Deep Attention-Guided Graph Clustering With Dual Self-Supervision

Zhihao Peng, Hui Liu, Yuheng Jia, Member, IEEE, and Junhui Hou, Senior Member, IEEE

Abstract—Existing deep embedding clustering methods fail to sufficiently utilize the available off-the-shelf information from feature embeddings and cluster assignments, limiting their performance. To this end, we propose a novel method, namely deep attention-guided graph clustering with dual self-supervision (DAGC). Specifically, DAGC first utilizes a heterogeneity-wise fusion module to adaptively integrate the features of the auto-encoder and the graph convolutional network in each layer and then uses a scale-wise fusion module to dynamically concatenate the multi-scale features in different layers. Such modules are capable of learning an informative feature embedding via an attention-based mechanism. In addition, we design a distribution-wise fusion module that leverages cluster assignments to acquire clustering results directly. To better explore the off-the-shelf information from the cluster assignments, we develop a dual self-supervision solution consisting of a soft self-supervision strategy with a Kullback-Leibler divergence loss and a hard self-supervision strategy with a pseudo supervision loss. Extensive experiments on nine benchmark datasets validate that our method consistently outperforms state-of-the-art methods. Especially, our method improves the ARI by more than 10.29% over the best baseline. The code will be publicly available at https://github.com/ZhihaoPENG-CityU/DAGC.

Index Terms—Unsupervised learning, deep embedding clustering, feature fusion, self-supervision.

I. INTRODUCTION

CLUSTERING is one of the fundamental tasks in data analysis, which aims to categorize samples into multiple groups according to their intrinsic similarities, and has been successfully applied to many real-world applications such as image processing [1], [2], [3], [4], [5], face recognition [6], [7], [8], and object detection [9], [10], [11]. Recently, with the booming of deep learning, numerous researchers have paid attention to deep embedding clustering analysis, which could effectively learn a clustering-friendly representation by extracting intrinsic patterns from the latent embedding space. For example, Hinton and Salakhutdinov [12] developed a deep auto-encoder (DAE) framework that first conducts embedding learning and then performs K-means [13] to obtain clustering results. Xie et al. [14] designed a deep embedding clustering method (DEC) to perform embedding learning and cluster assignment jointly. Guo et al. [15] improved DEC by introducing a reconstruction loss to preserve data structure. Although these DAE-based approaches obtain impressive improvement, they neglect the underlying topological structure among data, which has demonstrated its importance in various works [16], [17], [18].

Recently, a series of works have been proposed to use graph convolutional networks (GCNs) [19] to exploit the topological structure information. For instance, Kipf et al. [20] incorporated GCN into DAE and variational DAE, and proposed graph auto-encoder (GAE) and variational graph auto-encoder (VGAE), respectively. Pan et al. [21] designed an adversarially regularized graph auto-encoder network (ARGA) to promote GAE. Wang et al. [22] incorporated graph attention networks [23] into GAE for attributed graph clustering. Bo et al. [24] fused GCN into DEC to consider the node content and topological structure information at the same time.

However, these works still suffer from the following drawbacks. First, they equate the importance of the features extracted from DAE and GCN, e.g., in [24], the DAE and GCN features of a typical layer are averaged. Such a simple fusion strategy is not a good choice since those features contain different characteristic information. Second, they neglect the multi-scale information embedded in different layers, which may lead to inferior clustering results. Third, they output two probability distributions capable of obtaining the final clustering results; however, the complex real-world datasets are usually agnostic and vastly different, so it is difficult to decide which one should be used to get the final clustering results. To the best of our knowledge, this is a decision-making dilemma for those kinds of deep graph clustering methods. Last but not least, the previous approaches fail to adequately exploit the available information from the high-confidence clustering assignments.

To address the above-mentioned drawbacks, we propose a novel deep embedding clustering method, focusing on exploiting the available off-the-shelf information from feature...
embeddings and cluster assignments. As shown in Figure 1, the proposed method consists of a heterogeneity-wise fusion (HWF) module, a scale-wise fusion (SWF) module, a distribution-wise fusion (DWF) module, a soft self-supervision (SSS) strategy, and a hard self-supervision (HSS) strategy. A preliminary version of this work was published in ACM Multimedia 2021 [25], which can be regarded as a special case of the current version that focuses on embedding enhancement via the HWF and SWF modules. However, the conference paper suffers from the decision-making dilemma concerning two learned probability distributions from DAE and GCN, i.e., which one should be selected as the final clustering assignment result. In summary, the main contributions of this journal paper are as follows:

• To handle the decision-making dilemma, we propose a learning-aware fusion module to adaptively fuse the learned data probability distributions to predict the clustering results.

• In addition, for the two learned distributions, we improve the soft self-supervision strategy to better preserve the distribution consistency alignment.

• Moreover, for the aforementioned fused distribution, we develop a hard self-supervision strategy with a pseudo supervision loss to employ the high-confidence clustering assignments to improve clustering performance.

• Extensive experiments on nine benchmark datasets validate that our method consistently outperforms the conference version [25]. For instance, on DBLP, our method improves the ARI by more than 14.80%. In addition, the ablation studies and visualizations quantitatively and qualitatively validate the effectiveness of this method.

We organize the rest of this paper as follows. Section II briefly reviews the related works. Section III introduces the proposed network architecture, followed by the experimental results and analyses in Section IV. Finally, we conclude this paper in Section V.

Here, ‘heterogeneity’ indicates the discrimination of feature structure, e.g., the DAE-based feature structure and the GCN-based feature structure.

Fig. 1. The overall flowchart of the proposed method, namely deep attention-guided graph clustering with dual self-supervision (DAGC). It consists of a DAE module, a heterogeneity-wise fusion (HWF) module, a scale-wise fusion (SWF) module, a distribution-wise fusion (DWF) module, a soft self-supervision (SSS) strategy, and a hard self-supervision (HSS) strategy. Specifically, HWF and SWF conduct the weight fusion in the sum and concatenation manner, respectively, where both modules involve a multilayer perceptron module, a normalization operation, and a GCN module. DWF uses a softmax function to infer a probability distribution. To achieve an end-to-end self-supervision, SSS drives the soft assignments to achieve distributions alignment between $Q$ and $Z$ distributions, and HSS transfers the cluster assignment to a hard one-hot encoding. The detailed architectures of HWF, SWF, DWF, and SSS are given in Figures 2, 3, and 4, respectively.

