Every Node is Different: Dynamically Fusing Self-Supervised Tasks for Attributed Graph Clustering

Pengfei Zhu, Qian Wang, Yu Wang*, Jialu Li, Qinghua Hu

College of Intelligence and Computing, Tianjin University, Tianjin, China
{zhupengfei, wang_qian0806, wang.yu, jialuli, huqinghua}@tju.edu.cn

Abstract

Attributed graph clustering is an unsupervised task that partitions nodes into different groups. Self-supervised learning (SSL) shows great potential in handling this task, and some recent studies simultaneously learn multiple SSL tasks to further boost performance. Currently, different SSL tasks are assigned the same set of weights for all graph nodes. However, we observe that some graph nodes whose neighbors are in different groups require significantly different emphases on SSL tasks. In this paper, we propose to dynamically learn the weights of SSL tasks for different nodes and fuse the embeddings learned from different SSL tasks to boost performance. We design an innovative graph clustering approach, namely Dynamically Fusing Self-Supervised Learning (DyFSS). Specifically, DyFSS fuses features extracted from diverse SSL tasks using distinct weights derived from a gating network. To effectively learn the gating network, we design a dual-level self-supervised strategy that incorporates pseudo labels and the graph structure. Extensive experiments on five datasets show that DyFSS outperforms the state-of-the-art multi-task SSL methods by up to 8.66% on the accuracy metric. The code of DyFSS is available at: https://github.com/q086/DyFSS.

Introduction

Attributed Graph Clustering (AGC) aims to partition the graph nodes into several groups without explicit supervision. A key challenge of AGC is the lack of proper supervision (Zhu et al. 2022; Mrabah et al. 2023). Self-supervised learning (SSL) finds supervision from the data and has been shown to be greatly helpful for obtaining good performance (Wu et al. 2023).

Many studies on designing SSL methods in AGC have been proposed. Generative learning-based methods (Pan et al. 2018; Hou et al. 2022) aim to reconstruct graph structure or masked node features to learn node representations. Contrastive learning-based methods assist clustering by maximizing the mutual information of positive samples (Velickovic et al. 2019). Predictive learning-based methods construct pretext tasks by exploiting natural supervision provided by the graph data itself (You et al. 2020). Despite showing effectiveness, the advantage of using a single SSL task is limited due to ignoring helpful information from other tasks (Jin et al. 2022).

To address the problem, several methods are proposed to simultaneously learn multiple graph SSL tasks and show promising results. (Jin et al. 2022) proposed an AUTOSSL model that automatically searches the weights of SSL tasks in a multi-task learning framework by maximizing the designed pseudo-homophily to learn node embeddings. (Ju et al. 2023) designed a PARETOGNN model that minimizes the potential conflicts between different SSL tasks to learn better node representations.

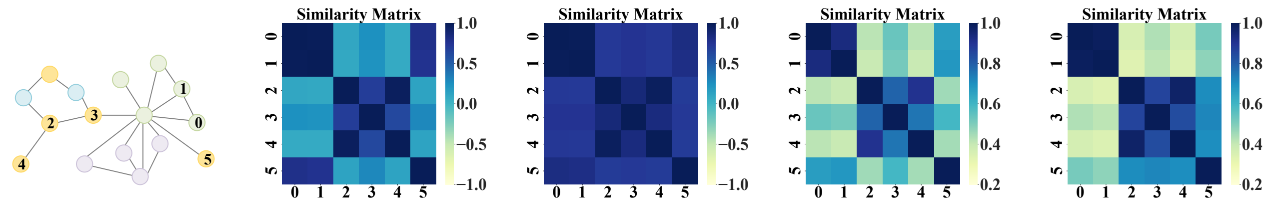
Although these methods can automatically learn the weights of various SSL tasks, they are globally shared in the whole graph. This leads to a quite straightforward concern: Is a fixed strategy of fusing different SSL tasks appropriate for learning good representations of all the graph nodes?

We empirically explore and analyze the question on the Cora dataset. Fig. 1(a) displays a subgraph of Cora, in which nodes of the same classes are annotated in the same color. We visualize the similarity matrix of nodes obtained by different SSL tasks, *i.e.,* node attributes clustering (CLU) and graph partition (PAR) SSL tasks (You et al. 2020), respectively, in Figs. 1(b) and (c). CLU encourages the encoder to extract attribute information of nodes to assist the AGC task. In contrast, PAR emphasizes the learning of local topology information by the encoder. In Fig. 1, we observe that node representations learned from different SSL tasks show different similarity relationships between nodes, and both of them can not reflect the true clustering structure and lead to unsatisfactory clustering results.

Furthermore, we use AUTOSSL to learn the node embeddings by automatically searching the weights of PAR and CLU tasks. The visualization of its corresponding similarity matrix is shown in Fig.1(d). To learn proper weights for most nodes, AUTOSSL prefers to use the topology information extracted from the PAR task, making the distribution of the similarity matrix roughly the same as the similarity matrix obtained by the PAR task. Due to all nodes sharing a set of SSL task weights, AUTOSSL can not learn the intrinsic clustering structure.

