

MoEGCL: Mixture of Ego-Graphs Contrastive Representation Learning for Multi-View Clustering

Jian Zhu^{1†}, Xin Zou^{2†}, Jun Sun^{1*}, Cheng Luo^{1*}, Lei Liu³, Lingfang Zeng¹, Ning Zhang^{1*}, Bian Wu¹, Chang Tang⁴, Lirong Dai³

¹ Zhejiang Lab, Hangzhou, China

² Hong Kong University of Science and Technology, Guangzhou, China

³ University of Science and Technology of China, Hefei, China

⁴ Huazhong University of Science and Technology, Wuhan, China

† Contributed equally to this work, *Corresponding Author

Abstract

In recent years, the advancement of Graph Neural Networks (GNNs) has significantly propelled progress in Multi-View Clustering (MVC). However, existing methods face the problem of coarse-grained graph fusion. Specifically, current approaches typically generate a separate graph structure for each view and then perform weighted fusion of graph structures at the view level, which is a relatively rough strategy. To address this limitation, we present a novel Mixture of Ego-Graphs Contrastive Representation Learning (MoEGCL). It mainly consists of two modules. In particular, we propose an innovative Mixture of Ego-Graphs Fusion (MoEGF), which constructs ego graphs and utilizes a Mixture-of-Experts network to implement fine-grained fusion of ego graphs at the sample level, rather than the conventional view-level fusion. Additionally, we present the Ego Graph Contrastive Learning (EGCL) module to align the fused representation with the view-specific representation. The EGCL module enhances the representation similarity of samples from the same cluster, not merely from the same sample, further boosting fine-grained graph representation. Extensive experiments demonstrate that MoEGCL achieves state-of-the-art results in deep multi-view clustering tasks. The source code is publicly available at <https://github.com/HackerHyper/MoEGCL>.

Introduction

With the rapid advancement of digitalization, data is increasingly being collected from multiple heterogeneous sources. For example, autonomous vehicles rely on multi-camera systems to capture diverse visual perspectives for decision-making. In structural biology, proteins exhibit complex quaternary structures that can be characterized through different analytical techniques. Modern medical diagnostics integrate multimodal data from various clinical tests to enhance accuracy. The term “multi-view data” is used to describe an item from multiple data sources. Multi-View Clustering (MVC) (Tang et al. 2023) aims to integrate heterogeneous data from multiple views to discover meaningful cluster structures in an unsupervised manner, playing a pivotal role in modern data mining applications. These methods utilize a view-specific encoder network to generate a powerful representation, which is fused from various views for

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

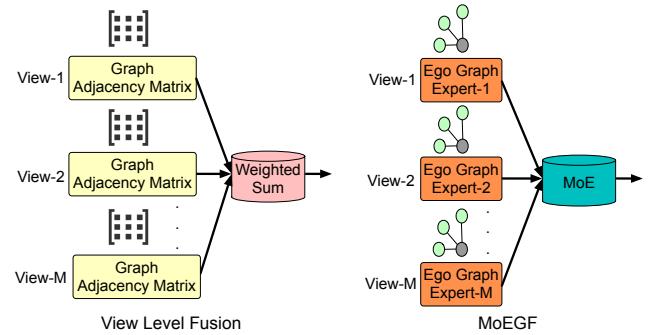


Figure 1: Mixture of Ego-Graphs Fusion (MoEGF). Firstly, construct an ego graph for each sample in each view. Secondly, input each ego graph as an expert into the Mixture-of-Experts (MoE) architecture and calculate the importance of each expert. Thirdly, merge the importance coefficient with the experts to form a fused ego graph. Lastly, aggregate the ego graphs of all samples to generate a global connected graph.

clustering tasks. However, there exists the issue of heterogeneity among data from different views. To solve the issue, many alignment methods have been proposed. For example, some methods use KL divergence to align the label or representation distributions among different views (Hershey and Olsen 2007). Additionally, MVC employs contrastive learning (CL) to align representations across different views.

In recent years, Graph Neural Networks (GNNs) have emerged as a powerful paradigm for extracting and processing graph-structured information, achieving remarkable success across various domains. Although GNNs have significantly improved the MVC, the problem of coarse-grained graph fusion still exists. For instance, existing methods like DFMVC (Ren et al. 2024) and CAGL (Wang et al. 2022b) first construct an independent graph for each view and then perform weighted fusion at the view level. However, this approach assigns a fixed fusion coefficient to the graph structure of each view, ignoring the differences between samples. This fusion approach operates at a coarse granularity and fails to capture fine-grained structural relationships, which ultimately limits performance in deep multi-view clustering

tasks.

To address the problem of coarse-grained graph fusion, we propose a novel Mixture of Ego-Graphs Contrastive Representation Learning (MoEGCL). It consists of two modules, which are Mixture of Ego-Graphs Fusion (MoEGF) and Ego Graph Contrastive Learning (EGCL). As illustrated in Figure 1, MoEGF is a new paradigm in deep multi-view clustering. Our proposed paradigm creates ego graphs and uses a Mixture-of-Experts (MoE) architecture (Jacobs et al. 1991) for the fusion of ego graphs at the sample level. The MoE architecture empowers MoEGF to dynamically integrate information from multiple domain-specific experts. We use the ego graph of each view as an expert in MoE architecture. Different ego graphs can denote the connectivity relationship of a sample in different views. Therefore, the MoEGF module adaptively calculates the confidence of the ego graph, rather than using a fixed weight to measure the graph of each view. This enables better utilization of refined information, achieving fine-grained graph fusion. In addition, we propose the EGCL module to enhance the representation similarity of the samples in the same cluster, rather than the same sample. The EGCL module overcomes the limitations of contrastive learning in current multi-view clustering. It obtains better semantic structure information, thereby improving the fine-grained graph representation.