### II. Related Work

DAE is a typical deep neural network that allows computational models composed of multiple processing layers to learn data representations with multiple levels of abstraction. Recently, benefiting from the powerful representation ability
of DAE, deep embedding clustering has achieved remarkable development [26], [27], [28], [29], [30], [31], [32]. For example, Hinton and Salakhutdinov [12] used DAE to extract the feature representation of input data, on which K-means [13] is performed to obtain the clustering results. Reference [14] jointly conducted embedding learning and cluster assignment in an iterative optimization manner. The improved DEC (IDEC) [15] enhanced the clustering performance by adding a reconstruction loss function into DEC. A series of works [2], [3], [33] introduced multi-view information into the DAE framework to further improve embedding learning. However, these DAE-based methods neglect the underlying topological structure among data, which has demonstrated its effectiveness for data clustering [16], [17], [18], [34], [35], thus limiting their performance.

Graph embedding is a new paradigm for clustering to capture the topological structure among samples [17], [36], [37], [38], [39], [40], [41], and many recent approaches [42], [43], [44], [45], [46] have explored GCN to achieve graph embedding. For instance, Kipf and Welling [20] provided GAE and VGAE methods by incorporating GCN into DAE and variational DAE frameworks, respectively. Reference [21] extended GAE by introducing a designed adversarial regularization. Wang et al. [22] merged GAE and the graph attention network [23] to build a deep attentional embedding framework for attributed graph clustering (DAEGC). The structural deep clustering network (SDCN) [24] fused the node content and topological structure information to achieve deep embedding clustering. Peng et al. [25] designed the attention-driven graph clustering network (AGCN) to merge numerous features to enhance the embedding learning via an adaptive mechanism. The deep fusion clustering network (DFCN) [47] exploited a designed fusion strategy to combine the DAE and GAE frameworks to merge the node attribute and topological structure information. He et al. [48] developed an adaptive graph convolutional clustering (AGCC) model to update the graph structure and the data representation layer by layer.

Optimizing a deep clustering network is a fundamental yet challenging task because there are no ground-truth labels as supervision. Previous works [14], [15], [22], [24], [25] minimize the Kullback-Leibler (KL) divergence to tackle this challenge, and its effectiveness has been proven. Specifically, it first uses the Student’s t-distribution [49], [50] as a kernel function to measure the similarity between the extracted feature $h_i$ and its corresponding centroid vector $m_j$, in which the measured similarity can be regarded as a probability distribution $Q$ with its $i,j$-th element being

$$q_{i,j} = \frac{(1 + \|h_i - m_j\|^2/\alpha)^{\frac{-1}{\alpha}}}{\sum_{j'}(1 + \|h_i - m_{j'}\|^2/\alpha)^{\frac{-1}{\alpha}}}$$

where $\alpha$ is set to 1. Then, it implements the KL divergence minimization between $Q$ and a distribution $B$ with the $i,j$-th element $b_{i,j} = \frac{q_{i,j}^2}{\sum_{j'}q_{i,j'}^2}$, i.e.

$$\min KL(B, Q) = \sum_i \sum_j b_{i,j} \log \frac{b_{i,j}}{q_{i,j}}$$

where $KL(\cdot, \cdot)$ is the Kullback-Leibler divergence function that measures the distance between two distributions. Such an auxiliary distribution generation normalizes the high-confidence probability value as a large value, capable of learning the high-confidence assignments. However, previous works fail to sufficiently utilize the available off-the-shelf information from high-confidence clustering assignments, inevitably leading to inferior clustering results.

### III. PROPOSED METHOD

Figure 1 illustrates the architecture of the proposed method, where we will detail the main components in what follows.

#### A. Heterogeneity-Wise Fusion

We first exploit a DAE module with a series of encoders and decoders to extract the latent representation by adapting the reconstruction loss, i.e.,

$$L_R = \|X - \hat{X}\|^2_F$$

where $X$ and $\hat{X}$ denote the input matrix and the reconstructed matrix, respectively. Here, $H_0 = X$, $\hat{H}_l = \hat{X}$, $H_l = \phi(W_{l}^eH_{l-1} + b_{l}^e)$, $\hat{H}_l = \phi(W_{l}^d\hat{H}_{l-1} + b_{l}^d)$, where $H_l$ and $\hat{H}_l$ denote the encoder and decoder outputs from the $i_{th}$ layer, respectively; $l$ denotes the number of encoder/decoder layers, $W_{l}^e$, $b_{l}^e$, $W_{l}^d$, and $b_{l}^d$ denote the network weight and bias of the $i_{th}$ encoder and decoder layer, respectively, and $\phi(\cdot)$ denotes an activation function, such as Tanh or ReLU [51]. Particularly, we set $H = H_l$ for convenience. In addition, we denote the GCN feature learned from the $i_{th}$ layer as $Z_i \in \mathbb{R}^{n \times d_i}$ with $d_i$ being the dimension of the $i_{th}$ layer, where $Z_0 = X$. Previous works (e.g., SDCN [24]) combine the heterogeneity-wise representation on the $i_{th}$ layer ($Z_i$ and $H_i$) via a fixed fusion strategy (i.e., $Z'_i = 0.5Z_i + 0.5H_i$) to enhance representation learning. However, such a fusion strategy is simple but unreasonable since the heterogeneity-wise representations $Z_i$ and $H_i$ owe different characteristic information. To this end, we propose a learning-aware fusion strategy to develop an adaptive fusion strategy to dynamically weight $Z_i$ and $H_i$. Specifically, to learn the corresponding attention coefficients of $Z_i$ and $H_i$, we first concatenate them as $[Z_i][H_i] \in \mathbb{R}^{n \times 2d_i}$ and then build a fully connected layer parameterized by a weight matrix $W_{l}^f \in \mathbb{R}^{2d_i \times 2}$, and the output of the LReLU unit via the softmax function and the $\ell_2$ normalization (i.e., $\text{softmax-}\ell^2$ normalization). Formally, we formulate the prediction of the corresponding attention coefficients as

$$M_i = [m_{i,1}][m_{i,2}] = \Theta_{A}([Z_i][H_i]W_{l}^f)$$

where $M_i \in \mathbb{R}^{n \times 2}$ is the attention coefficient matrix with entries being greater than 0, $m_{i,1}$ and $m_{i,2}$ are the weight vectors for measuring the importance of $Z_i$ and $H_i$, respectively, and $\Theta_{A}(\cdot) = \ell_2(\text{softmax}(LReLU(\cdot)))$. Thus, we can adaptively fuse the GCN feature $Z_i$ and the DAE feature $H_i$ on the $i_{th}$ layer as

$$Z'_i = (m_{i,1}1) \odot Z_i + (m_{i,2}1) \odot H_i$$

where $\odot$ denotes the element-wise product.
More specifically, we first learn the weights through an attention-based mechanism (the left dashed box in the triple-solid line box) and then integrate the
where $w_i$ denotes the weight matrix from the $i$th layer. 