Intuitively, to obtain better clustering results, Nodes #2-#5 should use more attribute information, while Nodes #0-#1 should use more topology information. Therefore, we ob-

*Corresponding author



(a) (b) (c) (d) (e)

Figure 1: (a): A subgraph of Cora dataset. The same classes of nodes are annotated in the same color, and the number denotes the node index. (b)-(e): The heat maps of similarity matrices in latent space with node embeddings obtained by graph partition (PAR) SSL task, node attributes clustering (CLU) SSL task, AUTOSSL, and dynamic fusion operation.

tain the node-wise fusion representations by weighing two kinds of features and subsequently visualize the fusion representations. As shown in Fig. 1(e), it can clearly reflect the clustering structure. From this experiment, we empirically find that each node in the graph is different and it should adaptively fuse features extracted from various SSL tasks to learn discriminative node representations for clustering.

To tackle the above problem, we propose a novel framework, Dynamically Fusing Self-Supervised Learning (DyFSS), to fully utilize various graph self-supervised information for node-wise representation construction. Specifically, a dynamic fusion network based on the Mixture of Expert (MoE) framework is carefully designed for fusing features extracted from multiple SSL tasks. In this design, each SSL task is assigned to an expert to extract task-specific features. Each node can adaptively select a set of weights by a gating network and perform feature fusion. Such adaptability facilitates learning discriminative node representations. After that, we calculate the clustering distribution of fusion embeddings by using student's t-distribution to achieve end-to-end clustering. In the initial training stage, clustering alignment loss is unreliable owing to the subpar quality of fused features. To address this, we design a dual-level self-supervised strategy (*i.e.,*, pseudo-label level, and graph structure level). This innovative strategy offers effective guidance for the fusion network, thereby enhancing node representations. Overall, the contributions of this paper are summarized as follows:

- We find the problem that the fusion strategy shared by all the nodes is not appropriate for learning node embeddings for clustering. Accordingly, we propose a multi-task SSL graph clustering framework that dynamically fuses the features extracted from multiple SSL tasks for each node using distinct weights derived from a gating network.
- To achieve reliable dynamic fusion, we design a dual-level supervised strategy to train the fusion network. The pseudo-label supervised information facilitates distinguishing different clusters, while the graph structure supervised information enhances the reliability of the fusion embeddings by capturing the underlying connections between nodes.
- Extensive experiment results on five public benchmark datasets verify that DyFSS can effectively boost the clustering performance by combing various SSL tasks. Our experiments show that DyFSS improves the accuracy metric by up to 8.66% on the Photo dataset over the state-of-the-art multi-task SSL methods.

Related Work

Self-Supervised Learning in GNNs

Graph Neural Networks (GNNs) possess robust representation capabilities in analyzing graph-structured data, while they rely on expensive task-specific labels. Many self-supervised learning methods are proposed to alleviate the demand for labeled data in graph-related tasks.

Generative SSL techniques strive to reconstruct the adjacency matrix, masked node features, or both, as demonstrated in studies like (Pan et al. 2018; Hou et al. 2022). Contrastive approaches (Velickovic et al. 2019) leverage positive and negative samples to acquire node representations. Predictive methodologies, such as (Peng et al. 2020; You et al. 2020), formulate pretext tasks using inherent supervisory signals derived from the graph data.

Recently, (Jin et al. 2022) proposed AUTOSSL, and (Ju et al. 2023) proposed PARETOGNN, have emerged as approaches that leverage multiple SSL tasks to enhance node representations and achieve improved generalization performance in downstream tasks. Both methods employ the multi-task learning framework, enabling automatic loss weight adjustment for each SSL task.

Our method designs a dynamic fusion network, which automatically fuses the diversity features extracted from various SSL tasks according to the initial node embeddings of each node for clustering.

Attributed Graph Clustering

Efficiently leveraging varied information inherent in graphs to learn distinctive node representations holds the utmost importance for the AGC task. Benefiting from the rapid development of GNNs, researchers have developed a series of AGC algorithms based on GNNs.

In particular, researchers proposed the graph auto-encoder (Kipf and Welling 2016) along with its variations (Pan et al. 2018; Vaibhav, Huang, and Frederking 2019). These methods aim to obtain node representations through graph data