The main contributions can be summarized as follows:

- We present the MoEGF module in the context of multi-view learning. It implements fine-grained graph fusion via a Mixture-of-Experts network at the sample level.
- Different with previous approaches that consider different views on a sample to be positive points, we propose the EGCL module to improve the representation similarity of samples within the cluster, which enables fine-grained graph representation.
- Extensive experiments demonstrate that the MoEGCL module establishes new state-of-the-art performance in deep multi-view clustering, outperforming existing methods by significant margins on the six public datasets.

Related Works

Mult-view Clustering

There are mainly two kinds of methods in Multi-View Clustering (MVC) (Chen et al. 2022): Shallow Multi-View Clustering methods and Deep Multi-View Clustering methods.

Shallow Multi-View Clustering methods (Chen et al. 2022) are classified into two categories: Graph Multi-View Clustering and Subspace Multi-View Clustering. Graph Multi-View Clustering methods (Fang et al. 2022; Pan and Kang 2021) generally follow these steps: construct view-specific graphs, followed by the creation of a fusion graph from several view-specific graphs using different regularization terms, and finally the generation of the clustering results using spectral clustering, graph-cut approaches, or other algorithms. Subspace Multi-View Clustering methods (Zhang et al. 2022; Huang et al. 2023) create a common subspace self-representation vector from every view, and then separate the samples into different subspaces using different regularization terms on the consensus self-representation matrix.

Many recent works have focused on Deep Multi-View Clustering, motivated by encouraging developments in deep learning (Yu et al. 2025; Xu et al. 2023). Specifically, these methods encode the non-linear feature using a deep neural network. The adversarial methodology is used in Deep Multi-View Clustering methods (Li and Liao 2021; Wang et al. 2022a) to coordinate patterns of hidden representations from different views and learn the latent representation. The attention mechanism is used by Zhou et al. (Zhou and Shen 2020) to allocate each view a weight value. To obtain a consensus representation, they then use the weighted total of all view-wise presentations. The consensus representation is produced by Wang et al. (Wang et al. 2022a) using a weighted aggregation and $l_{1,2}$ -norm restriction. Contrastive Learning (CL) may align representations from different perspectives at the sample level, which facilitates label distribution alignment. These CL methods (Trost et al. 2021; Xu et al. 2022) have outperformed the previous distribution alignment strategies in Multi-View Clustering, yielding state-of-the-art results on numerous public datasets.

Mixture of Experts

Jacobs et al. (Jacobs et al. 1991) present the first Mixture-of-Experts (MoE) structure. It is viewed as a tree-structured framework that employs the divide-and-conquer tactic. Sparsely-gated MoE (Shazeer et al. 2017) is the first method to show significant gains in model capacity, training time, or model quality when gating is used. MH-MoE (Wu et al. 2024b) divides each input token into many smaller tokens, which are then assigned to and processed by a variety of experts concurrently before being smoothly reintegrated into the original token form.

We use a Mixture-of-Experts network to implement fine-grained graph fusion at the sample level. To the best of our knowledge, we are the first to present the fusion of ego graphs.

The Proposed Methodology

This paper proposes an innovative Mixture of Ego-Graphs Contrastive Representation Learning (MoEGCL), which aims to overcome the limitation of coarse-grained graph fusion in deep multi-view clustering. As illustrated in Figure 2, MoEGCL comprises two key components: 1) Mixture of Ego-Graphs Fusion (MoEGF) and 2) Ego Graph Contrastive Learning (EGCL). The multi-view data, which consists of N samples with M views, is denoted as $\{\mathbf{X}^m = \{x_1^m; \dots; x_N^m\} \in \mathbb{R}^{N \times D_m}\}_{m=1}^M$, where D_m represents the feature dimensionality for the m -th view.

Autoencoder Network

We create the representation of each view by using the Autoencoder architecture (Song et al. 2018). It contains two parts: an encoder and a decoder. For the m -th view, f^m represents the encoder module. The encoder module generates the representation of each view as follows:

$$z_i^m = f^m(x_i^m), z_i^m \in \mathbb{R}^{d_\psi}, \quad (1)$$

where z_i^m denotes the representation of the i -th instance in the m -th view. d_ψ represents the feature size of the z_i^m .