$$Z_{i+1} = \text{ReLU}((D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}Z_i)W_i),$$

where $W_i$ denotes the weight matrix from the $i$th GCN layer, and $D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}$ normalizes $A$ by using renormalization with a self-loop normalized $A$ and the corresponding $D$.

### B. Scale-Wise Fusion

As aforementioned, previous works neglect the off-the-shelf multi-scale information embedded in different layers, which is of great importance for embedding learning. To this end, we propose the SWF module to concatenate the multi-scale features from different layers via an attention-based mechanism. The right border of Figure 2 shows the overall architecture of SWF.

We aggregate the multi-scale features with a concatenation manner to dynamically combine various scale features with different dimensions. Afterwards, we build a fully connected layer parametrized by a weight matrix $W^3 \in \mathbb{R}^{(\sum_{j=1}^{l}d_j + d_{l+1}) \times (l+1)}$ to capture the relationship among the multi-scale features. Formally, we formulate the whole process as

$$U = \mathcal{Y}_A \left( \mathcal{Z}_{j=1}^{l+1} \mathcal{Z}_j W^3 \right),$$

where $\mathcal{Z}_{j=1}^{l+1} \mathcal{Z}_j = [||Z_k|| \cdots ||Z_k||_{Z_{l+1}}]$ denotes the concatenation operation of multiple elements. We then conduct the feature fusion as

$$Z' = \mathcal{Z}_{j=1}^{l+1} ((u_j, I_j) \circ Z_j),$$

where $u_j$ is the $j$-th element of $U$, i.e., $\mathcal{Z}_{j=1}^{l+1} u_j = U$. In addition, we use a Laplacian smoothing operator [53] and the softmax function to make the fused feature $Z'$ as a reasonable probability distribution, i.e.,

$$Z = \text{softmax}(D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}Z'W),$$

where $W$ denotes the learnable parameters.

### C. Distribution-Wise Fusion

As we obtain the feature $H$ from DAE, we can exploit it to calculate its cluster center embedding $\mu$ with K-means. Afterward, we measure the similarity between the extracted feature $h_i$ and its corresponding centroid vector $\mu_j$, in which the measured similarity can be regarded as a probability distribution $Q$ with its $i, j$-th element being $q_{i,j} = \frac{(1 + ||h_i - \mu_j||^2)^{-1}}{\sum_{j}(1 + ||h_i - \mu_j||^2)^{-1}}$, following Eq. (1). Both $Z$ and $Q$ can generate the final clustering results; however, it is challenging to choose which one to obtain the final clustering result within different scenarios. To the best of our knowledge, this is an unsolved decision-making dilemma commonly existing in the previous deep graph clustering methods. To handle this challenge, we propose the DWF module to fuse the learned data probability distributions in an attention-driven manner to predict cluster labels.

Specifically, we first learn the importance of $Z$ and $Q$ by an attention-based mechanism, i.e.,

$$V = [v_1 \parallel v_2] = \mathcal{Y}_A \left( [Z \parallel Q] \hat{W} \right),$$
where $V \in \mathbb{R}^{n \times 2}$ is the attention coefficient matrix, $\hat{W}$ is a learned weight matrix via a fully connected layer. We then adaptively leverage $Z$ and $Q$ as

$$F = (v_1 1) \odot Z + (v_2 1) \odot Q,$$

where $1 \in \mathbb{R}^{1 \times k}$ denotes the vector of all ones. Finally, we apply the softmax function to normalize $F$ with

$$F = \text{softmax}(F) \quad \text{s.t.} \quad \sum_{j=1}^{k} f_{i,j} = 1, \quad f_{i,j} > 0,$$

where $f_{i,j}$ is the element of $F$. When the network is well-trained, we can directly infer the predicted cluster label through $F$, i.e.,

$$y_i = \arg \max_j f_{i,j} \quad \text{s.t.} \quad j = 1, \ldots, k,$$

where $y_i$ is the predicted label of $x_i$. In this way, the cluster structure can be represented explicitly in $F$.

D. Dual Self-Supervision

As unsupervised clustering lacks reliable guidance, we propose a novel dual self-supervision scheme that combines a soft self-supervision strategy with a Kullback-Leibler (KL) divergence loss and a hard self-supervision strategy with a pseudo supervision loss to guide the overall network training, as illustrated in Figure 4.

1) Soft Self-Supervision: Since we take advantage of the high-confidence assignments to iteratively refine the clusters by utilizing the soft assignments (i.e., the probability distributions $Q$ and $Z$), we term this supervision strategy as the **soft self-supervision** strategy. Concretely, since $Z$ involves the graph information through the HWF and SWF modules, we first derive an auxiliary distribution $P$ via $Z$ by normalizing per cluster after squaring $z_{i,j}$, i.e.,

$$p_{i,j} = \frac{z_{i,j}^2}{\sum_{k=1}^{n} z_{i,k}^2},$$

where $0 \leq p_{i,j} \leq 1$ is the element of $P$. Then, we minimize the KL divergence not only between a learned distribution and its auxiliary distribution (i.e., $KL(P, Z)$ and $KL(P, Q)$), but also between two learned distributions (i.e., $KL(Z, Q)$) to promote a highly consistent distribution alignment to train our model, i.e.,

$$\mathcal{L}_S = \lambda_1 * (KL(P, Z) + KL(P, Q)) + \lambda_2 * KL(Z, Q)$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are the trade-off parameters.