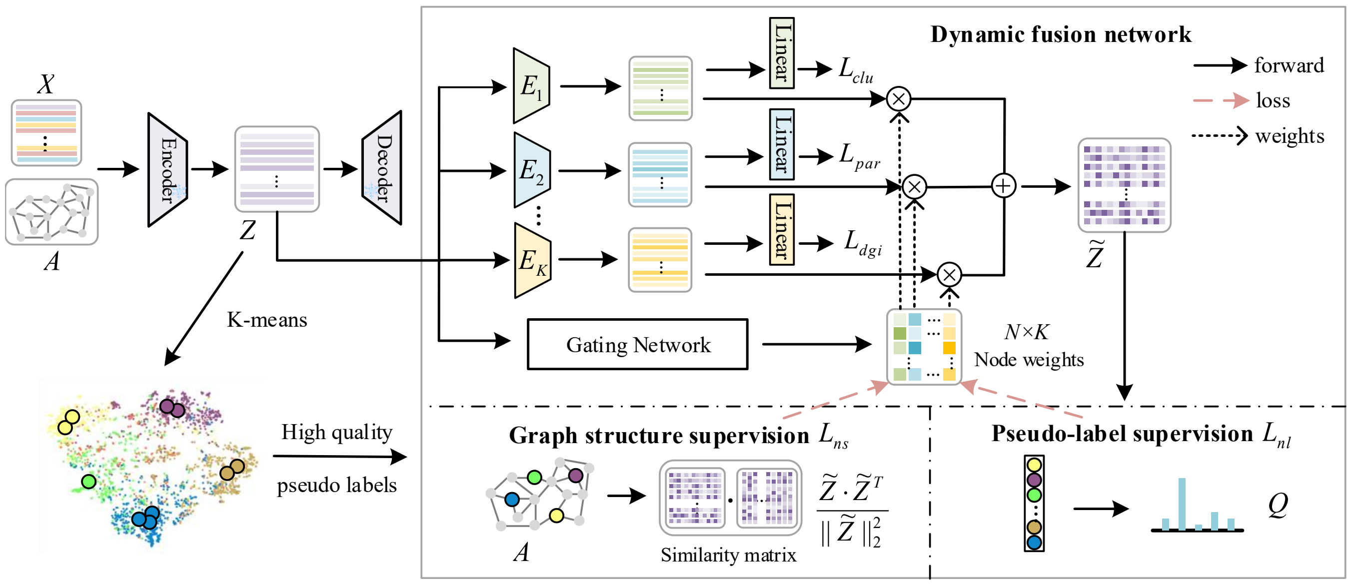


Figure 2: Architecture of DyFSS model. We first use the pre-trained model to obtain the initial node embeddings. The embeddings are then fed into the dynamic fusion network to obtain the fusion embeddings. Specifically, each SSL task is allocated to a task-specific GCN layer (*i.e.,* an expert) to extract features using the corresponding SSL loss, *i.e.,*L_{clu}, L_{par}, L_{dgi}, *etc.* Simultaneously, the gating network generates a set of weights for each node, culminating in the subsequent execution of the feature fusion operation. Lastly, we use high-quality labels and graph structure as supervised information to provide effective guidance for the training of the dynamic fusion network.

reconstruction, followed by the K-means algorithm to yield clustering results. Building upon this, the deep clustering network proposed by (Wang et al. 2019) achieves simultaneously learn node representations and cluster assignments. Additionally, some methods (Xia et al. 2022; Yang et al. 2023) also use contrastive learning to guide the clustering process learning by comparing the similarities and differences between nodes.

Differing from these approaches, DyFSS uses various information from multiple SSL tasks for clustering. Furthermore, we design a dual-level supervised strategy to enhance the node representations.

Method

Notations and Overview

Given an attributed graph $G' = \{V', E', \boldsymbol{X}\}$, where $V' = \{v_1, v_2, ..., v_N\}$ representations the node set with N nodes, E' is the edge set of the graph, $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_N}\} \in \mathbb{R}^{N\times d}$ is the initial feature matrix, and $\boldsymbol{x_i} = \boldsymbol{X}[i,:] \in \mathbb{R}^d$ denotes the node feature of v_i. The topological structure of graph G' is represented by an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, where $a_{ij} = \boldsymbol{A}[i,j] = 1$ if $(v_i, v_j) \in E'$, otherwise $a_{ij} = 0$. To keep the scale of the feature vectors unchanged, adjacency matrix $\boldsymbol{A}$ needs to be normalized through calculating $\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$, where $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix with self-connection, $\widetilde{\boldsymbol{D}} = \sum_j \tilde{a}_{ij}$ is the degree matrix of $\widetilde{\boldsymbol{A}}$, and $\boldsymbol{I}$ is the identity matrix.

In this section, we will introduce the novel AGC method named Dynamically Fusion Self-Supervised Learning (DyFSS), which aims to adaptively fuse the features extracted from multiple SSL tasks for clustering. As illustrated in Fig. 2, DyFSS mainly consists of two components, *i.e.,* a dynamic fusion network and a dual-level of self-supervised strategy module. Next, we will introduce the proposed two modules and the final network objectives in detail.

Dynamic Fusion Network

Achieving an adaptive node-wise fusion of features from diverse SSL tasks becomes crucial to enhance node representations for clustering. The existing multi-task learning frameworks lack the ability to achieve dynamic node-wise fusion. The Mixture of Experts (MoE) (Shazeer et al. 2017) network is an ensemble learning technique, which can address this limitation.

Therefore, we propose the dynamic fusion network, leveraging the MoE framework to fuse the features extracted from multiple SSL tasks for each node. Specifically, the dynamic fusion network comprises K expert networks $E_1, ..., E_K$, alongside a gating network G that produces a K-dimensional vector. In this design, each expert is assigned to an individual SSL task, responsible for extracting task-specific features.

Expert Network. Each expert network is a GCN layer that aggregates information from neighbors of each node. For the k^{th} expert, the output $\boldsymbol{Z_k} \in \mathbb{R}^{N\times d'}$ is formulated as

$$\boldsymbol{Z_k} = f_{linear}(\boldsymbol{Z}, \boldsymbol{A}|\boldsymbol{W^{(k)}}) = \phi(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{Z}\boldsymbol{W^{(k)}}), \tag{1}$$

where $\boldsymbol{Z} \in \mathbb{R}^{N\times d'}$ is the latent node representations obtained by the pre-trained model, and $\boldsymbol{W^{(k)}}$ is the parameter to be trained. For node v_i, the output of k^{th} expert is represented as $E_k(\boldsymbol{z_i}) = \boldsymbol{Z_k}[i,:] \in \mathbb{R}^{d'}$.