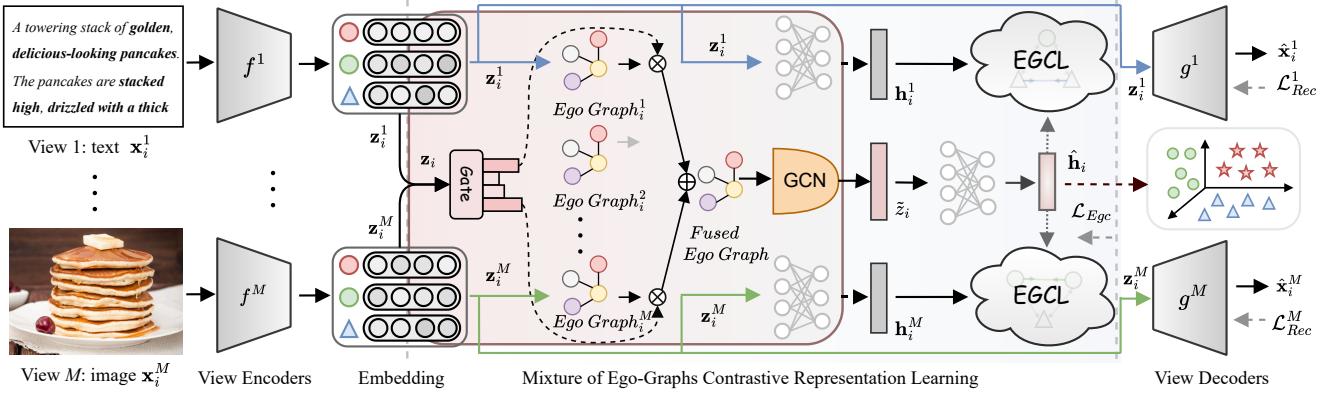


Figure 2: The architecture diagram of MoEGCL. The MoEGCL method contains two modules: MoEGF and EGCL. MoEGF first constructs ego graphs and then aggregates ego graphs of all views into a fused ego graph by a Mixture-of-Experts network. In addition, the EGCL module is proposed to improve the representation similarity of samples in the same cluster, as opposed to only concentrating on the consistency at the sample level. It significantly enhances fine-grained graph representation.

The x_i^m is restored by the decoder module utilizing the feature z_i^m . Let g^m represents the decoder module in the m -th view. The recovered sample \hat{x}_i^m is produced by decoding z_i^m in the decoder module:

$$\hat{x}_i^m = g^m(z_i^m). \quad (2)$$

The reconstruction loss is calculated as follows:

$$\begin{aligned} \mathcal{L}_{\text{Rec}} &= \sum_{m=1}^M \|X^m - \hat{X}^m\|_F^2 \\ &= \sum_{m=1}^M \sum_{i=1}^N \|x_i^m - g^m(z_i^m)\|_2^2, \end{aligned} \quad (3)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Mixture of Ego-Graphs Fusion (MoEGF)

We present the Mixture of Ego-Graphs Fusion (MoEGF) to achieve fine-grained graph fusion. It consists of three components: Ego Graph Networks, Mixture-of-Experts Network, and Graph Convolutional Network.

Ego Graph Networks First, we create Ego Graph Networks at all views. In particular, we construct the graph $G^m = \{Z^m, S^m\}$ for the m -th view, where $Z^m = [z_1^m; z_2^m; \dots; z_N^m]$ represents the node features of graph G^m . S^m is the adjacency matrix. In the m -th view, the edge between samples i and j is represented by S_{ij}^m . If $S_{ij}^m = 1$, samples i and j are connected; otherwise, the two samples are unconnected. We create the adjacency matrix S^m using the KNN graph (Peterson 2009) as follows:

$$S_{ij}^m = \begin{cases} 1 & j \in K_i^m; \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

In the m -th view, K_i^m is the collection of k nearest neighbors of z_i^m based on the Euclidean distance.

$$S^m = (V_1^m; V_2^m; \dots; V_N^m)^T, \quad (5)$$

where V_i^m represents the connection relationship of the ego graph, which is from the m -th view of the i -th sample. Specifically, it is implemented using the row vectors of the adjacency matrix, also known as the adjacency vector.

Mixture-of-Experts Network We use the Mixture-of-Experts Networks to integrate ego graphs. It consist of gating network and an expert network. We concatenate the vectors of M views as input to the gating network.

$$z_i = \text{cat}(z_i^1; z_i^2; \dots; z_i^M), z_i \in \mathbb{R}^{Md_\psi}. \quad (6)$$

We calculate the gating coefficients of experts related to input z_i . It is implemented through a multi-layer perceptron neural network mlp^1 , followed by classification using a softmax activation function. The number of output categories by softmax is M . The specific calculation is as follows:

$$C_i = \text{softmax}(mlp^1(z_i)). \quad (7)$$

Let V_i^m be the m -th expert network. Combining the gate network and expert network, a weighted sum is performed to implement the fusion of ego graphs at the sample level:

$$V_i = \sum_{m=1}^M C_i^m V_i^m, \quad (8)$$

V_i denotes the fused adjacency vector of the i -th sample. Aggregating all adjacency vectors can generate an adjacency matrix.

$$S = (V_1; V_2; \dots; V_N)^T, \quad (9)$$

where S denotes the fused adjacency matrix.

Graph Convolutional Network To improve the clustering results using both global features and structural data, we additionally use the Graph Convolutional Networks (GCN) (Kipf and Welling 2017) to generate boosted graph representations. We adopt a two-layer GCN module to obtain the topological structure information. The graph representation \tilde{Z}_i is computed as follows:

$$\tilde{Z} = \left(\tilde{D}^{-\frac{1}{2}} \tilde{S} \tilde{D}^{-\frac{1}{2}} \right) \left[\left(\tilde{D}^{-\frac{1}{2}} \tilde{S} \tilde{D}^{-\frac{1}{2}} \right) Z W^0 \right] W^1, \quad (10)$$

where $\tilde{S} = I_N + S$. I_N is the identity matrix. \tilde{D} is a diagonal matrix, $\tilde{D}_{ii} = \sum_j \tilde{S}_{ij}$. W^0 and W^1 are the training parameters. $Z = \{z_1; z_2; \dots; z_N\}$ is an input feature, and $\tilde{Z} = \{\tilde{z}_1; \tilde{z}_2; \dots; \tilde{z}_N\}$ is an output feature.

