$, graph Laplacian smoothing and view-specific MLP encoders are first employed to obtain the node embeddings $\mathbf{Z}^1$ and $\mathbf{Z}^2$. Cross-view consistency learning is applied to enhance consistency between views. The optimal transport module constructs globally balanced soft pseudo-labels $\mathbf{Q}^v$. The dual-distribution constraint module jointly modele feature-space cluster centers and neighborhood distribution centers. Finally, the fused embedding $\mathbf{Z}$ is clustered using K-Means.

techniques such as Student's t-distribution or KL-divergence minimization (Bo et al., 2020). However, most existing methods rely solely on a single clustering center in the feature space and fail to explicitly model the neighbor consensus information implied by the graph topology as a parallel clustering constraint center. Our dual-constraint mechanism is designed to bridge this gap, achieving dual-drive integration of feature and topological information.

3 METHODOLOGY

This section begins by defining the symbols and clarifying the problem statement. It then introduces the model components, including cross-view consistency learning, pseudo-label generation via optimal transport, dual-distribution constraints. The overall framework is illustrated in Fig 1. Detailed descriptions of these modules will be provided in the subsequent sections.

3.1 NOTATIONS AND PROBLEM DEFINITION

Consider a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ represents *Vertices* and $\mathbf{E}$ represents *Edges*. The topological structure of the graph $\mathbf{G}$ is typically represented by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$. $\mathbf{D} \in \mathbb{R}^{n\times n}$ denotes the degree matrix corresponding to $\mathbf{A}$, and the Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Owing to the nodes in graph $\mathbf{G}$ exhibiting feature attributes from different views, these attributes collectively form a multi-view node attribute set, denoted as $\mathcal{X} = \left\{\mathbf{X}^v \in \mathbb{R}^{n\times d}\right\}_{v=1}^{V}$, where $\mathbf{X}^v$ represents the node attribute matrix for the v-th view with n nodes and d features. The model aims to obtain a low-dimensional latent embedding $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\} \in \mathbb{R}^{n\times d}$ suitable for clustering tasks that partition the nodes into k clusters.

3.2 CROSS-VIEW CONSISTENCY LEARNING

To effectively learn node representations in graph-based data, we apply a widely-used Laplacian filter to reduce high-frequency noise in the node attributes, while integrating attribute information with graph structure. This process is formulated as follows:

$$\tilde{\mathbf{X}}^v = \left(\mathbf{I}_N - k\tilde{\mathbf{L}}\right)^t \mathbf{X}^v, \tag{1}$$

where k denotes a real-value parameter, and t specifies the number of filtering layers. In our method, we set $k = 1$, transforming the filter into a GCN filter. $\tilde{\mathbf{X}}^v$ indicates the resulting smoothed attribute matrix. Subsequently, node representations are derived by employing multi-layer perceptrons

(MLPs) with shared parameters, which is expressed as:

$$\mathbf{Z}^v = \mathrm{MLP}\big(\tilde{\mathbf{X}}^v\big), \tag{2}$$

Topology-Guided Contrastive Learning To optimize the consistency between the node similarity and the graph topological structure, we adopt the contrastive loss as the optimization objective. We define the node pairs with adjacency connections as positive pairs and the others as negative pairs, which pulls close the positive pairs and pushes away the negative pairs. The contrastive loss function is defined as:

$$\mathcal{L}_{\mathrm{c}} = -\frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \log\left(\sigma\left(\mathbf{S}_{i,j}\right)\right) - \frac{1}{|\mathcal{N}|}\sum_{(i,j)\in\mathcal{N}} \log\left(\sigma\left(-\mathbf{S}_{i,j}\right)\right), \tag{3}$$

where $\mathbf{S} \in \mathbb{R}^{N\times N}$ is the node similarity matrix, and $\mathbf{S}_{i,j} = \mathbf{Z}_i^1 \left(\mathbf{Z}_j^2\right)^T$ denotes the similarity score between node i and node j. The set of positive node pairs $\mathcal{P} = \{(i,j) \mid \mathbf{A}_{i,j} > 0\}$ represents pairs of nodes that are connected by an edge in the graph, $|\mathcal{P}|$ is the number of such positive pairs. $\mathcal{N} = \{(i,j) \mid \mathbf{A}_{i,j} = 0\}$ represents pairs of nodes that have no edge connection in the graph, and $|\mathcal{N}|$ is the number of such negative pairs. This loss function ensures that the learned node embeddings not only accurately reflect the topological relationships of the graph but also maintain consistency and discriminability across different view transformations.