2) Hard Self-Supervision: Although the soft self-supervision strategy has become a helpful tool for unsupervised clustering, it preserves the low-confidence predicted probabilities, limiting the clustering performance. To further make use of the available off-the-shelf information from the cluster assignments, we introduce the pseudo supervision technique [54] and set the pseudo-label $\hat{y}_i$ as $\hat{y}_i = y_i$. Considering that the pseudo-labels may contain many incorrect labels, we select the high-confidence ones as supervisory information by a large threshold $r$, i.e.,

$$g_{i,j} = \begin{cases} 1 & \text{if } f_{i,j} \geq r, \\ 0 & \text{otherwise}. \end{cases}$$

In the experiment, we set $r = 0.8$. Then, we leverage the high-confidence pseudo-labels to supervise the network training, i.e.,

$$\mathcal{L}_H = \lambda_3 \sum_j g_{i,j} \star \gamma_{CE}(f_{i,j}, \gamma_{OH}(\hat{y}_i)),$$

where $\lambda_3 > 0$ is the trade-off parameter, $\gamma_{CE}$ denotes the cross-entropy [55] loss, and $\gamma_{OH}$ transforms $\hat{y}_i$ to its one-hot form. As shown in Figure 4, the pseudo-labels transfer the cluster assignment to the hard one-hot encoding, we thus name it as **hard self-supervision** strategy.

In addition, we have empirically observed that only using HSS does not perform well in all scenarios. The reason may be that the distribution probability values are small in some situations, making a weak self-supervision for guiding network training with the HSS strategy. To this end, we combine the SSS and HSS strategies together to drive the network training. Combining Eqs. (3), (15), and (17), our overall loss function can be written as

$$\mathcal{L} = \min_F (\mathcal{L}_R + \mathcal{L}_S + \mathcal{L}_H).$$

The whole training process is shown in Algorithm 1.

E. Computational Complexity Analysis

For the DAE module, the time complexity is $O(n \sum_{i=2}^{l} d_{i-1} d_i)$. For the GCN module, as the operation can be computed efficiently using sparse matrix computation, the time complexity is only $O(\ell \sum_{i=2}^{l} d_{i-1} d_i)$ according to [21]. For Eq. (1), the time complexity is $O(nk + n \log n)$ based on [14]. For HFW, SWF, and DWF modules, the total time complexity is $O((\sum_{i=2}^{l} d_i)(l+1) + O(k))$. Thus, the overall computational complexity of Algorithm 1 in
As those non-graph datasets (i.e., USPS, Reuters, and HHAR) lack the topology graph, we used a typical graph construction approach to generate their graph data. Specifically, we first employed the cosine similarity to construct an undirected KNN graph. The constructed KNN graph can depict the topological structure information based on the DAE and GAE.

IV. EXPERIMENTS

We conducted quantitative and qualitative experiments on nine commonly used benchmark datasets to evaluate the proposed model. In addition, we performed ablation studies to investigate the effectiveness of the proposed modules and the adopted strategies. Moreover, we performed a series of parameter analyses to verify the robustness of our method.

A. Datasets and Compared Methods

We conducted experiments on one image dataset (USPS [56]), one text dataset (Reuters [57]), one record dataset (HHAR [58]), and six graph datasets (ACM, Citeseer, DBLP, Amazon Photo, PubMed, and AIDS [60]), which are briefly summarized in Table II.

We compared the proposed method with the classic clustering method K-means [13], three DAE-based embedding clustering methods [12], [14], [15], and seven GCN-based clustering methods [20], [21], [22], [23], [24], [25], [47], the details of which are listed as follows.

- **DAE** [12] uses deep auto-encoder to learn latent feature representations and then performs K-means on that feature to obtain clustering results.
- **DEC** [14] jointly conducts embedding learning and clustering assignment with an iterative procedure.
- **SDCN** [24] fuses DAE and GCN to merge the topological structure information into deep embedding clustering.
- **ARGA** [21] guides embedding learning with a designed adversarial regularization.
- **DAEGC** [22] achieves a neighbor-wise embedding learning with an attention-driven strategy and supervises the network training with a clustering loss.
- **AGCC** [48] replaces the graph layer by layer to mine the latent connected relationship between data.

B. Implementation Details

1) Evaluation Metrics: We used four metrics to evaluate the clustering performance, including Average Rand Index (ARI), macro F1-score (F1), Accuracy (ACC), and Normalized Mutual Information (NMI). For each metric, a larger value implies a better clustering result.

2) Graph Construction: As those non-graph datasets (i.e., USPS, Reuters, and HHAR) lack the topology graph, we used a typical graph construction approach to generate their graph data. Specifically, we first employed the cosine distance to compute the similarity matrix $S$, i.e.,

$$S = \frac{XX^T}{\|X\|_F \|X^T\|_F},$$

(19)

where $\|X\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_{i,j}|^2}$ and $X^T$ denote the Frobenius norm and the transpose operation of $X$, respectively. Then, we keep the top-$k$ similar neighbors of each sample to construct an undirected $k$-nearest neighbor (KNN) graph. The constructed KNN graph can depict the topological structure of a dataset and hence is used as GCN input.

3) Training Procedure: Similar to [14], [15], [24], and [25], we first pre-trained the DAE module with 30 epochs and the learning rate equal to 0.001. Then, we trained the whole network with 200 iterations. We set the dimension of the auto-encoder and the GCN layers to $500 - 500 - 2000 - 10$, the batch size to 256, and the negative input slope of LReLU

\[\text{Algorithm 1 Training Process of Our Method}\]

**Input:** Input matrix $\mathbf{X}$; Adjacency matrix $\mathbf{A}$; Cluster number $k$; Trade-off parameters $\lambda_1, \lambda_2, \lambda_3$; Maximum iterations $i_{\text{MaxIter}}$.