Given the presence of K SSL tasks, each expert is designated to an individual SSL task, functioning as an encoder responsible for extracting task-specific features to facilitate fusion. The graph encoder of k^{th} SSL task is represented as $f_{\theta_g^{(k)}}(\cdot;\theta_g^{(k)})$, where the $\theta_g^{(k)} = \boldsymbol{W^{(k)}}$, and it is obvious that $f_{\theta_g^{(k)}}(G') = \boldsymbol{Z_k}$. The loss function k^{th} task is represented as $L_k(G';\theta_g^{(k)},\theta_t^{(k)})$, where the $\theta_t^{(k)}$ refers to the task-specific parameters (*i.e.,* a linear layer). Then, we minimize the total loss of all SSL tasks, enabling the extraction of task-specific features for each expert,

$$L_{ssl} = \sum_{k=1}^{K} L_k(G';\theta_g^{(k)},\theta_t^{(k)}). \tag{2}$$

Gating Network. Utilizing the gating network, we derive a set of expert weights for each node based on its corresponding initial node embedding, and it is formulated as:

$$G(\boldsymbol{z_i}) = softmax(\boldsymbol{z_i}\cdot \boldsymbol{W_n}), \tag{3}$$

where the $\boldsymbol{z_i} = \boldsymbol{Z}[i,:]$ is the initial node embeddings of node v_i, $\boldsymbol{W_n}$ is the trainable parameter, and $softmax(\cdot)$ is a function that converts the vector into weights for each SSL expert.

Node-wise Feature Fusion. For the node v_i, with the output of each expert $E_1(\boldsymbol{z_i}),...,E_K(\boldsymbol{z_i})$ and the output of the gating network $G(\boldsymbol{z_i})$, a linear combination operation is used to fuse the features from different experts,

$$\tilde{\boldsymbol{z}}_{\boldsymbol{i}} = \sum_{k=1}^{K} G(\boldsymbol{z_i})_k E_k(\boldsymbol{z_i}), \tag{4}$$

where the $G(\boldsymbol{z_i})_k$ represents the k^{th} element of $G(\boldsymbol{z_i})$, and $\tilde{\boldsymbol{z}}_{\boldsymbol{i}}$ is the fused embedding of node v_i. Finally, we can obtain the fused node representations $\widetilde{\boldsymbol{Z}} \in \mathbb{R}^{N\times d'}$ of all nodes for clustering.

Dual-Level Self-Supervised Strategy

To train the dynamic fusion network, a conventional approach offers guidance by aligning the distribution between the soft clustering distribution and target distribution (Wang et al. 2019; Bo et al. 2020). However, owing to the subpar quality of fused features in the initial training stage, the clustering alignment loss is unreliable, potentially leading to optimization in the wrong direction.

To address this issue, we design a dual-level self-supervised strategy to train the dynamic fusion network. Some simple samples can learn better embeddings through pretraining, and high-quality pseudo labels generated by these nodes are used as supervised information to distinguish distinct clusters and enhance the compactness of fusion representations within each cluster. In addition, the learned initial node representations may be inaccurate for other difficult samples. Therefore, we use the original adjacency structure of the graph itself as supervised information, which ensures that the representations capture the true underlying connections between nodes, enhancing the reliability of the fused node representations.

Pseudo-Label Level Self-Supervised Strategy. After obtaining latent node representations $\boldsymbol{Z}$, followed by (Wang et al. 2019; Zhu et al. 2022), the student's t-distribution is used to calculate the clustering assignment distribution $\boldsymbol{Q} \in \mathbb{R}^{N\times C}$. Subsequently, the pseudo-label node set M and the corresponding pseudo-label set Y are generated as:

$$M = \left\{ i \,|\, (\max_{j\in\{1,...,C\}} q_{ij}) \geq \gamma, \forall i \in V \right\}, \tag{5}$$

$$Y = \left\{ \operatorname*{argmax}_{j\in\{1,...,C\}} q_{ij} \,|\, \forall i \in V \right\}, \tag{6}$$

where the $\gamma \in (0,1)$ represents the m-th percentile P_m calculated from the maximum clustering scores across all nodes in $\boldsymbol{Q}$.

Next, we calculate the soft clustering distribution $\widetilde{\boldsymbol{Q}} \in \mathbb{R}^{N\times C}$ of fusion embeddings $\widetilde{\boldsymbol{Z}}$. After that, the cross-entropy loss is used to achieve the pseudo-label level supervision of nodes, formulated as:

$$L_{nl} = -\frac{1}{|M|}\sum_{i=1}^{|M|}\sum_{j=1}^{C} \tilde{y}'_{ij} log(\tilde{q}_{ij}), \tag{7}$$

where the $\tilde{y}'_{ij}$ is the one-hot pseudo label of node $v_i \in M$ after alignment between different clustering results from $\boldsymbol{Q}$ and $\widetilde{\boldsymbol{Q}}$.

Graph Structure Level Self-Supervised Strategy. Specifically, after obtaining the fusion embeddings $\widetilde{\boldsymbol{Z}}$, we scale the embeddings to the $[0,1]$ interval by min-max scaler for variance reduction. Subsequently, we utilize the cosine function to calculate the pair-wise similarity matrix $\widetilde{\boldsymbol{S}} \in \mathbb{R}^{N\times N}$ of nodes, formulated as:

$$\widetilde{\boldsymbol{S}} = \frac{\widetilde{\boldsymbol{Z}}\widetilde{\boldsymbol{Z}}^{\boldsymbol{T}}}{||\widetilde{\boldsymbol{Z}}||_2^2}. \tag{8}$$

Then we formulate the graph structure supervision by employing the cross-entropy loss, as follows:

$$L_{ns} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} -\tilde{a}_{ij}log(\tilde{s}_{ij}) - (1-\tilde{a}_{ij})log(1-\tilde{s}_{ij}), \tag{9}$$

where $\tilde{a}_{ij}$ is the element of the adjacency matrix with self-connection $\widetilde{\boldsymbol{A}}$.