Cross-View Node Consistency Given the node embeddings $\mathbf{Z}^1$ and $\mathbf{Z}^2$ of two views, we employ a cross-view node-level contrastive objective to enhance the semantic consistency between the two views. Specifically, the representations of the same node in different views are treated as positive pairs, while the representations of different nodes are treated as negative pairs. The cross-view node contrastive loss is formulated as

$$\mathcal{L}_{\mathrm{n}} = -\frac{1}{2N}\sum_{i=1}^{2N} \log \frac{\exp\left(s_{i,p(i)}/\tau_{\mathrm{n}}\right)}{\sum_{j\neq i}\exp\left(s_{ij}/\tau_{\mathrm{n}}\right)}, \tag{4}$$

where $p(i)$ denotes the index of the positive counterpart of node i from the other view, τ_{n} is the temperature parameter. This objective pulls the representations of the same node across two views closer while pushing the representations of different nodes apart.

3.3 Globally Balanced Pseudo-Label Generation via Optimal Transport

In this section, we present an optimal-transport-based module to produce globally balanced soft pseudo-labels. Instead of assigning nodes independently by prototype similarity, our method constructs a global probabilistic matching between nodes and learnable semantic prototypes under the entropy-regularized optimal transport framework. All nodes form the source distribution, and cluster prototypes serve as the target distribution.We construct the cost matrix by integrating node-prototype feature similarity and graph topological degree information. The solved optimal transport matrix globally assigns all nodes to clusters in a single run to minimize the overall assignment cost, yielding balanced cross-view soft pseudo-labels.

For the v-th view, the attribute cost $\mathbf{M}_{\text{attr}}^v \in \mathbb{R}^{n\times k}$ is based on the original similarity between nodes and prototypes, and is defined as follows:

$$\mathbf{M}_{\text{attr}}^v = -\mathbf{L}^v, \quad \mathbf{L}^v = \mathbf{Z}^v\mathbf{P}^T, \tag{5}$$

where $\mathbf{P} \in \mathbb{R}^{k\times d}$ is a set of learnable semantic prototypes. $\mathbf{L}^v$ is the similarity matrix between nodes and prototypes, $\mathbf{L}_{i,j}^v$ represents the probability that node i belongs to prototype j in the v-th view. To prevent optimal transport (OT) from performing assignments solely based on attribute similarity, the cost matrix further introduces degree discrepancy to measure the compatibility between each node and prototype, such that nodes with high connectivity tend to match prototypes with similar local structural roles. The topological cost matrix $\mathbf{M}_{\text{topo}}^v \in \mathbb{R}^{n\times k}$ is calculated as:

$$\mathbf{M}_{\text{topo}}^v\left(i,j\right) = \left\|\deg_v(i) - \deg_{\text{proto}}(j)\right\|^2, \tag{6}$$

where $\deg_v(i)$ is the normalized degree of node i in the v-th view, which reflects the connection strength of the node in the topological structure of the current view. $\deg_{\text{proto}}(j)$ is the normalized

degree of prototype j, which reflects the connection strength of the prototype in its topological structure. If the normalized degrees of node i and prototype j are closer, the value of $\mathbf{M}^v_{\text{topo}}[i,j]$ becomes smaller, indicating better topological structure matching between the two, conversely the topological matching cost increases. The attribute-level and topology-level costs are fused as

$$\mathbf{M}^v = (1-\alpha_{ot})\mathbf{M}^v_{\text{attr}} + \alpha_{ot}\mathbf{M}^v_{\text{topo}}, \tag{7}$$

where $\alpha_{ot} \in [0,1]$ controls the contribution of the topological cost, and is typically set to 0.5. Let $\mathbf{T}^v \in \mathbb{R}_+^{n\times k}$ denote the transport plan between the nodes and prototypes. We define the source marginal distribution uniformly over all nodes as $\mathbf{a} = \frac{1}{N}\mathbf{1}_N$ and each node is assigned an equal probability mass. The target marginal distribution $\mathbf{b} = \frac{\mathbf{w}}{\mathbf{1}^\top\mathbf{w}}$ corresponds to the cluster prior probability of semantic prototypes, where $\mathbf{w}$ is adaptively updated throughout model training. The entropy-regularized optimal transport problem is formulated as