**Output:** Reconstructed matrix $\hat{\mathbf{X}}$; Clustering result $\mathbf{y}$.

1. Initialization: $l = 4, i_{\text{Iter}} = 1$; $\mathbf{Z}_0 = \mathbf{X}; \mathbf{H}_0 = \mathbf{X}$.
2. Initialize the parameters of the DAE network.
3. while $i_{\text{Iter}} < i_{\text{MaxIter}}$ do
4. Obtain the feature $\mathbf{H}$ by Eq. (3).
5. Obtain the feature $\mathbf{Z}$ via Eq. (9).
6. Obtain the cluster center embedding $\mu$ with K-means based on the feature $\mathbf{H}$.
7. Calculate the distribution $\mathbf{Q}$ via Eq. (1).
8. Calculate the distribution $\mathbf{P}$ via Eq. (14).
9. Calculate the distribution $\mathbf{F}$ via Eq. (12).
10. Conduct the soft self-supervision via Eq. (15).
11. Conduct the hard self-supervision via Eq. (17).
12. Minimize the overall loss function via Eq. (18).
13. Conduct the back propagation and update parameters in the proposed network.
14. $i_{\text{Iter}} = i_{\text{Iter}} + 1$.
15. end while
16. Calculate the clustering results $\mathbf{y}$ with $\mathbf{F}$ by Eq. (13).

one iteration is about $O(n \sum_{i=2}^{n-1} d_i + |E| \sum_{i=2}^{n-1} d_i + (n + 1)n + \log n + \sum_{i=1}^{n-1} (d_i + (\sum_{i=1}^{n-1} d_i)(l + 1))$.

### Table II

| Dataset      | Type     | Samples | Classes | Edges |
|--------------|----------|---------|---------|-------|
| USPS         | Image    | 9298    | 10      | 27894 |
| Reuters      | Text     | 10000   | 4       | 30000 |
| HHAR         | Record   | 10299   | 6       | 30897 |
| ACM          | Graph    | 3025    | 3       | 13128 |
| CiteSeer     | Graph    | 3327    | 6       | 4552  |
| DBLP         | Graph    | 4057    | 4       | 3528  |
| Amazon Photo | Graph    | 7650    | 8       | 119081|
| PubMed       | Graph    | 19717   | 3       | 44324 |
| AIDS         | Graph    | 31385   | 38      | 64780 |

- **IDEC** [15] introduces a reconstruction loss into DEC to improve the clustering performance.
- **GAE** [20] and **VGA** [20] incorporate DAE and variational DAE into GCN frameworks, respectively.
- **DAEGC** [22] achieves a neighbor-wise embedding learning with an attention-driven strategy and supervises the network training with a clustering loss.
- **ARGA** [21] guides embedding learning with a designed adversarial regularization.
- **SDCN** [24] fuses DAE and GCN to merge the topological structure information into deep embedding clustering.
- **AGCN** [25] focuses on enhancing the embedding learning.
- **DFCN** [47] merges the node attribute and topological structure information based on the DAE and GAE.

2\text{http://dl.acm.org}
3\text{http://CiteSeerx.ist.psu.edu/}
4\text{https://dblp.uni-trier.de}
TABLE III
CLUSTERING RESULTS (MEAN±STD) WITH TWELVE COMPARED METHODS ON NINE BENCHMARK DATASETS. THE BEST AND SECOND-BEST RESULTS ARE BOLDED AND UNDERLINED, RESPECTIVELY. ‘OOM’ DENOTES THE OUT OF MEMORY CASE

| Datasets       | Method               | Reuters | DARE [24] | ECM [31] | GAE [49] | NIPS [15] | SDCN [24] | TUM [30] | ACM [33] | DBLP [25] | PubMed [37] | HHAR | AMEX | DBLP [25] | ACM [33] | CiteSeer | Amazon Photo | ORS       |
|----------------|----------------------|---------|-----------|-----------|-----------|-----------|-----------|---------|----------|-----------|------------|------|-------|-----------|----------|---------|-------------|----------|
|                |                      |         | (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)| (Mean±Std)|
|                |                      |         | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 | 40.70±0.80 |
reducing the graph quality. Although AGCN also uses the KNN graph, its auxiliary distribution $P$ was inferred by the output of the conventional auto-encoder. Differently, the proposed method uses the graph convolutional network output to derive $P$ for utilizing rich graph information of $Z$. Thus, if the graph quality is terrible, the clustering performance of the proposed method may be worse than AGCN.

- Our method obtains the best clustering performance on AIDS with four metrics, where AIDS is a large-scale long-tailed dataset that one class accounts for 62.34% number, and the other thirty-seven classes share 37.66%.

D. Ablation Study

We conducted comprehensive ablation studies to validate the effectiveness of the proposed modules and self-supervision strategies. Table IV lists the quantitative results, where the first row of each dataset denotes the baseline that merges the DAE and GCN features in a half-and-half mechanism (i.e., without HWF) and uses the last GCN layer feature (i.e., without SWF and DWF) to conduct the optimization via the reconstruction loss and self-optimizing embedding loss following [14], [15], [22], [24], [25] (i.e., without SSS and HSS). The second, third, and fourth rows denote the baseline that adopts the proposed module HWF, HWF+SWF, and HWF+SWF+DWF, respectively. The fifth, sixth, and seventh rows denote the methods optimized by the introduced hard self-supervision, the soft self-supervision, and both.

1) Heterogeneity-Wise Fusion Module: By comparing the first and second rows of each dataset in Table IV, we can observe that HWF can typically improve clustering performance in most cases, validating its effectiveness. For example, on Reuters, it produces a 5.01% performance improvement on ARI, 2.19% on F1, 2.00% on ACC, and 4.20% on NMI.

2) Scale-Wise Fusion Module: We can examine the effectiveness of SWF by comparing the second and third rows of each dataset in Table IV, in which the compared results with four metrics indicate the superiority of the SWF module in most datasets.

3) Distribution-Wise Fusion Module: By comparing the results of the third and fourth rows of each dataset in Table IV, we observe that DWF also improves clustering performance, benefiting from the adaptive fusion of the information of two distributions.

To qualitatively validate the significant performance of the DWF module, we plotted 2D t-distributed stochastic neighbor embedding (t-SNE) [62] visualizations of the distributions $Q$, $Z$, and $F$ on DBLP in Figure 5, where we can see that our adaptively aggregated one is better than others, benefiting from adaptively (due to the DWF module) and effectively (due to the dual self-supervision solution) merging the information of two distributions.

4) Hard Self-Supervision Strategy: From the results of the fourth and fifth rows of each dataset in Table IV, it can be seen that on the non-graph dataset HHAR and graph dataset DBLP, there is about 2.00% improvement when involving HSS, validating its effectiveness.