The overall learning objective comprises four parts, *i.e.,* the task-specific loss of all SSL tasks, the pseudo-label level loss, the graph structure level loss, and the clustering loss:

$$L = L_{nl} + L_{ns} + \lambda_1 L_{ssl} + \lambda_2 L_{pq}, \tag{10}$$

where λ_1 and λ_2 are hyper-parameters that balance the weight of different losses. L_{pq} is the clustering alignment loss (Wang et al. 2019; Xie, Girshick, and Farhadi 2016; Guo et al. 2017). The detailed learning procedure of our method is shown in Algorithm 1.

Dataset	Type	Cora			Citeseer			Photo			Computers			CS		
		ACC	NMI	F1	ACC	NMI	F1	ACC	NMI	F1	ACC	NMI	F1	ACC	NMI	F1
K-Means	-	33.64	13.76	23.39	58.04	31.64	55.23	40.55	28.92	24.37	34.36	17.41	13.10	58.13	64.27	39.68
AE	S-F	43.06	24.15	42.84	53.65	26.43	50.44	42.65	35.51	40.36	31.39	31.30	25.79	55.14	56.35	37.51
DeepWalk	S-G	54.74	31.82	54.48	33.63	12.13	33.47	76.14	66.72	70.26	50.68	49.37	**47.75**	54.69	54.51	52.70
VGAE	S-F&G	61.67	46.10	57.67	54.19	28.64	52.43	54.00	49.49	53.18	44.28	39.48	37.77	61.67	69.09	55.49
ARVGA	S-F&G	63.80	45.00	62.70	54.40	26.10	52.90	43.24	32.77	34.84	28.27	26.81	18.62	64.44	71.62	59.03
DAEGC	S-F&G	67.02	48.22	66.19	62.73	37.82	60.64	62.27	58.88	60.35	52.46	46.38	42.22	68.65	71.23	54.31
GraphMAE	S-F&G	62.78	54.75	63.74	68.95	43.74	64.25	75.42	67.25	69.01	53.04	52.66	40.76	59.83	70.85	49.27
DGI	S-F&G	69.90	54.95	66.95	69.13	44.48	64.36	44.07	36.74	36.56	39.96	35.11	25.02	69.45	76.97	66.27
GCA	S-F&G	60.86	48.16	57.94	66.97	41.53	63.25	63.19	51.80	52.10	49.43	43.48	38.95	66.98	74.34	64.87
CCA-SSG	S-F&G	64.73	54.63	54.85	67.93	42.89	61.04	69.59	57.88	66.21	47.59	44.56	42.69	73.31	73.64	58.98
BGRL	S-F&G	69.28	54.91	67.15	64.20	38.40	59.78	75.01	65.68	71.41	46.94	44.12	41.45	69.82	76.46	71.52
AUTOSSL	M-F&G	69.57	52.82	67.21	69.16	44.59	64.37	68.17	52.76	63.31	50.58	44.62	37.36	68.92	**76.98**	66.17
PARETOGNN	M-F&G	68.55	52.17	59.17	68.27	43.28	61.30	71.15	64.02	66.77	47.18	47.09	46.69	69.82	70.49	65.34
DyFSS	M-F&G	**72.19**	**55.49**	**68.09**	**70.18**	**44.80**	**64.68**	**79.81**	**71.86**	**73.56**	**56.95**	**53.68**	46.78	**75.81**	76.80	**72.69**

Table 1: Clustering performance on five datasets. The bold and underlined values indicate the best and the runner-up results, respectively. The notation S-F represents models that leverage single SSL task with node features as inputs. The notation M-F&G represents models that concurrently exploit both node features and graph structure as inputs and utilize varied information from multiple SSL tasks.

Experiments

Experiment Setup

Benchmark Datasets and Evaluation Metrics. We conducted experiments on five widely used real-world datasets in the literature (Jin et al. 2022; Zhang et al. 2021), including Cora, Citeseer, Photo, Computers, and CS. Three widely used metrics were employed to evaluate the clustering results: Accuracy (ACC), Normalized Mutual information (NMI), and macro F1-score (F1).

Algorithm 1: DyFSS

Input:Graph $G' = (\boldsymbol{X}, \boldsymbol{A})$. Pseudo-label node set M and pseudo-label set Y. Initial node embeddings $\boldsymbol{Z}$. Iteration number I. Hyper-parameters λ_1, λ_2. Number of classes C.
Output: Clustering results R.

Generate the labels of five SSL tasks.
Initialize the parameters of K experts and the gating network to obtain $\widetilde{\boldsymbol{Z}}$.
Initialize the clustering centers and clustering results O with K-means based on $\widetilde{\boldsymbol{Z}}$.
Align the pseudo labels set Y with the initial clustering results O.
for $i = 1, 2, ..., I$ **do**
    Update $\widetilde{\boldsymbol{Z}}$ by Eq.(1), Eq.(3) and Eq.(4).
    Calculate soft assignment distributions $\widetilde{\boldsymbol{Q}}$ and pairwise similarity matrix $\widetilde{\boldsymbol{S}}$ by Eq.(8).
    Calculate L_{nl}, L_{ns}, L_{ssl}, and L_{pq} respectively.
    Update the whole network by minimizing Eq.(10).
end for
Obtain the clustering results R by final soft clustering distribution $\widetilde{\boldsymbol{Q}}$.
return R.