$$\mathbf{T}^v = \underset{\mathbf{T}\geq 0}{\arg\min}\,\langle \mathbf{T}, \mathbf{M}^v\rangle - \varepsilon_{\text{ot}} H(\mathbf{T}) \quad \text{s.t.} \quad \mathbf{T}\mathbf{1}_K = \mathbf{a}, \quad \mathbf{T}^\top\mathbf{1}_N = \mathbf{b}, \tag{8}$$

where ε_{ot} is a hyperparameter that balances the matching cost and regularization smoothness. The marginal constraints enforce the overall assignment probability distributed to each prototype to conform to the predefined cluster prior, thereby achieving globally coordinated node clustering. This mechanism effectively alleviates the severe cluster imbalance issue caused by independent local assignment. To efficiently solve the entropy-regularized optimal transport problem, we adopt the iterative Sinkhorn normalization algorithm. We first construct the Gibbs kernel $\mathbf{K}^v = \exp\left(-\frac{\mathbf{M}^v}{\varepsilon_{\text{ot}}}\right)$ based on the view-specific cost matrix $\mathbf{M}^v$. Two scaling vectors $\mathbf{u}^v$ and $\mathbf{r}^v$ are iteratively updated to satisfy the marginal constraints:

$$\mathbf{u}^v \leftarrow \frac{\mathbf{a}}{\mathbf{K}^v\mathbf{r}^v + \epsilon_0}, \quad \mathbf{r}^v \leftarrow \frac{\mathbf{b}}{(\mathbf{K}^v)^\top\mathbf{u}^v + \epsilon_0}, \tag{9}$$

After a fixed number of iterations, the optimized transport plan is reconstructed by the dual scaling vectors:

$$\mathbf{T}^v = \text{diag}(\mathbf{u}^v)\mathbf{K}^v\,\text{diag}(\mathbf{r}^v), \tag{10}$$

Finally, we perform row normalization on the obtained transport plan to generate the view-aware soft pseudo-label matrix:

$$\mathbf{Q}^v = \frac{\mathbf{T}^v}{\mathbf{T}^v\mathbf{1}_{K_p} + \epsilon_0}. \tag{11}$$

Each row $\mathbf{q}_i^v$ of $\mathbf{Q}^v$ represents the soft probabilistic cluster assignment distribution of node i under view v. Building upon this, we compute the cross-view swap prediction loss to drive self-supervised learning. We use the soft assignment matrix of one view as a pseudo-label to supervise the prototype similarity logits of the other view. The final loss is given by:

$$\mathcal{L}_{\text{s}} = \frac{1}{2}\Big[\text{KL}\left(\mathbf{Q^1}\|\mathbf{L^2}\right) + \text{KL}\left(\mathbf{Q^2}\|\mathbf{L^1}\right)\Big], \tag{12}$$

3.4 Dual Constraints of Features and Neighborhood Distributions

Although the learned node embeddings preserve semantic information, they may still fail to fully reflect the topological community structure of the graph. Hence, we introduce the dual-distribution constraint method that collaboratively refines feature-space cluster centers and topology-aware neighborhood distribution centers, thereby constructing stronger clustering supervision signals. Specifically, we firstly fuse the latent embeddings from the two views, enabling the resultant latent embeddings to be utilized for subsequent optimization:

$$\boldsymbol{Z} = \frac{1}{2}\big(\boldsymbol{Z}^1 + \boldsymbol{Z}^2\big), \tag{13}$$

Constraint of Features We first utilize the embedding $\boldsymbol{Z}$ to generate the target distribution. The probability f_{ij} that node i belongs to the cluster j is calculated :

$$f_{ik} = \frac{\big(1 + \|\boldsymbol{z}_i - \boldsymbol{\mu}_k\|^2/\theta\big)^{-\frac{\theta+1}{2}}}{\sum_k \big(1 + \|\boldsymbol{z}_i - \boldsymbol{\mu}_k\|^2/\theta\big)^{-\frac{\theta+1}{2}}}, \tag{14}$$

where $\boldsymbol{\mu}_k$ denotes the k-th clustering center, which is dynamically updated via an online K-means algorithm. θ is the temperature parameter. In our experiment, we set θ to 1. To provide clear optimization direction and prevent degenerate solutions, we apply a sharpening operation to the feature-space assignments to obtain the target distribution:

$$t_{ik} = \frac{f_{ik}^2 / \sum_i f_{ik}}{\sum_j \left(f_{ij}^2 / \sum_j f_{ij} \right)}, \tag{15}$$