5) Soft Self-Supervision Strategy: We can validate the effectiveness of SSS by comparing the results of the fourth and sixth rows of each dataset in Table IV. Specifically, on DBLP, SSS produces 13.73% improvement on ARI, 7.15% on F1, 7.37% on ACC, and 10.82% on NMI. Such impressive improvement is credited to that the SSS strategy refines the cluster assignment by minimizing a Kullback-Leibler divergence loss to promote consistent distribution alignment among distributions $Q$, $Z$, and $P$.

6) Dual Self-Supervision (DSS): By comparing the results of the fourth, fifth, sixth, and seventh rows of each dataset in Table IV, we can observe that DSS, which combines HSS and SSS, almost produces the best results on all nine benchmark datasets.

E. Parameters Analysis

1) Analysis of the Number of Neighbors: As the number of neighbors $k$ directly decides the KNN graph with respect to (w.r.t.) the quality of the adjacency matrix, we tested different $k$ on the non-graph datasets, i.e., USPS, Reuters, and HHAR. From Figure 6, we can observe that our model is not sensitive to $k$. In the experiments, we fixed $k$ to 3 to construct the KNN graph for the non-graph datasets.

2) Analysis of Hyperparameters: We investigated the influence of the hyperparameters, i.e., $\lambda_1$, $\lambda_2$, and $\lambda_3$, on DBLP. Figure 7 illustrates four metrics results in a 4D figure manner where the color indicates the fourth direction, i.e., the corresponding experimental results. From Figure 7, we have the following observations.

- The parameters setting of $\lambda_1$ and $\lambda_2$ is critical to the proposed model. Specifically, the highest clustering result occurs when $\lambda_1$ and $\lambda_2$ tend to the same value. This phenomenon reflects the importance of balancing the regularization term in constraining the distribution alignment.

- Our model is robust to the hyperparameter $\lambda_3$, i.e., our method can obtain the optimal performance in a wide and common parameter range of $\lambda_3$.
3) Analysis of the Threshold Value: We investigated the effect of the threshold value $r$ on clustering performance. Figure 8 shows the clustering results with various thresholds (i.e., 0.5, 0.6, 0.7, 0.8, and 0.9). From Figure 8, we have the following conclusions.

- A small threshold value unavoidably degrades the clustering performance compared with the ones using a large threshold value. For example, when we set $r$ to 0.5 or 0.6, all four metrics results on DBLP have degraded performance. Apparently, a small
threshold value can easily generate a lot of incorrect pseudo-labels.

- A large threshold value is capable of leading to high clustering performance. However, setting $r$ to a tremendous value like 0.9 cannot improve clustering performance. The reason is that with a larger threshold, the number of selected supervised labels will reduce, resulting in weak label propagation. Thus, we set $r$ to 0.8 in this paper.

4) Analysis of the Learned Attention-Aware Weights: We added the results of the learned weight on DBLP to verify the effectiveness of the designed attention mechanism in Table V, where $x_j$ indicates the $j$-th sample; $m^{(1)}_i$, $m^{(2)}_i$, and $m^{(3)}_i$ indicate the HWF learned weights of $Z_i$ and $H_i$ in the $i$-th layer, respectively; $u_1$, $u_2$, $u_3$, $u_4$, and $u_5$ indicate the SWF learned weights; $v_1$ and $v_2$ indicate the DWF learned weights of $Z$ and $Q$, respectively; AVG indicates the average value of the weight results. The clustering results of $Z$ and $Q$ are inferred through their column indexes of the maximum in each row, and those results of other features are obtained with K-means, where the higher clustering performance, the better the feature representation. We can see that the representation corresponding to a large weight value typically performs better clustering results than the one corresponding to a small weight value, substantiating the effectiveness of the designed attention mechanism in the weighted fusion.

F. Time and Space Complexity Analysis

We repeated the experiment 10 times to compare the mean values, the standard deviations (i.e., mean±std), the parameters number, and the running time of the proposed method with the baselines [24], [25], [47] on DBLP in Table VI. Specifically, the experiments are implemented with Python 3.6.12 and Pytorch-1.9.0+cu102 on an NVIDIA GeForce RTX 2080 Ti and an i7-8700K CPU. M and s are the abbreviations of the million and second, respectively. From Table VI, we can observe that our method obtains a significant clustering improvement at the cost of acceptable resource consumption.

G. Visual Comparison

To qualitatively evaluate the effectiveness of the proposed method, we plotted 2D t-SNE visualizations of baselines [24], [25], [47], and Ours, where different colors represent different clusters.

Fig. 7. Analysis of different hyperparameters ($\lambda_1$, $\lambda_2$, and $\lambda_3$) with four metrics on DBLP. We illustrate the results w.r.t. these hyperparameters in a 4D figure manner, where the color indicates the fourth direction, i.e., the corresponding experimental results.

Fig. 8. Investigation of the effect of the threshold value for pseudo supervision on (a) USPS, (b) Reuters, (c) HHR, (d) ACM, (e) CiteSeer, (f) DBLP, (g) Amazon Photo, (h) PubMed, and (i) AIDS.

Fig. 9. Comparison of the visualization of the learned representations on DBLP by (a) SDCN [24], (b) AGCN [25], (c) DFCN [47], and (d) Ours, where different colors represent different clusters.

of the weight results. The clustering results of $Z$ and $Q$ are inferred through their column indexes of the maximum in each row, and those results of other features are obtained with K-means, where the higher clustering performance, the better the feature representation. We can see that the representation corresponding to a large weight value typically performs better clustering results than the one corresponding to a small weight value, substantiating the effectiveness of the designed attention mechanism in the weighted fusion.
[25], [47] and the proposed method on DBLP in Figure 9, where we can find that the feature representation obtained by our method shows the best separability for different clusters, where we can find that the feature representation obtained by [25], [47] and the proposed method on DBLP in Figure 9, where we can find that the feature representation obtained by our method shows the best separability for different clusters, the gap between different groups is the most obvious one. This phenomenon substantiates that our method produces the most clustering-oriented representation compared with state-of-the-art methods.

### V. CONCLUSION

We have presented a novel deep embedding clustering method that simultaneously enhances embedding learning and cluster assignment. Specifically, we first designed heterogeneity-wise and scale-wise fusion modules to learn an informative representation adaptively. Then, we utilized a distribution-wise fusion module to achieve cluster enhancement via an attention-based mechanism. Finally, we proposed a soft self-supervision strategy with a Kullback-Leibler divergence loss and a hard self-supervision strategy with a pseudo supervision loss to utilize the available off-the-shelf information from the cluster assignments. The quantitative and qualitative experiments and analyses demonstrate that our method consistently outperforms state-of-the-art approaches. We also provided comprehensive ablation studies to validate the effectiveness and advantage of our network.