Ego Graph Contrastive Learning (EGCL)

We propose the Ego Graph Contrastive Learning (EGCL) module to improve fine-grained graph representation in deep multi-view clustering tasks. EGCL needs to unify the dimensions of each view feature and the fused feature. The specific calculation is as follows:

$$\hat{h}_i = mlp^2(\tilde{z}_i), \hat{h}_i \in \mathbb{R}^{d_\phi}, \quad (11)$$

where the dimension of \tilde{z}_i is decreased via the mlp^2 operator. d_ϕ denotes the feature dimension of the \hat{h}_i . Similarly, we minimize dimension on each view feature z_i^m using the $mlp^{3,m}$ operator,

$$h_i^m = mlp^{3,m}(z_i^m), h_i^m \in \mathbb{R}^{d_\phi}. \quad (12)$$

For the h_i^m , d_ϕ is the feature size. The cosine function is used to compute the similarity between common embedding \hat{h}_i and view-specific embedding h_i^m :

$$C(\hat{h}_i, h_i^m) = \cos(\hat{h}_i, h_i^m). \quad (13)$$

The loss function of the EGCL module is calculated as follows:

$$\mathcal{L}_{Egc} = -\frac{1}{2N} \sum_{i=1}^N \sum_{m=1}^M \log \frac{e^{C(\hat{h}_i, h_i^m)/\tau}}{\sum_{j=1}^N e^{(1-S_{ij})C(\hat{h}_i, h_j^m)/\tau} - e^{1/\tau}}, \quad (14)$$

where τ denotes the temperature coefficient. Based on Eq. (9), we obtain S_{ij} . We incorporate S_{ij} into Eq. (14) to improve the similarity of the samples in the cluster. When S_{ij} is low, the two samples are not from the same cluster. At this point, the $(1 - S_{ij}) C(\hat{h}_i, h_j^m)$ plays a significant role. When S_{ij} is relatively large, it indicates the two samples belong to the same cluster. At this time, the effect of $(1 - S_{ij}) C(\hat{h}_i, h_j^m)$ is small.

The overall loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}_{Rec} + \lambda \mathcal{L}_{Egc}, \quad (15)$$

where λ balances the contribution of the two loss terms. Here, \mathcal{L}_{Rec} denotes the reconstruction loss, while \mathcal{L}_{Egc} represents the ego graph contrastive loss. Our training pipeline consists of two distinct phases: pre-training phase and fine-tuning phase. \mathcal{L}_{Rec} is the loss function we use during the pre-training phase, while \mathcal{L} is the loss function during the fine-tuning phase.

Clustering Module

We utilize the k-means method as the clustering function (MacKay, Mac Kay et al. 2003; Yan et al. 2023). Specifically, the common embedding \mathbf{H} has the following factorization:

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} \quad & \|\mathbf{H} - \mathbf{UV}\|^2, \mathbf{H} = \{\hat{h}_1, \dots, \hat{h}_N\}, \\ s.t. \quad & \mathbf{U} \mathbf{1} = \mathbf{1}, \mathbf{U} \geq \mathbf{0}, \end{aligned} \quad (16)$$

where $\mathbf{U} \in \mathbb{R}^{N \times k}$ denotes the matrix of cluster indicator. We use $\mathbf{V} \in \mathbb{R}^{k \times d_\phi}$ as the matrix of clustering center.

Table 1: Summary of the six public datasets.

Datasets	Samples	Views	Clusters	#-view Dimensions
Caltech5V	1400	5	7	[40, 254, 1984, 512, 928]
WebKB	1051	2	2	[2949, 334]
LGG	267	4	3	[2000, 2000, 333, 209]
MNIST	60000	3	10	[342, 1024, 64]
RGBD	1449	2	13	[2048, 300]
LandUse	2100	3	21	[20, 59, 59]

Experiments

In this section, we conduct extensive experiments to evaluate MoEGCL on six benchmark datasets for deep multi-view clustering. To demonstrate its superiority, we compare against eight state-of-the-art baselines across multiple metrics. Furthermore, we perform convergence analysis, visualization, and hyperparameter optimization.

Benchmark Datasets

As illustrated in Table 1, we utilize six benchmark datasets with various scales for analysis and comparison. These six datasets are Caltech5V, WebKB, LGG, MNIST, RGBD, and LandUse. These six datasets are publicly available and commonly used to evaluate the performance of Multi-View Clustering.

Compared Methods

To evaluate the effectiveness of the proposed MoEGCL, we use eight recent state-of-the-art multi-view clustering methods as baselines. These methods are deep learning methods (including DEMVC (Xu et al. 2021), DSMVC (Tang and Liu 2022), DealMVC (Yang et al. 2023), GCFAggMVC (Yan et al. 2023), SCMVC (Wu et al. 2024a), MVCAN (Xu et al. 2024), ACCMVC (Yan et al. 2024), and DMAC (Wang et al. 2025)).

Implementation Details

The source code of this paper is implemented using the PyTorch framework. To improve the generalization ability of the model, we set the dropout parameter of the network to 0.1. During the model training process, we set the batch size b to 256. Model training is divided into two stages: pre-training and fine-tuning. The number of pre-training epochs T_p is 200, and the number of fine-tuning epochs T_f is 300. The temperature coefficient τ for contrastive learning loss is set to 0.5. We unify the dimensions of the vectors outputted by the encoder and the input vectors for the contrastive loss, making d_ψ equal to 512 and d_ϕ equal to 128. The learning rate of deep network training is 0.0003. The combination coefficient λ of the two loss functions is set to 1. The hardware platforms we used in our experiment are Intel(R) Xeon(R) Platinum 8358 CPU and Nvidia A40 GPU.

Evaluation Metrics

We utilize three standard quantitative metrics, including unsupervised clustering accuracy (ACC), normalized mutual information (NMI), and purity (PUR). The larger these three

Table 2: Clustering result comparison on the MNIST, LGG, and WebKB datasets. The best results in bold, second in underline.