The feature-space constraint encourages the current cluster assignments to approach the sharpened target distribution, which improves the compactness and discriminability of clusters in the feature space. The feature constraint loss function is calculated as follows:

$$\mathcal{L}_{\text{feat}} = \mathrm{KL}\left(\mathbf{T} \| \mathbf{F}\right), \tag{16}$$

Constraint of Neighborhood Distributions The feature-based assignment alone may not fully reflect the community structure of the graph. The probability that the neighborhood distribution of node i belongs to cluster k is computed as:

$$d_{ik} = \frac{\left(1 + \|\boldsymbol{e}_i - \boldsymbol{c}_k\|^2 / \theta\right)^{-\frac{\theta+1}{2}}}{\sum_k \left(1 + \|\boldsymbol{e}_i - \boldsymbol{c}_k\|^2 / \theta\right)^{-\frac{\theta+1}{2}}}, \mathbf{C} = \mathrm{Norm}\left(\mathbf{Y}^\top \mathbf{A}\mathbf{Y}\right) \tag{17}$$

where $\mathbf{E} = \mathbf{A}\mathbf{F}$ is the neighbor distribution $\boldsymbol{e}_i$ of node i. $\mathbf{C}$ stands for neighborhood-distribution centers and $\mathbf{Y}$ is the current hard cluster assignment matrix. The neighborhood-distribution constraint uses the same target distribution to align the topological assignments with the feature-space clustering structure:

$$\mathcal{L}_{\text{nei}} = \mathrm{KL}\left(\mathbf{T} \| \mathbf{D}\right), \tag{18}$$

This constraint encourages nodes with similar neighborhood cluster distributions to receive consistent cluster assignments, thereby improving the structural coherence of the learned clusters. The two constraints are jointly optimized to obtain the dual-constraint objective:

$$\mathcal{L}_{\text{d}} = \mathcal{L}_{\text{feat}} + \mathcal{L}_{\text{nei}}, \tag{19}$$

3.5 Joint Optimization

The total loss function is eventually formulated as:

$$\mathcal{L} = \mathcal{L}_s + \alpha \mathcal{L}_c + \beta \mathcal{L}_n + \gamma \mathcal{L}_{\text{d}}, \tag{20}$$

where α, β and γ are hyperparameters. After training, clustering is performed by applying the k-means algorithm to the fused representations Z.

4 Experimental Analysis

4.1 Experimental Setting

Databases and Metrics We evaluated our approach on four datasets: AMAP (Shchur et al., 2019), WIKI (Mernyei & Cangea, 2022), Citeseer (Yang et al., 2016), and Cora (McCallum et al., 2000). Table 1 summarizes the dataset statistics. We adopt three widely recognized evaluation metrics to quantify the clustering effectiveness of all compared methods: Accuracy (ACC) for measuring the overall clustering correctness, Normalized Mutual Information (NMI) for evaluating the similarity between clustering results and ground-truth labels, and Adjusted Rand Index (ARI) for assessing the correlation between predicted clusters and real category partitions while correcting for random chance.

Implementation Details In this study, all experiments were performed on a Windows server equipped with an NVIDIA GeForce RTX 4090 graphics card with driver version 552.41 and CUDA version 12.4. The experimental environment was based on the Python platform using the PyTorch framework (Python version 3.10.13) and the MATLAB experimental simulation software to realize the proposed method and its comparison with existing data clustering methods.

Comparison of Algorithms To evaluate the performance of our proposed method, we compared it with SOTA clustering algorithms, including ten classical deep graph clustering methods and five

Table 1: Dataset statistics

Dataset	Dimension	Classes	Nodes	Edges
AMAP	745	8	7650	119081
WIKI	4973	17	2405	8261
CITESEER	3327	6	3327	4732
CORA	1433	7	2708	5429

Table 2: Performance comparison of different methods on four datasets.