Baseline Methods. We compared our proposed method with thirteen baselines to evaluate the effectiveness of our method. K-means (Hartigan and Wong 1979) and AE proposed by (Yang et al. 2017) are methods that only use node features for clustering, while DeepWalk proposed by (Perozzi, Al-Rfou, and Skiena 2014) only uses the graph structure to learn node embeddings. Additionally, some GCN-based baseline methods use both node attributes and graph structure to learn node representations. VGAE (Kipf and Welling 2016), ARVGA (Pan et al. 2018), DAEGC (Wang et al. 2019), and GraphMAE (Hou et al. 2022) are generative methods. DGI (Velickovic et al. 2019), GCA (Zhu et al. 2021), CCA-SSG (Zhang et al. 2021), and BGRL (Thakoor et al. 2022) are contrastive methods. Furthermore, we compared the clustering results with the state-of-the-art multi-task self-supervised learning methods, *i.e.,* AUTOSSL (Jin et al. 2022) and ParetoGNN (Ju et al. 2023).

Basic SSL task and Training Procedure. For SSL tasks, we chose the five SSL pretext tasks adopted by AUTOSSL, including PAR, CLU, PAIRDIS, PAIRSIM, and DGI. The training of DyFSS includes two stages. Firstly, we pretrained the base model to obtain node representations $\boldsymbol{Z}$ for at least 100 epochs with the multi-task framework by minimizing reconstruction loss and self-supervised loss of five pretext tasks. Thereafter, we trained the fusion clustering network with node representations $\boldsymbol{Z}$ under the guidance of Eq.(10) for at least 200 epochs until convergence. Our method was implemented with the PyTorch platform, an NVIDIA 1070 GPU, and an NVIDIA 3090 GPU.

Parameters Setting. For PARETOGNN, we reproduced their model by using the same five SSL tasks as our method and following the parameter settings of the original paper. For AUTOSSL, we use the AUTOSSL-DS strategy to save the search cost. For other baselines, we reproduced their

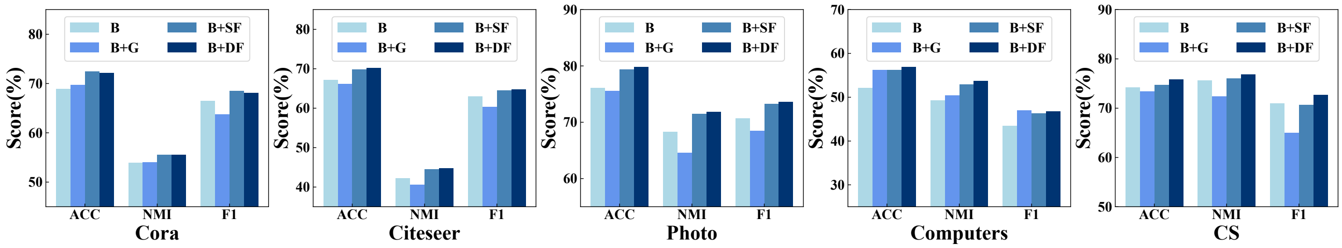


Figure 3: Ablation studies of dynamic fusion network. The baseline (B) is a per-trained ARVGA model, and B+DF is our method. The B+G indicates the baseline involving dynamic gating mechanisms and fixed experts. Conversely, B+SF signifies the baseline that uses a fixed gating network in combination with dynamic experts, representing a form of static fusion approach.

source code by following the setting of the original literature and reported corresponding clustering results. For our method, we used the ARVGA as the base model, and the encoder of it applied a two-layer GCN with 256 and 128 filters. All ablation studies were trained with the Adam optimizer, and the learning rate was set to 1e-3. For trade-off hyper-parameters, we set λ_2 to 1. Furthermore, λ_1 was set to 0.01 for Photo and Computers datasets and 0.1 for others. Moreover, m was set to 40 for the Citeseer dataset and 50 for others.

Clustering Performance Comparisons

The comparative clustering performance of our method and thirteen baselines across five benchmark datasets is outlined in Tab. 1. It can be observed from these results that:

(1) The node-wise dynamic fusion operation boosts the clustering performance compared to the multi-task SSL learning methods. Different from AUTOSSL and PARETOGNN, DyFSS achieves node-wise dynamic fusion operation by integrating features from various SSL tasks. Compared with these methods, DyFSS obtains remarkable performances. For example, DyFSS outperforms AUTOSSL by 2.62%, 2.67%, and 0.88% on the Cora dataset, with respect to ACC, NMI, and F1, respectively. The experiments show the effectiveness of DyFSS, underscoring its efficacy in enhancing feature representations.

(2) Fully utilizing various information from multiple SSL tasks facilitates the clustering task. Compared with the methods that only use a single SSL task, the methods utilizing multiple SSL tasks demonstrate a significant performance advantage. Additionally, for different datasets, the contribution of various information to the AGC task varies. We observe that DeepWalk outperforms AE by 33.49% on Photo, while AE outperforms DeepWalk by 20.02% on Citeseer, in terms of ACC. It shows that the Photo dataset prefers structure information while Citeseer prefers attribute information for clustering. Therefore, it is necessary to utilize various information from SSL tasks to assist in clustering for different datasets.

(3) Algorithms that incorporate both feature and structure information tend to yield superior clustering results compared to methods that only leverage one source of information. Across the majority of datasets, GCN-based methods exhibit superior performance compared with AE and DeepWalk.