Methods	MNIST			LGG			WebKB		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
DEMVC [IS'21]	0.9734	0.9587	0.9734	0.5318	0.2840	0.5693	0.6156	0.0158	0.7812
DSMVC [CVPR'22]	0.9671	0.9215	0.9671	0.5993	0.1721	0.6105	0.7165	0.2303	0.7812
DealMVC [MM'23]	0.9804	0.9593	0.9804	0.6067	0.4500	0.6592	0.7755	0.0170	0.7812
GCFAggMVC [CVPR'23]	0.6949	0.6768	0.6949	0.5094	0.0355	0.5131	0.8259	0.3031	0.8259
SCMVC [TMM'24]	0.8610	0.8263	0.8610	0.5318	0.1467	0.5655	0.8230	0.3702	0.8230
MVCAN [CVPR'24]	0.9863	0.9612	0.9863	0.6517	0.4329	0.6742	0.8202	0.2023	0.8202
ACCMVC [TNNLS'24]	0.9886	0.9659	0.9886	0.4981	0.1611	0.5618	0.8696	0.4863	0.8696
DMAC [AAAI'25]	0.9720	0.9302	0.9720	0.4607	0.0396	0.5131	0.8145	0.1844	0.8145
MoEGCL (Ours)	0.9920	0.9747	0.9920	0.7491	0.4534	0.7491	0.9515	0.6503	0.9515

Table 3: Clustering result comparison on the Caltech5V, RGBD, and LandUse datasets. The best in bold, the second underlined.

Methods	Caltech5V			RGBD			LandUse		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
DEMVC [IS'21]	0.1693	0.0443	0.1786	0.3858	0.3058	0.4865	0.2057	0.2032	0.2171
DSMVC [CVPR'22]	0.5979	0.4400	0.6121	0.4341	0.3740	0.5418	<u>0.2695</u>	<u>0.3494</u>	<u>0.3048</u>
DealMVC [MM'23]	0.6157	0.5081	0.6157	0.4320	0.2591	0.4645	0.1895	0.2014	0.1929
GCFAggMVC [CVPR'23]	0.2893	0.1264	0.2979	0.3375	0.3098	0.4624	0.2610	0.2991	0.2833
SCMVC [TMM'24]	0.6086	0.4210	0.6100	0.3879	0.3093	0.5231	0.2595	0.2791	0.2752
MVCAN [CVPR'24]	<u>0.7936</u>	<u>0.6970</u>	<u>0.7936</u>	<u>0.4458</u>	<u>0.4134</u>	<u>0.5977</u>	0.2395	0.3099	0.2867
ACCMVC [TNNLS'24]	0.6807	0.5516	0.7129	0.2926	0.2695	0.4500	0.2610	0.2952	0.2767
DMAC [AAAI'25]	0.5850	0.4609	0.5907	0.3106	0.2218	0.4472	0.2429	0.2899	0.2786
MoEGCL (Ours)	0.8207	0.7000	0.8207	0.4886	0.5516	0.6473	0.3381	0.4206	0.3500

indicators are, the better the clustering performance of the model.

Experimental comparative results

As shown in Table 2 and Table 3, MoEGCL achieves superior performance compared to eight state-of-the-art baselines (DEMVC, DSMVC, DealMVC, GCFAggMVC, SCMVC, MVCAN, ACCMVC, and DMAC) across six benchmark datasets. In particular, we derive the following findings: On the WebKB dataset, MoEGCL surpasses the second-best method (ACCMVC) by 8.19% in ACC. Similarly, on the RGBD dataset, our method outperforms MVCAN by 4.28% in ACC, with consistent gains in NMI and PUR. In the other four datasets, MoEGCL achieves state-of-the-art results on all remaining datasets, demonstrating robust generalizability. The performance advantage stems from two key innovations:

- We present the MoEGF module in multi-view learning, which uses a Mixture-of-Experts network to achieve the fusion of ego graphs.
- We propose the EGCL module to improve the representation similarity of samples in the cluster, which enhances fine-grained graph representation.

Ablation Study

We conduct an ablation experiment to evaluate each component of the proposed MoEGCL on the six public datasets.

Effectiveness of MoEGF module. The “w/o MoEGF” denotes removing the MoEGF module from the MoEGCL framework. The fused representation is set to z_i , which is the concatenation of all view-specific representations $\{z_i^m\}_{m=1}^M$. Table 4 illustrates that, in the ACC term, the results of w/o MoEGF are 37.64, 6.00, 1.03, 40.92, 27.34, and 8.67 percent less than those of the MoEGCL. The results demonstrate that the MoEGF module significantly improves the performance of deep multi-view clustering.

Validity of EGCL module. The “w/o EGCL” represents the elimination of the EGCL module in the MoEGCL framework. As illustrated in Table 4, the results of w/o EGCL are lower than those of the MoEGCL method by 24.43, 0.41, 0.15, 16.27, 16.11, and 3.52 percent on the ACC term. As opposed to merely using the same sample, the similarity of view presentation from the same cluster improves the fine-grained graph representation. Therefore, it improves the results in multi-view clustering tasks.