METHOD	AMAP			WIKI			CITESEER			CORA		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means	0.280	0.329	0.059	0.439	0.384	0.152	0.544	0.312	0.285	0.500	0.317	0.239
GAE	0.741	0.649	0.534	0.332	0.320	0.151	0.380	0.174	0.141	0.530	0.397	0.293
VGAE	0.757	0.646	0.556	0.493	0.461	0.299	0.392	0.163	0.101	0.592	0.408	0.347
DAEGC	0.762	0.657	0.584	0.521	0.432	0.337	0.672	0.397	0.410	0.704	0.528	0.496
GATE	0.696	0.611	0.473	0.482	0.343	0.188	0.616	0.401	0.381	0.658	0.527	0.451
SDCN	0.543	0.457	0.324	0.432	0.396	0.227	0.663	0.390	0.406	0.495	0.275	0.215
EGAE	0.762	0.652	0.573	0.515	0.480	0.331	0.674	0.412	0.432	0.724	0.540	0.472
DCCA	0.382	0.285	0.147	0.358	0.366	0.170	0.450	0.221	0.204	0.436	0.214	0.160
DCCAE	0.475	0.332	0.235	0.321	0.343	0.156	0.503	0.240	0.211	0.472	0.289	0.221
CO-GCN	0.682	0.622	0.511	0.515	0.450	0.330	0.655	0.432	0.423	0.735	0.567	0.512
NCLA	0.679	0.647	0.479	0.508	0.454	0.349	0.616	0.376	0.339	0.523	0.326	0.383
SGCMC	0.734	0.642	0.571	0.478	0.436	0.275	0.612	0.336	0.366	0.658	0.502	0.382
CMGEC	0.751	0.639	0.565	0.557	0.496	0.316	**0.728**	0.403	0.443	0.737	0.504	0.438
SGFormer	0.670	0.588	0.478	0.447	0.406	0.254	0.533	0.257	0.249	0.591	0.436	0.346
DMGC-GTN	0.786	0.684	0.599	0.576	0.514	0.408	0.699	0.444	0.463	**0.763**	0.566	0.532
Ours	**0.819**	**0.737**	**0.669**	**0.590**	**0.554**	**0.419**	**0.728**	**0.466**	**0.490**	0.753	**0.581**	**0.542**

contrastive deep graph clustering methods. The details are as follows:
Deep Graph Clustering Methods(Non-Contrastive): **K-Means** (MacQueen, 1967), **GAE** (Kipf & Welling, 2016), **VGAE** (Kipf & Welling, 2016), **DAEGC** (Wang et al., 2019), **GATE** (Salehi & Davulcu, 2019), **SDCN** (Bo et al., 2020), **EGAE** (Zhang et al., 2023), **DCCA** (Andrew et al., 2013), **DCCAE** (Wang et al., 2015), **CO-GCN** (Li et al., 2020).

Contrastive Deep Graph Clustering Methods: **NCLA** (Shen et al., 2023), **SGCMC** (Xia et al., 2022a), **CMGEC** (Wang et al., 2023a), **SGFormer**, **DMGC-GTN** (Wang et al., 2025).

4.2 Experimental Results

In Table 2, we comprehensively compare our proposed method with ten classical deep graph clustering methods and five contrastive deep graph clustering methods on all evaluation metrics. Our method performs better than classical deep graph clustering and contrastive deep graph clustering methods. On the one hand, compared with classical deep graph clustering methods, the optimal transport-driven pseudo-label generation achieves more robust semantic alignment across views. On the other hand, unlike existing contrastive learning-based graph clustering approaches, our method innovatively introduces a dual-distribution constraint mechanism, which performs joint optimization in both the feature space and the neighborhood distribution space. This design empowers the learned clustering assignments with strong discriminability as well as structural consistency. Overall, our method achieves superior performance over most baselines on four datasets in terms of all three metrics. Taking the result on the AMAP dataset for example, our method achieves an ACC of 0.819, significantly outperforming existing top-performing methods. Regarding NMI, our model secures 0.737, surpassing the DMGC-GTN model of 0.684 by approximately 5.3%. For ARI, our model scores 0.669, improving by about 7.0% over DMGC-GTN. These observations validate the superiority of our proposed framework.

Table 3: Ablation study of different loss components on four datasets.