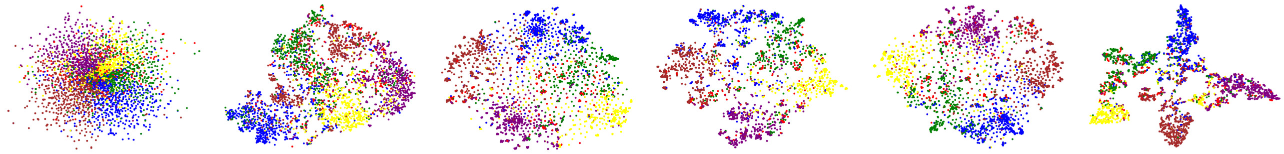
Datasets	Model	ACC	NMI	F1
Cora	DyFSS w/o L_{dual}	70.75	55.40	66.99
	DyFSS w/o L_{ns}	71.34	55.04	67.28
	DyFSS	**72.19**	**55.49**	**68.09**
Citeseer	DyFSS w/o L_{dual}	68.53	43.73	64.19
	DyFSS w/o L_{ns}	69.37	43.92	63.90
	DyFSS	**70.18**	**44.80**	**64.68**
Photo	DyFSS w/o L_{dual}	77.83	70.28	72.72
	DyFSS w/o L_{ns}	78.36	70.19	72.41
	DyFSS	**79.81**	**71.86**	**73.56**
Computers	DyFSS w/o L_{dual}	47.96	52.18	45.14
	DyFSS w/o L_{ns}	55.86	52.24	45.71
	DyFSS	**56.95**	**53.68**	**46.78**
CS	DyFSS w/o L_{dual}	75.56	**76.81**	72.52
	DyFSS w/o L_{ns}	75.60	76.39	71.63
	DyFSS	**75.81**	76.80	**72.69**

Table 2: Ablation comparisons of the dual-level supervised strategy L_{dual} which includes pseudo-label level supervised information L_{nl} and graph structure level supervised information L_{ns}.

Ablation Studies

Effectiveness of Dynamically Fusion Network. In this part, we conducted ablation studies to verify the effectiveness of the dynamic fusion network and reported the results in Fig. 3. From the figures, we can observe that B+DF and B+SF consistently outperform the B and B+G methods, which shows that fixed experts are not conducive to clustering and it is necessary to learn the multiple features for different experts. Additionally, the B+DF method improves the clustering performance over the B+SF method on four of five datasets. Both of the observations collectively demonstrate that the novel dynamic fusion network possesses the capability to extract a range of diverse features and dynamically leverage these features to generate node representations, leading to improved clustering decisions.

Effectiveness of Dual-Level Supervised Strategy. To verify the superiority of the dual-level supervised strategy, we compared our method with the other two counterparts. The results of these methods are summarized in



(a) Raw data (b) VGAE (c) DGI (d) GraphMAE (e) AUTOSSL (f) DyFSS

Figure 4: t-SNE visualization on Citeseer dataset.

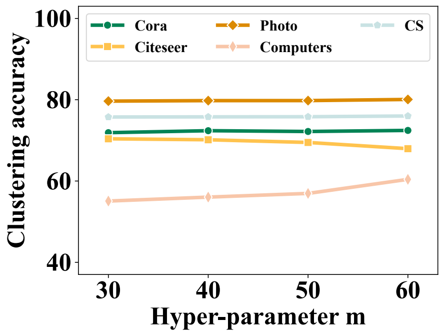


Figure 5: The sensitivity analysis of DyFSS with variation of hyper-parameter m on five datasets.

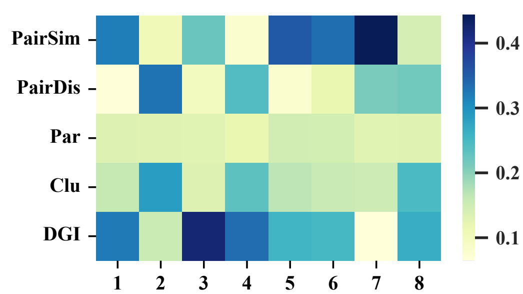


Figure 6: Visualization of the weights assigned by five SSL experts on the Citeseer dataset. The x-axis shows eight randomly selected nodes and the y-axis presents five SSL tasks.

Tab. 2. In terms of ACC, DyFSS w/o L_{ns} outperforms the DyFSS w/o L_{dual} by 0.84%, 0.53%, and 7.90% on Citeseer, Photo, and Computers datasets, respectively. Additionally, DyFSS demonstrates even further advancements, surpassing the DyFSS w/o L_{ns} method by 0.88%, 1.67%, and 1.44% on the same respective datasets, in terms of NMI. Based on the results, it is evident that the proposed dual-level supervised strategy avoids the problem of clustering alignment loss deviating from the optimal optimization direction, and each kind of supervised information contributes to the final performance. Notably, the pseudo-label supervised information aids in distinguishing diverse clusters and enhancing the compactness of node representations within each cluster. Concurrently, the guidance from graph structure supervision complements the pseudo-label supervised information, facilitating learning reliable fusion embeddings.

Further Analysis

Hyper-Parameter Analysis of m. We investigate the influence of hyper-parameter m, which plays a pivotal role in establishing the threshold γ for the generation of pseudo labels. We conduct experiments to show the effect of this parameter on all datasets. Based on the results illustrated in Fig. 5, it can be observed that DyFSS is insensitive to hyper-parameter m. Except for the Citeseer dataset, the accuracy metric has slight increments with the increasing value m. Notably, our method demonstrates stable performance across a wide range of m.