Importance of MoE module. In Table 4, the “w/o MoE” indicates the removal of the MoE module and takes the average of the graph matrices for all views. We present the experimental results of “w/o MoE” on six common datasets. On the WebKB dataset, compared with “w/o MoE”, MoEGCL achieves improvements of 20.55%, 42.56%, and 17.03% on three evaluation metrics, respectively. Additionally, MoEGCL improves the three assessment metrics on the Caltech5V dataset by 6.21%, 7.59%, and 6.21%, respectively. Through the above experiments, it

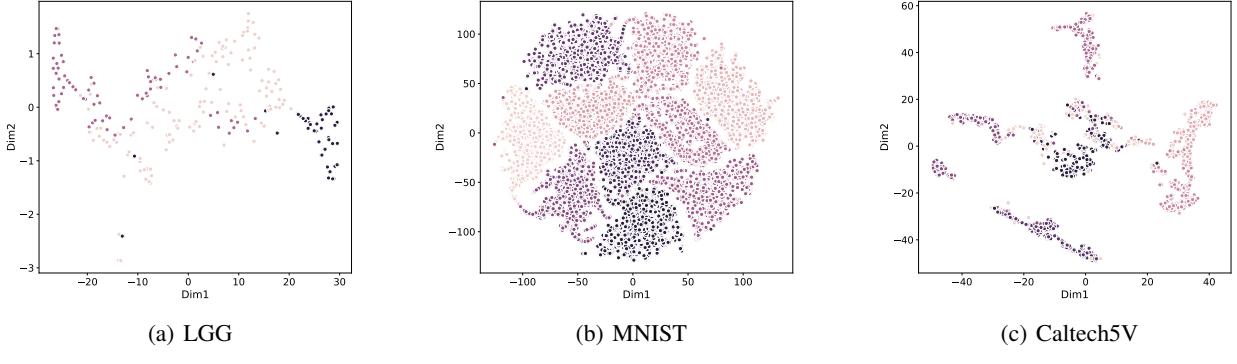


Figure 3: The visualization results of the fused representations $\{\hat{h}_i\}_{i=1}^N$ on the LGG, MNIST, and Caltech5V datasets after convergence. From the visualization results of the three subfigures, it can be concluded that our proposed method separates the samples completely in the feature space, thus proving that our method is very effective.

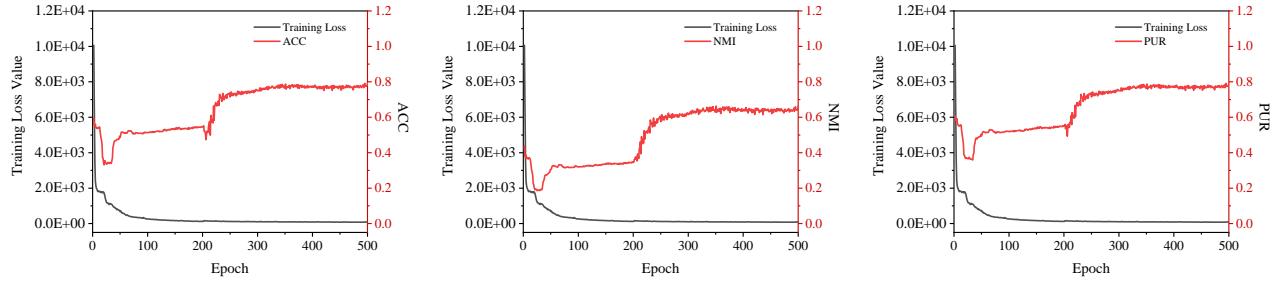


Figure 4: The convergence analysis on the Caltech5V dataset. In the figure, the test ACC, NMI, and PUR are shown at the top, and the training loss is depicted at the bottom. It can be observed that around 400 epochs, MoEGCL reaches a steady state, with the training loss no longer decreasing.

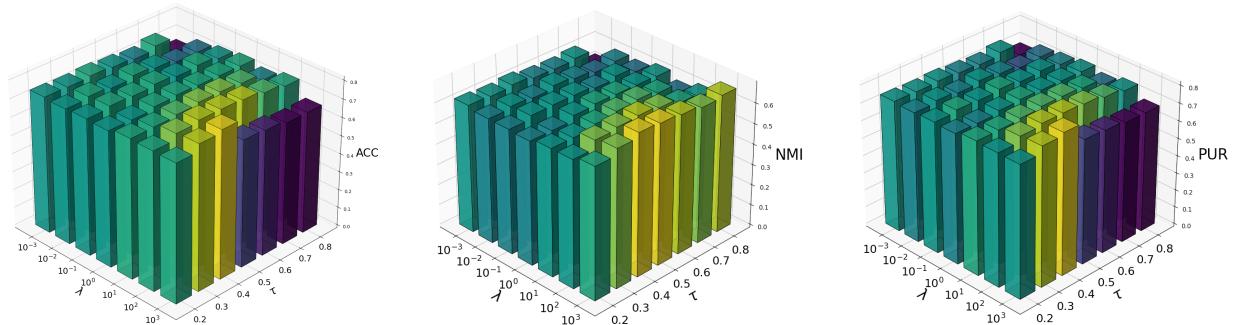


Figure 5: The parameter analysis on the Caltech5V dataset. The figure shows the changes in three evaluation metrics: ACC, NMI, and PUR. The metrics are influenced by two hyperparameters λ and τ . λ is the combination coefficient of two loss functions. τ denotes temperature coefficient.

Table 4: Ablation study on the six common datasets.