Method				AMAP			WIKI			CITESEER			CORA		
$\mathcal{L}_c$	$\mathcal{L}_s$	$\mathcal{L}_d$	$\mathcal{L}_n$	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
×	√	√	√	0.743	0.652	0.561	0.496	0.448	0.294	0.678	0.431	0.436	0.706	0.524	0.466
√	×	√	√	0.748	0.657	0.567	0.515	0.452	0.316	0.653	0.390	0.395	0.710	0.536	0.468
√	√	×	√	0.777	0.671	0.588	0.564	0.530	0.372	0.702	0.445	0.451	0.750	0.555	0.526
√	√	√	×	0.769	0.666	0.578	0.533	0.519	0.360	0.693	0.442	0.449	0.726	0.551	0.470
√	√	√	√	**0.819**	**0.737**	**0.669**	**0.590**	**0.554**	**0.419**	**0.728**	**0.466**	**0.490**	**0.753**	**0.581**	**0.542**

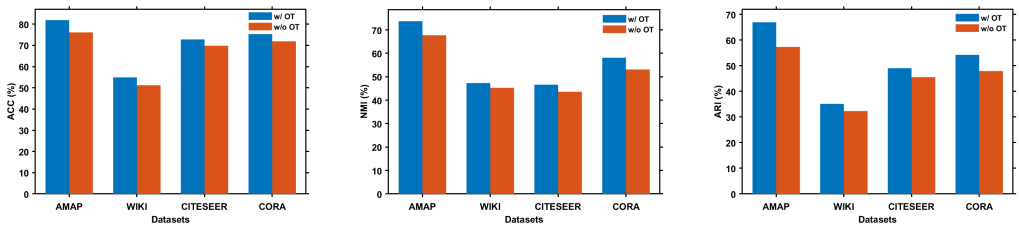
4.3 Ablation Experimental Analysis

Ablation on Model Modules. We conduct systematic ablation experiments to analyze the contributions of each core loss function in our model. The experiments are performed on the four dataset by successively removing the contrastive learning loss ($\mathcal{L}_c$), the cross-view node consistency loss($\mathcal{L}_n$), the optimal transport-based swapped prediction loss ($\mathcal{L}_s$), and the dual-distribution constraint loss ($\mathcal{L}_d$), in order to evaluate the impact of each component on the overall clustering performance. In particular, removing the graph-guided contrastive loss or the optimal transport-based swapped prediction loss causes the largest performance drops, highlighting their importance for learning discriminative representations and globally consistent cross-view pseudo-labels. The degradation caused by removing the dual-distribution constraint and cross-view node consistency losses further confirms their complementary roles in preserving feature-level and topology-aware cluster consistency.

Ablation of Optimal Transport. We have further explored OT. To be specific, we consider removing OT from Eq.(8), thus adopting the common approach of pseudo-label generation. The specific experimental results can be found in Fig 2, where "w/ OT" and "w/o OT" represent method with and without OT, respectively. The figure shows that method with OT performs better than without OT. This improvement can be attributed to the ability of OT-based pseudo-labeling to achieve more accurate cross-view semantic matching and alignment, effectively enhancing the discriminability of learned representations.

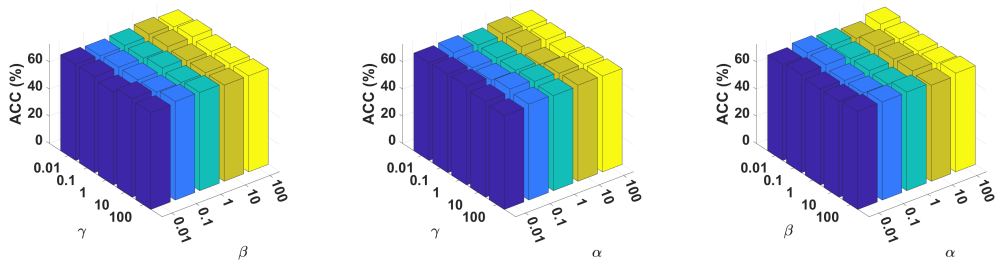
4.4 Parameter Analysis

To gain a deeper understanding of the sensitivity of the proposed model to the key hyperparameters α (contrastive learning loss weight), β (cross-view node consistency loss weight), and γ (dual-distribution constraint loss weight), we conducted a systematic parameter study on the Citeseer dataset. We investigated the impact of two parameters when the third is fixed through three sets of experiments, using accuracy (ACC) as evaluation metrics. The α, β and γ are chosen from $\{0.01, 0.1, 1, 10, 100\}$. From Figs 3, we can observe that different parameter combinations have a slight impact on the clustering performance but do not have much impact on the balance value. This shows that our method is not so sensitive to the selection of hyper-parameters, indicating the stability of our method.