### REFERENCES

[1] R. Vidal, Y. Ma, and S. Sastry, “Generalized principal component analysis (GPCA),” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.

[2] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao, “Deep multi-view subspace clustering with unified and discriminative learning,” *IEEE Trans. Multimedia*, vol. 23, pp. 3483–3493, 2020.

[3] Z. Deng, C. Deng, X. Yang, and H. Huang, “Multi-scale fusion subspace clustering using similarity constraint,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6658–6667.

[4] X. Wang, S. Fan, K. Kuang, C. Shi, J. Liu, and B. Wang, “Decorrelated clustering with data selection bias,” in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2177–2183.

[5] Y. Jia, H. Liu, J. Hou, and Q. Zhang, “Clustering ensemble meets low-rank tensor approximation,” in *Proc. AAAI*, vol. 35, no. 9, 2021, pp. 7970–7978.

[6] S. Yang, W. Deng, M. Wang, J. Du, and J. Hu, “Orthogonality loss: Learning discriminative representations for face recognition,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2301–2314, Jun. 2021.

[7] Y. Jia, J. Hou, and S. Kwong, “Constrained clustering with dissimilarity propagation-guided graph-Laplacian PCA,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 3985–3997, Sep. 2021.

[8] Y. Wu, L. Du, and H. Hu, “Parallel multi-path age distinguish network for cross-age face recognition,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3482–3492, Sep. 2021.

[9] Z. Peng, W. Zhang, N. Han, X. Fang, P. Kang, and L. Teng, “Active transfer learning,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 1022–1036, Apr. 2020.

[10] Y. Jia, H. Liu, J. Hou, S. Kwong, and Q. Zhang, “Multi-view spectral clustering tailored for low-rank representation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4784–4797, Dec. 2021.

[11] J. Peng, Y. Jia, H. Liu, and Q. Zhang, “Maximum entropy subspace clustering network,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2199–2210, Apr. 2022.

[12] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[13] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. 5th Berkeley Symp. Math. Statist. Probabil.*, vol. 1. Berkeley, CA, USA, Jan. 1967, pp. 281–297.

[14] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in *Proc. ICML*, New York, NY, USA, 2016, pp. 478–487.

[15] X. Gao, L. Gao, X. Liu, and J. Yin, “Improved deep embedded clustering with local structure preservation,” in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.

[16] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, “Simplifying graph convolutional networks,” in *Proc. ICML*, 2019, pp. 6861–6871.

[17] T. Kim and A. Oh, “How to find your friendly neighborhood: Graph attention design with self-supervision,” in *Proc. ICLR*, Vienna, Austria: ICLR, 2021, pp. 1–14.

[18] L. Wu, P. Cui, J. Pei, and L. Zhao, *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer, 2022.

[19] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. ICLR*, 2017, pp. 1–10.

[20] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” in *Proc. NIPS Workshop*, Barcelona, Barcelona: NIPS, 2016, pp. 1–3.

[21] S. Pan, R. Hu, S.-F. Fang, G. Long, J. Jiang, and C. Zhang, “Learning graph embedding with adversarial training methods,” *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2475–2487, Jun. 2020.

[22] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, “Attributed graph clustering: A deep convolutional embedding approach,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3670–3676.

[23] P. Velicković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *Proc. ICLR*, Vancouver, BC, Canada: ICLR, 2018, pp. 1–12.

[24] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, “Structural deep clustering network,” in *Proc. Web Conf.*, Apr. 2020, pp. 1400–1410.

[25] Z. Peng, H. Liu, Y. Jia, and J. Hou, “Attention-driven graph clustering network,” in *Proc. ACM MM*, 2021, pp. 935–943.

[26] X. Dong, L. Liu, L. Zhu, Z. Cheng, and H. Zhang, “Unsupervised deep K-means hashing for efficient image retrieval and clustering,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3266–3277, Aug. 2021.

[27] J. Huang, S. Gong, and X. Zou, “Deep semantic clustering by partition confidence maximisation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8849–8858.

[28] Z. Peng, Y. Zou, and Z. Zhang, “Cluster attention contrast for video anomaly detection,” in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2463–2471.

[29] K. Han, A. Vedaldi, and A. Zisserman, “Learning to discover novel visual categories via deep transfer clustering,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8401–8409.

[30] Y. Ou, Z. Chen, and F. Wu, “Multimodal local-global attention network for affective video content analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 5, pp. 1901–1914, May 2021.

[31] X. Wang et al., “Dynamic attention guided multi-trajectory analysis for single object tracking,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4895–4908, Dec. 2021.

[32] Y. Liu et al., “Deep graph clustering via dual correlation reduction,” in *Proc. AAAI*, 2022, pp. 7662–7670.

[33] Y. Hu, Z. Song, B. Wang, T. Jiang, Gao, Y. Sun, and B. Yin, “AKM3C: Adaptive K-multiple-means for multi-view clustering,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 11, pp. 4214–4226, Nov. 2021.

[34] Z. Zhang, J. Wang, J. Ye, and F. Wu, “Rethinking graph convolutional networks in knowledge graph completion,” in *Proc. ACM Web Conf.*, Apr. 2022, pp. 798–807.

[35] H. He, J. Wang, Z. Zhang, and F. Wu, “Compressing deep graph neural networks via adversarial knowledge distillation,” in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 534–544.
[36] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, “AM-GCN: Adaptive multi-channel graph convolutional networks,” in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2020, pp. 1243–1253.

[37] Z. Zhang, P. Cui, and W. Zhu, “Deep learning on graphs: A survey,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 1, pp. 249–270, Jan. 2022.

[38] B. Tu, J. Zhang, Y. Li, G. Lu, and W. Zhang, “Multi-label chest X-ray image classification via semantic similarity graph embedding,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 4, pp. 2455–2468, Apr. 2022.

[39] J. He, T. Zhang, Y. Zheng, M. Xu, Y. Zhang, and F. Wu, “Consistency graph modeling for semantic correspondence,” IEEE Trans. Image Process., vol. 30, pp. 4932–4946, 2021.