Datasets	Method	ACC	NMI	PUR
Caltech5V	w/o MoE	0.7586	0.6241	0.7586
	w/o GCN	0.6857	0.5875	0.7164
	w/o MoEGF	0.4443	0.2934	0.4943
	w/o EGCL	0.5764	0.4757	0.5907
	MoEGCL	0.8207	0.7000	0.8207
RGBD	w/o MoE	0.2202	0.1722	0.3865
	w/o GCN	0.2961	0.2551	0.4741
	w/o MoEGF	0.4286	0.4071	0.5845
	w/o EGCL	0.4845	0.5460	0.6342
	MoEGCL	0.4886	0.5516	0.6473
MNIST	w/o MoE	0.9915	0.9735	0.9915
	w/o GCN	0.9901	0.9704	0.9901
	w/o MoEGF	0.9817	0.9492	0.9817
	w/o EGCL	0.9905	0.9706	0.9905
	MoEGCL	0.9920	0.9747	0.9920
WebKB	w/o MoE	0.7460	0.2247	0.7812
	w/o GCN	0.5100	0.1112	0.7812
	w/o MoEGF	0.5423	0.0435	0.7812
	w/o EGCL	0.7888	0.2739	0.7888
	MoEGCL	0.9515	0.6503	0.9515
LGG	w/o MoE	0.5655	0.3094	0.6330
	w/o GCN	0.5805	0.2017	0.5805
	w/o MoEGF	0.4757	0.0968	0.5206
	w/o EGCL	0.5880	0.1715	0.5880
	MoEGCL	0.7491	0.4534	0.7491
LandUse	w/o MoE	0.3238	0.4038	0.3429
	w/o GCN	0.2652	0.3097	0.2767
	w/o MoG	0.2514	0.3083	0.2876
	w/o EGCL	0.3029	0.3912	0.3452
	MoEGCL	0.3381	0.4206	0.3500

is proved that MoEGCL is superior to “w/o MoE”.

Usefulness of GCN module. As illustrated in Table 4, the “w/o GCN” is the removal of the GCN module from the MoEGCL. It is shown that MoEGCL is better than “w/o GCN” on all six public datasets.

Visualization

To further verify the effectiveness of MoEGCL, we visualize the fused representations $\{\hat{h}_i\}_{i=1}^N$ using the t-SNE method (Van der Maaten and Hinton 2008), as shown in Figure 3. We conduct visualization experiments on the three datasets: LGG, MNIST, and Caltech5V.

In Figure 3, (a), (b), and (c) respectively represent the visualization results of the LGG, MNIST, and Caltech5V datasets. From the clustering results, it can be seen that the clusters after convergence are completely separated and the boundaries are relatively clear. There is basically no overlap between clusters. The visualization of Figure 3(a) shows that the samples are divided into 3 categories, and the visualization effect is very good, which can also correspond to the 3 labels of the LGG dataset. Similarly, the visualization results of Figures 3(b) and 3(c) show 10 categories and 7 categories, which are consistent with the labels of the MNIST and Caltech5V datasets.

Convergence Analysis and Generalization Ability

The convergence and generalization capabilities of MVC are crucial. To validate the convergence and generalization effectiveness of MoEGCL, we conduct tests on the Caltech5V dataset. The test ACC, NMI, PUR, and training loss are shown in Figure 4.

Figure 4 consists of three subfigures, which illustrate the variation curves of test ACC, NMI, PUR, and training loss. The loss curve steadily decreases as training progresses, stabilizing at 300 epochs. This indicates that the deep multi-view clustering network is functioning effectively. Right from the start of the experiment, the test ACC, NMI, and PUR for the evaluation measure show a significant increase. After 400 epochs, the ACC, NMI, and PUR stabilize and remain unchanged with further training, indicating robust generalization capabilities and the absence of overfitting. Similar convergence and generalization results are observed across multiple datasets.

Parameter Analysis

As shown in Figure 5, we conduct experiments on hyperparameter analysis. Specifically, we analyze the impact of two hyperparameters, λ and τ , on clustering evaluation metrics. λ represents the combination coefficient of two loss functions. τ denotes temperature coefficient. The evaluation metrics is the test ACC, NMI, and PUR. We conduct hyperparameter experiments on the Caltech5V dataset. The value of λ varies from 10^{-3} to 10^3 . The values of τ are 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, and 0.8. We draw three subfigures representing the changes in the three metrics ACC, NMI, and PUR under the influence of hyperparameters. Figure 5 shows that ACC, NMI, and PUR are not significantly affected by the two hyperparameters and exhibit relatively stable performance.

Conclusion and Future Work

This paper proposes a novel Mixture of Ego-Graphs Contrastive Representation Learning (MoEGCL) framework. It aims to implement fine-grained graph fusion for deep multi-view clustering tasks. It is composed of two modules, Mixture of Ego-Graphs Fusion (MoEGF) and Ego Graph Contrastive Learning (EGCL). The MoEGF module, which constructs ego graphs and utilizes the Mixture-of-Experts network, is presented for the fusion of ego graphs at the sample level. Furthermore, the EGCL module is proposed to lessen conflicts in the same clusters and get better semantic structure information. It further improves fine-grained graph representation. Extensive experiments demonstrate that MoEGCL achieves superior performance over state-of-the-art methods in deep multi-view clustering tasks. Specifically, we compare eight baseline methods on six public datasets to verify the effectiveness of our proposed method. In summary, MoEGCL is a simple and effective representation learning method. We believe it has good application prospects.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (Grant No. 021ZD0201501), the National Natural Science Foundation of China (No. 22574146, No. 32200860, and No. 62306289), the Youth Foundation Project of Zhejiang Province (Grant No. LQ22F020035), the “Pioneer” and “Leading Goose” R&D Program of Zhejiang (2024SSYS0007), the Zhejiang provincial “Ten Thousand Talents Program” (2021R52007), the National Key R&D Program of China (2022YFB4500405).

