Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Deep Clustering via Gradual Community Detection

Tianyu Cheng
Northwestern Polytechnical University
Na Chen
Northwestern Polytechnical University
Siyi Yang
Northwestern Polytechnical University
Chuyi Fan
Northwestern Polytechnical University
Qun Chen
chenbenben@nwpu.edu.cn

Northwestern Polytechnical University

Research Article

Keywords: Deep Clustering, Community Detection, Complex Network Analysis, Self-Supervised Learning

Posted Date: March 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9074038/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

Deep Clustering via Gradual Community Detection

Tianyu Cheng[1], Na Chen[2], Siyi Yang[2], Chuyi Fan[2], Qun Chen[1,2*]

[1*]School of Computer Science, Northwestern Polytechnical University, 1 Dongxiang Road, Chang'an District, Xi'an, 710129, Shaanxi, China.
[2*]School of Software, Northwestern Polytechnical University, 1 Dongxiang Road, Chang'an District, Xi'an, 710129, Shaanxi, China.

*Corresponding author(s). E-mail(s): chenbenben@nwpu.edu.cn;
Contributing authors: cty18029@163.com; chenna05@mail.nwpu.edu.cn; yangsiyi@mail.nwpu.edu.cn; fanchuyi@mail.nwpu.edu.cn;

Abstract

Deep clustering is an essential task in modern artificial intelligence, aiming to partition a set of data samples into a given number of homogeneous groups (i.e., clusters). Recent studies have proposed increasingly advanced deep neural networks and training strategies for deep clustering, effectively improving performance. However, deep clustering generally remains challenging due to the inadequacy of supervision signals. Building upon the existing representation learning backbones, this paper proposes a novel clustering strategy of gradual community detection. It initializes clustering by partitioning samples into many pseudo-communities and then gradually expands clusters by community merging. Compared with the existing clustering strategies, community detection factors in the new perspective of cluster network analysis in the clustering process. The new perspective can effectively leverage global structural characteristics to enhance cluster pseudo-label purity, which is critical to the performance of self-supervision. We have implemented the proposed approach based on the popular backbones and evaluated its efficacy on benchmark image datasets. Our extensive experiments have shown that the proposed clustering strategy can effectively improve the SOTA performance. Our ablation study also demonstrates that the new network perspective can effectively improve community pseudo-label purity, resulting in improved self-supervision.

Keywords: Deep Clustering, Community Detection, Complex Network Analysis, Self-Supervised Learning

1 Introduction

Due to the success of deep learning, deep clustering has attracted extensive attention from the research community in recent years. The existing approaches for deep clustering can be broadly classified into two categories: two-stage solutions [1, 2] and single-stage ones [3–7]. The two-stage solutions typically alternate between a feature representation learning stage and a cluster assignment stage, which are usually designed to be separate but complementary. In contrast, the single-stage solutions choose to jointly learn feature representation and cluster assignment within an end-to-end framework. It has been well recognized that both two-stage and single-stage solutions have their own advantages and disadvantages. By separating the stages of representation learning and cluster assignment, the two-stage solutions can easily leverage a wide array of existing techniques on either of them. They also usually require less training cost and can be more easily implemented. In comparison, the single-stage solutions can more effectively align representation learning with clustering objectives, achieving improved performance in some applications. However, it is noteworthy that in both two-stage and single-stage solutions, the efficacy of self-supervision remains as the core challenge of deep clustering [8].

To address the challenge of self-supervision, most of the existing studies have focused on designing increasingly advanced representation learning backbones and training strategies. The popular idea was to minimize intra-class distance while maximizing inter-class distance through specific contrastive objectives, e.g., SCAN [2] and TCL [9]. More recent work proposed to optimize image representation via graph-based contrastive learning, e.g., SDCN [10], AGCN [11] and CoNR [12]. Different from basic contrastive clustering that only assumes an image and its augmentation should share similar representations, the graph-based approaches lifted the instance-level consistency to the cluster-level consistency with the assumption that samples in one cluster and their augmentations should all be similar. Some researchers also proposed to exploit text-based supervision signals for deep clustering [13–15]. Their general idea was to generate additional semantic information on images via external knowledge bases or pre-trained multimodal large models (e.g., CLIP [16]) and then leverage them to enhance image representation.