[40] J. Wang, Z. Zhang, Z. Shi, J. Cai, S. Ji, and F. Wu, “Duality-induced regularizer for random matching knowledge graph embeddings,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Mar. 24, 2022, doi: 10.1109/TPAMI.2022.3161804.

[41] L. Hu, Z. Dai, L. Tian, and W. Zhang, “Class-oriented self-learning graph embedding for image compact representation,” IEEE Trans. Circuits Syst. Video Technol., early access, Aug. 10, 2022, doi: 10.1109/TCVS.2022.3197746.

[42] A. Markovits, G. Shari, I. Friedman, L. Zelnik-Manor, and S. Avidan, “Graph-embedded pose clustering for anomaly detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10539–10547.

[43] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi, “Symmetric graph convolutional autoencoder for unsupervised graph representation learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6519–6528.

[44] P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey,” Knowl.-Based Syst., vol. 151, pp. 78–94, Jul. 2018.

[45] G. Li, M. Müller, A. Thabet, and B. Ghanem, “DeepGCNs: Can GCNs go as deep as CNNs?” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9267–9276.

[46] Q. Huang, H. He, A. Singh, S.-N. Lim, and A. Benson, “Combining label propagation and simple models out-performs graph neural networks,” in Proc. ICLR Vienna, Austria, 2021, pp. 1–19.

[47] W. Tu et al., “Deep fusion clustering framework,” in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 11, 2021, pp. 9978–9987.

[48] X. He, B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, “Parallelly adaptive graph convolutional clustering model,” IEEE Trans. Neural Netw. Learning Syst., early access, May 26, 2022, doi: 10.1109/TNNLS.2022.3176411.

[49] F. Helmbert, “Die genaue Parallelität der formel von peters zur berechnung des wahrscheinlichen beobachtungsfehlers director beobachtungen gleicher genauerheit,” Astronomische Nachrichten, vol. 88, p. 113, Jun. 1876.

[50] W. S. Gosset, “The probable error of a mean,” Biometrika, vol. 6, no. 1, pp. 1–25, Mar. 1908.

[51] X. Ghou, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in Proc. AISTATS. Fort Lauderdale, FL, USA, 2011, pp. 315–323.

[52] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in Proc. ICML. Atlanta, GA, USA: Citeseer, 2013, p. 3.

[53] Q. Li, Z. Han, and X.-M. Wu, “Deeper insights into graph convolutional networks for semi-supervised learning,” in Proc. AAAI, vol. 32, Hilton, LA, USA: AAAI Press, 2018, pp. 1–8.

[54] D.-H. Lee et al., “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in Proc. Workshop Challenges Represent. Learn. (ICML), vol. 3, no. 2, 2013, p. 896.

[55] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A tutorial on the cross-entropy method,” Ann. Oper. Res., vol. 134, no. 1, pp. 19–67, 2005.

[56] J. J. Hull, “A database for handwritten text recognition research,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 16, no. 5, pp. 550–554, May 1994.

[57] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, “RCV1: A new benchmark collection for text categorization research,” J. Mach. Learn. Res., vol. 5, pp. 361–397, Dec. 2004.

[58] L. Su et al., “Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition,” in Proc. SenSys. New York, NY, USA: ACM, 2015, pp. 127–140.

[59] S. Wan, Y. Zhan, L. Liu, B. Yu, S. Pan, and C. Gong, “Contrastive graph Poisson networks: Semi-supervised learning with extremely limited labels,” in Proc. NIPS, vol. 34, 2021, pp. 6316–6327.

[60] K. Binkowski, H. Binkowski, and H. Binkowski, “IAM graph datasets Study Abroad from the China Scholarship Council in 2015 and the Early Career Award (3/381) from the Hong Kong Research Grants Council in 2018. He was an Area Chair of ACM MM 2019–2022, IEEE ICME 2020, VCIP 2020–2022, ICIP 2022, MMSP 2022, and WACV 2021. He is also an Associate Editor of IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, Signal Processing: Image Communication, and the Visual Computer. He served as the Guest Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING and Journal of Visual Communication and Image Representation.

Zhihao Peng received the B.S. and M.S. degrees in computer science and technology from the Guangdong University of Technology, Guangzhou, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong, SAR, China. His current research interests include spectral clustering, subspace learning, and domain adaptation in image/text/graph processing with unsupervised learning.

Hui Liu received the B.Sc. degree in communication engineering from Central South University, Changsha, China, the M.Eng. degree in computer science from Nanyang Technological University, Singapore, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, Hong Kong. From 2014 to 2017, she was a Research Associate at the Maritime Institute, Nanyang Technological University. She is currently an Assistant Professor with the School of Computing Information Sciences, Curtius Institute of Higher Education, Hong Kong. Her research interests include image processing and machine learning.

Yuheng Jia (Member, IEEE) received the B.S. degree in automation and the M.S. degree in control theory and engineering from Zhengzhou University, Zhengzhou, China, in 2012 and 2015, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, SAR, China, in 2019. He is currently an Associate Professor with the School of Computer Science and Engineering, Southeast University, China. His research interests include machine learning, Bayesian method, spectral clustering, and low-rank modeling.

Junhui Hou (Senior Member, IEEE) received the B.Eng. degree in information engineering (Talented Students Program) from the South China University of Technology, Guangzhou, China, in 2009, the M.Eng. degree in signal and information processing from Northwestern Polytechnical University, Xi’an, China, in 2012, and the Ph.D. degree in electrical and electronic engineering from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2016. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests include the general areas of multimedia signal processing, such as image/video/3D geometry data representation, processing and analysis, graph-based clustering/classification, and data compression. He is an Elected Member of IEEE MSA-TC, IEEE VSPC-TC, and IEEE MMSP-TC. He received the Chinese Government Award for Outstanding Students Study Abroad from the China Scholarship Council in 2015 and the Early Career Award (3/381) from the Hong Kong Research Grants Council in 2018. He was an Area Chair of ACM MM 2019–2022, IEEE ICME 2020, VCIP 2020–2022, ICIP 2022, MMSP 2022, and WACV 2021. He is also an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Signal Processing: Image Communication, and the Visual Computer. He served as the Guest Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING and Journal of Visual Communication and Image Representation.