References

- Chen, M.; Lin, J.; Li, X.; Liu, B.; Wang, C.; Huang, D.; and Lai, J. 2022. Representation Learning in Multi-view Clustering: A Literature Review. *Data Science and Engineering*, 7: 225–241.
- Fang, R.; Wen, L.; Kang, Z.; and Liu, J. 2022. Structure-Preserving Graph Representation Learning. *IEEE International Conference on Data Mining*.
- Hershey, J. R.; and Olsen, P. A. 2007. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07*, volume 4, IV–317. IEEE.
- Huang, S.; Liu, Y.; Tsang, I. W.; Xu, Z.; and Lv, J. 2023. Multi-View Subspace Clustering by Joint Measuring of Consistency and Diversity. *IEEE Transactions on Knowledge and Data Engineering*, 35(8): 8270–8281.
- Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1): 79–87.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Li, Y.; and Liao, H. 2021. Multi-view clustering via adversarial view embedding and adaptive view fusion. *Applied Intelligence*, 51(3): 1201–1212.
- MacKay, D. J.; Mac Kay, D. J.; et al. 2003. *Information theory, inference and learning algorithms*. Cambridge university press.
- Pan, E.; and Kang, Z. 2021. Multi-view contrastive graph clustering. *Advances in neural information processing systems*, 34: 2148–2159.
- Peterson, L. E. 2009. K-nearest neighbor. *Scholarpedia*, 4(2): 1883.
- Ren, Y.; Pu, J.; Cui, C.; Zheng, Y.; Chen, X.; Pu, X.; and He, L. 2024. Dynamic weighted graph fusion for deep multi-view clustering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 4842–4850.
- Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q. V.; Hinton, G. E.; and Dean, J. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Song, J.; Zhang, H.; Li, X.; Gao, L.; Wang, M.; and Hong, R. 2018. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Transactions on Image Processing*, 27(7): 3210–3221.
- Tang, C.; Li, Z.; Wang, J.; Liu, X.; Zhang, W.; and Zhu, E. 2023. Unified One-Step Multi-View Spectral Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 6449–6460.
- Tang, H.; and Liu, Y. 2022. Deep safe multi-view clustering: Reducing the risk of clustering performance degradation caused by view increase. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 202–211.
- Trosten, D. J.; Lokse, S.; Jenssen, R.; and Kampffmeyer, M. 2021. Reconsidering representation alignment for multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1255–1265.
- Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
- Wang, B.; Zeng, C.; Chen, M.; and Li, X. 2025. Towards Learnable Anchor for Deep Multi-View Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 21044–21052.
- Wang, Q.; Tao, Z.; Xia, W.; Gao, Q.; Cao, X.; and Jiao, L. 2022a. Adversarial multiview clustering networks with adaptive fusion. *IEEE Transactions on Neural Networks and Learning Systems*.
- Wang, R.; Li, L.; Tao, X.; Wang, P.; and Liu, P. 2022b. Contrastive and attentive graph learning for multi-view clustering. *Information Processing & Management*, 59(4): 102967.
- Wu, S.; Zheng, Y.; Ren, Y.; He, J.; Pu, X.; Huang, S.; Hao, Z.; and He, L. 2024a. Self-weighted contrastive fusion for deep multi-view clustering. *IEEE Transactions on Multimedia*.
- Wu, X.; Huang, S.; Wang, W.; Ma, S.; Dong, L.; and Wei, F. 2024b. Multi-head mixture-of-experts. *Advances in Neural Information Processing Systems*, 37: 94073–94096.
- Xu, J.; Ren, Y.; Li, G.; Pan, L.; Zhu, C.; and Xu, Z. 2021. Deep embedded multi-view clustering with collaborative training. *Information Sciences*, 573: 279–290.
- Xu, J.; Ren, Y.; Tang, H.; Yang, Z.; Pan, L.; Yang, Y.; Pu, X.; Yu, P. S.; and He, L. 2023. Self-Supervised Discriminative Feature Learning for Deep Multi-View Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(7): 7470–7482.
- Xu, J.; Ren, Y.; Wang, X.; Feng, L.; Zhang, Z.; Niu, G.; and Zhu, X. 2024. Investigating and mitigating the side effects of noisy views for self-supervised clustering algorithms in practical multi-view scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22957–22966.

Xu, J.; Tang, H.; Ren, Y.; Peng, L.; Zhu, X.; and He, L. 2022. Multi-Level Feature Learning for Contrastive Multi-View Clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16051–16060.

Yan, W.; Zhang, Y.; Lv, C.; Tang, C.; Yue, G.; Liao, L.; and Lin, W. 2023. GCFAgg: Global and Cross-View Feature Aggregation for Multi-View Clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19863–19872.

Yan, W.; Zhang, Y.; Tang, C.; Zhou, W.; and Lin, W. 2024. Anchor-Sharing and Cluster-Wise Contrastive Network for Multiview Representation Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2): 3797–3807.

Yang, X.; Jiaqi, J.; Wang, S.; Liang, K.; Liu, Y.; Wen, Y.; Liu, S.; Zhou, S.; Liu, X.; and Zhu, E. 2023. Dealmvc: Dual contrastive calibration for multi-view clustering. In *Proceedings of the 31st ACM international conference on multimedia*, 337–346.

Yu, X.; Jiang, Y.; Chao, G.; and Chu, D. 2025. Deep Contrastive Multi-View Subspace Clustering With Representation and Cluster Interactive Learning. *IEEE Transactions on Knowledge and Data Engineering*, 37(1): 188–199.

Zhang, P.; Liu, X.; Xiong, J.; Zhou, S.; Zhao, W.; Zhu, E.; and Cai, Z. 2022. Consensus One-Step Multi-View Subspace Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 34(10): 4676–4689.

Zhou, R.; and Shen, Y.-D. 2020. End-to-end adversarial-attention network for multi-modal clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14619–14628.