(**a**) ACC (**b**) NMI (**c**) ARI

Figure 2: Experimental results of method with and without OT on four datasets.

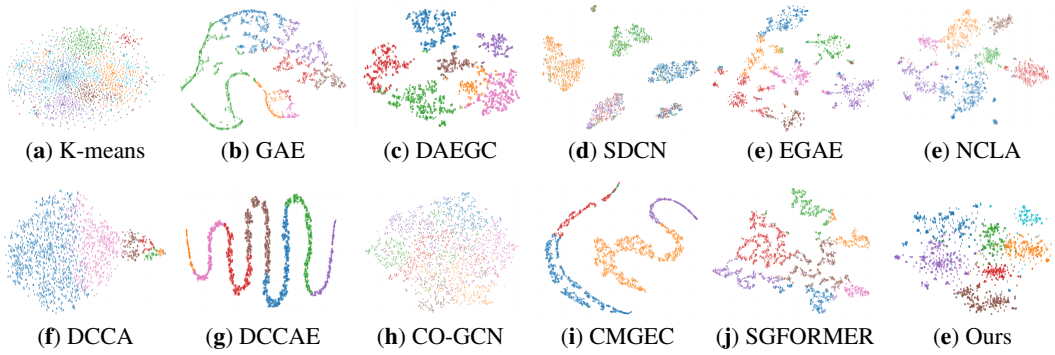


(**a**) constant value α (**b**) constant value β (**c**) constant value γ

Figure 3: The ACC values yielded by the method with different α, β and γ combinations.

4.5 Visualization Analysis

To provide an intuitive demonstration of the effectiveness of our proposed method, we conducted a comprehensive visualization analysis of the clustering results on the CORA datasets. We employ the t-SNE algorithm(van der Maaten & Hinton, 2008) to map high-dimensional node embeddings into a twodimensional space for visual inspection. Figs 4 present a comparative analysis of the clustering performance of our method against several state-of-the-art methods. The proposed method demonstrates superior clustering visualization performance, where nodes from different categories form well-separated and compact clusters in the embedding space with significantly reduced inter-category overlap. This indicates that our optimal transport-based pseudo-label generation mechanism and dual-distribution constraint strategy effectively learn discriminative and structurally consistent node representations, thereby enhancing both visual separability and semantic coherence for multi-view graph clustering.



(**a**) K-means (**b**) GAE (**c**) DAEGC (**d**) SDCN (**e**) EGAE (**e**) NCLA

(**f**) DCCA (**g**) DCCAE (**h**) CO-GCN (**i**) CMGEC (**j**) SGFORMER (**e**) Ours

Figure 4: The visualization of node representation using the t-SNE algorithm in the CORA dataset.

5 Conclusions

This paper proposes a dual-distribution-constrained self-supervised multi-view graph clustering framework that deeply integrates graph topology, multi-view features, and the clustering optimization process to significantly enhance graph data clustering performance. The method achieves cross-view semantic alignment through dual-view contrastive learning and optimal transport-based pseudo-label generation. It further introduces a dual-distribution constraint mechanism to jointly optimize the feature space and the topology-aware distribution space, enabling the clustering results to possess both discriminative power and structural awareness. Extensive experiments and ablation studies have been conducted on four datasets to validate the superiority of the proposed model and the effectiveness of its constituent components.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and polishing the manuscript, including improving the clarity, grammar, academic style, and LaTeX presentation of the text. We have not used generative AI tools to generate or modify datasets, fabricate or alter experimental results, make unsupported scientific claims, or replace the authors' judgment in methodological and experimental decisions. The proposed method, code implementation, experiment execution, result analysis, and final conclusions were determined and verified by the authors. All AI-assisted text and suggestions were carefully reviewed and revised by the authors. In particular, mathematical formulations, citations, descriptions of the implementation, and experimental claims were checked against the project code, experimental records, and the corresponding literature. The authors take full responsibility for the final content of this work, including all text, claims, equations, and other artifacts produced with the assistance of generative AI.

REFERENCES

Galen Andrew, Ramanathan Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pp. 1247–1255. PMLR, 2013.

Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *WWW*, pp. 1400–1410. ACM / IW3C2, 2020.

Jiafeng Cheng, Qianqian Wang, Zhiqiang Tao, Deyan Xie, and Quanxue Gao. Multi-view attribute graph convolution networks for clustering. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 2973–2979. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/411. URL https://doi.org/10.24963/ijcai.2020/411. Main track.

Lawrence J. Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. URL https://api.semanticscholar.org/CorpusID:189915041.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders, 2016. URL https://arxiv.org/abs/1611.07308.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. URL https://arxiv.org/abs/1609.02907.

Shu Li, Wen-Tao Li, and Wei Wang. Co-gcn for multi-view semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4691–4698, 2020.

J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967. URL https://api.semanticscholar.org/CorpusID:6278891.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Inf. Retr.*, 3(2):127–163, July 2000. ISSN 1386-4564. doi: 10.1023/A:1009953814988. URL https://doi.org/10.1023/A:1009953814988.

Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks, 2022. URL https://arxiv.org/abs/2007.02901.

Feiping Nie, Jing Li, and Xuelong Li. Self-weighted multiview clustering with multiple graphs. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pp. 2564–2570. AAAI Press, 2017. ISBN 9780999241103.

Amin Salehi and Hasan Davulcu. Graph attention auto-encoders. In *Arxiv*, 2019.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, September 2008. ISSN 0738-4602. doi: 10.1609/aimag.v29i3.2157. URL https://doi.org/10.1609/aimag.v29i3.2157.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation, 2019. URL https://arxiv.org/abs/1811.05868.

Xiao Shen, Dewang Sun, Shirui Pan, Xi Zhou, and Laurence T. Yang. Neighbor contrastive learning on learnable graph augmentation, 2023. URL https://arxiv.org/abs/2301.01404.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: A deep attentional embedding approach, 2019. URL https://arxiv.org/abs/1906.06532.

Hao Wang, Yan Yang, and Bing Liu. Gmc: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2020. doi: 10.1109/TKDE.2019.2903810.

Q. Wang, H. Xu, Z. Zhang, W. Feng, and Q. Gao. Deep multi-modal graph clustering via graph transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7835–7843, 2025. doi: 10.1609/aaai.v39i8.32844. URL https://doi.org/10.1609/aaai.v39i8.32844.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1083–1092. JMLR.org, 2015.

Yiming Wang, Dongxia Chang, Zhiqiang Fu, and Yao Zhao. Consistent multiple graph embedding for multi-view clustering. *IEEE Transactions on Multimedia*, 25:1008–1018, 2023a. doi: 10.1109/TMM.2021.3136098.

Yiming Wang, Dongxia Chang, Zhiqiang Fu, and Yao Zhao. Consistent multiple graph embedding for multi-view clustering. *IEEE Transactions on Multimedia*, 25:1008–1018, 2023b. doi: 10.1109/TMM.2021.3136098.

Wei Xia, Qianqian Wang, Quanxue Gao, Xiangdong Zhang, and Xinbo Gao. Self-supervised graph convolutional network for multi-view clustering. *IEEE Transactions on Multimedia*, 24:3182–3192, 2022a. doi: 10.1109/TMM.2021.3094296.

Wei Xia, Qianqian Wang, Quanxue Gao, Xiangdong Zhang, and Xinbo Gao. Self-supervised graph convolutional network for multi-view clustering. *IEEE Transactions on Multimedia*, 24:3182–3192, 2022b. doi: 10.1109/TMM.2021.3094296.

Jie Xu, Yazhou Ren, Guofeng Li, Lili Pan, Ce Zhu, and Zenglin Xu. Deep embedded multi-view clustering with collaborative training. *Information Sciences*, 573:279–290, 2021. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2020.12.073. URL https://www.sciencedirect.com/science/article/pii/S0020025520312391.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings, 2016. URL https://arxiv.org/abs/1603.08861.

Hongyuan Zhang, Pei Li, Rui Zhang, and Xuelong Li. Embedding graph auto-encoder for graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):9352–9362, 2023. doi: 10.1109/TNNLS.2022.3158654.

Xiaofeng Zhu, Shichao Zhang, Wei He, Rongyao Hu, Cong Lei, and Pengfei Zhu. One-step multi-view spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):2022–2034, 2019. doi: 10.1109/TKDE.2018.2873378.