Visualization. For an intuitive demonstration of the effectiveness of our method, we employ the t-SNE algorithm (van der Maaten and Hinton 2008) on the Citeseer dataset. This allows us to visualize the distribution of the raw data, and the learned node embeddings produced by four baseline methods, alongside our DyFSS model, in a two-dimensional space. As shown in Fig. 4, the visual results clearly demonstrate DyFSS's enhanced capability in revealing the inherent clustering structure within the data and improving the compactness of learned node representations within clusters.

Furthermore, we present a visualization of the weights assigned to five SSL tasks involving different nodes. Illustrated in Fig. 6, diverse nodes obtain varying sets of weights, underscoring the ability of the dynamic fusion network to generate a set of weights for each node flexibly.

Conclusion

To solve the problem that all the nodes share the same set of SSL weights, we proposed a novel Dynamically Fusing Self-Supervised Learning (DyFSS) method for the attributed graph clustering task. This method enables every node within the graph to dynamically fuse distinct features derived from various SSL tasks, yielding discriminative node representations. Additionally, the designed dual-level supervised strategy provides effective guidance for the training of the proposed dynamic fusion network, which facilitates distinguishing different clusters and enhancing the compactness of node embeddings within each cluster. Experiments on five benchmark datasets corroborate the effectiveness of our method in boosting the clustering performance. Furthermore, our method is insensitive to the hyper-parameter and the learned node representations can reflect the intrinsic clustering structure.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2022ZD0116500 and in part by the National Natural Science Foundation of China under Grants 62106174, 62222608, 62266035, 61925602, U23B2049, and 62076179.

References

Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; and Cui, P. 2020. Structural Deep Clustering Network. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 1400–1410. ACM / IW3C2.

Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017. Improved Deep Embedded Clustering with Local Structure Preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 1753–1759. ijcai.org.

Hartigan, J. A.; and Wong, M. A. 1979. A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1): 100–108.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; and Tang, J. 2022. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, 594–604. ACM.

Jin, W.; Liu, X.; Zhao, X.; Ma, Y.; Shah, N.; and Tang, J. 2022. Automated Self-Supervised Learning for Graphs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 1–13. OpenReview.net.

Ju, M.; Zhao, T.; Wen, Q.; Yu, W.; Shah, N.; Ye, Y.; and Zhang, C. 2023. Multi-task Self-supervised Graph Neural Networks Enable Stronger Task Generalization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 1–13. OpenReview.net.

Kipf, T. N.; and Welling, M. 2016. Variational Graph Auto-Encoders. *CoRR*, abs/1611.07308.

Mrabah, N.; Bouguessa, M.; Touati, M. F.; and Ksantini, R. 2023. Rethinking Graph Auto-Encoder Models for Attributed Graph Clustering (Extended abstract). In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, 3891–3892. IEEE.

Pan, S.; Hu, R.; Long, G.; Jiang, J.; Yao, L.; and Zhang, C. 2018. Adversarially Regularized Graph Autoencoder for Graph Embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*.

Peng, Z.; Dong, Y.; Luo, M.; Wu, X.; and Zheng, Q. 2020. Self-Supervised Graph Representation Learning via Global Context Prediction. *CoRR*, abs/2003.01604.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 701–710. ACM.

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.

Thakoor, S.; Tallec, C.; Azar, M. G.; Azabou, M.; Dyer, E. L.; Munos, R.; Velickovic, P.; and Valko, M. 2022. Large-Scale Representation Learning on Graphs via Bootstrapping. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Vaibhav; Huang, P.; and Frederking, R. E. 2019. RWR-GAE: Random Walk Regularization for Graph Auto Encoders. *CoRR*, abs/1908.04003.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 1–13. OpenReview.net.

Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; and Zhang, C. 2019. Attributed Graph Clustering: A Deep Attentional Embedding Approach. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 3670–3676. ijcai.org.

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

Xia, W.; Wang, Q.; Gao, Q.; Yang, M.; and Gao, X. 2022. Self-consistent Contrastive Attributed Graph Clustering with Pseudo-label Prompt. *IEEE Transactions on Multimedia*, 1–13.

Xie, J.; Girshick, R. B.; and Farhadi, A. 2016. Unsupervised Deep Embedding for Clustering Analysis. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 478–487. JMLR.org.

Yang, B.; Fu, X.; Sidiropoulos, N. D.; and Hong, M. 2017. Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 3861–3870. PMLR.

Yang, X.; Liu, Y.; Zhou, S.; Wang, S.; Tu, W.; Zheng, Q.; Liu, X.; Fang, L.; and Zhu, E. 2023. Cluster-Guided Contrastive Graph Clustering Network. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023,* *Washington, DC, USA, February 7-14, 2023*, 10834–10842. AAAI Press.

You, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. When Does Self-Supervision Help Graph Convolutional Networks? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 10871–10880. PMLR.

Zhang, H.; Wu, Q.; Yan, J.; Wipf, D.; and Yu, P. S. 2021. From Canonical Correlation Analysis to Self-supervised Graph Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 76–89.

Zhu, P.; Li, J.; Wang, Y.; Xiao, B.; Zhao, S.; and Hu, Q. 2022. Collaborative Decision-Reinforced Self-Supervision for Attributed Graph Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 2069–2080. ACM / IW3C2.