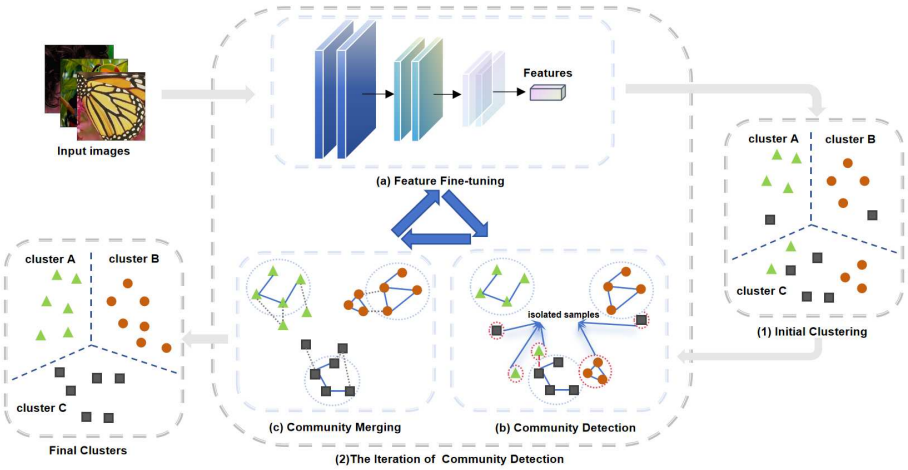


Fig. 1: The workflow of the proposed DCvCD: (1) initial clustering; (2) the iteration of community detection: a) fine-tuning an RL backbone using the labeled main communities; b) community detection on unlabeled samples; c) merging isolated communities with main communities.

Nonetheless, the task of deep clustering generally remains challenging in real scenarios due to the inadequacy of supervision signals. Compared with representation learning, clustering strategy has been much less investigated in the existing literature. In the two-stage solutions, the popular clustering objectives include the traditional K-means loss and maximum mutual information loss [17]. In the single-stage solutions, clustering strategies are instead implicitly encoded as contrastive losses. These strategies primarily focus on pairwise sample relationships and distance-based metrics. Even provided with reasonably good representations, they often group negative samples into a cluster, whose aggregation can adversely affect the performance of self-supervised learning [18].

In this paper, we propose a novel clustering strategy of gradual community detection. Unlike the existing approaches, which directly perform self-supervised learning after initial clustering, our proposed strategy expands clusters by community detection and merging. Community detection brings the new perspective of cluster network analysis into clustering consideration, and can thus improve pseudo-label purity in the self-supervised learning process. We have sketched the proposed approach, denoted by **DCvCD** (**D**eep **C**lustering **v**ia Gradual **C**ommunity **D**etection), in Figure 1. After initial clustering, it divides each cluster into many smaller communities by a community detection algorithm. For this purpose, DCvCD employs the Leiden algorithm [19], which can theoretically guarantee well-connected internal structures within communities. The Leiden algorithm can effectively overcome the "badly-connected communities" problem often found in traditional detection algorithms, thereby ensuring the robustness and high quality of the final partitioning results [19]. Then, DCvCD would select the largest community in a cluster as its main community. Finally, it fine-tunes a Representation Learning (RL) backbone by the pseudo-labels in the main

communities, and updates feature representation. In the next iteration, it performs a new round of community detection on unlabeled samples, and then merges the resulting isolated communities into the main communities. To improve pseudo-label purity, the algorithm of community merging factors in global structural metrics, e.g., network modularity and average degree, besides the traditional distance metrics. The approach alternates between RL backbone fine-tuning and community merging until all the isolated samples are merged.

Built upon the existing RL backbones, DCvCD's main motivation is to improve initial clustering results by community refinement. After initial clustering, the samples within a cluster usually exhibit higher purity compared with before, which makes it easier to generate high-purity main communities. Therefore, in the initialization phase, DCvCD performs community detection on initial clusters instead of an entire dataset. It is noteworthy that DCvCD as a clustering strategy, can in principle work with any RL backbone, including the most recent SOTA ones. Our experiments have shown that by leveraging the recently proposed clustering approaches (e.g., ProPos [20] and CoNR [12]) for initial clustering, DCvCD can effectively improve the SOTA performance, demonstrating the benefit of leveraging global structural characteristics for cluster assignment.

It is worthy to point out that the existing work on graph-based contrastive learning primarily focuses on leveraging pairwise node relationships to improve representation learning. Our proposed DCvCD approach instead introduces the new perspective of complex network analysis into clustering consideration and leverages global cluster structural characteristics for cluster refinement. The major contributions of this work can be summarized as follows:

1. We introduce a novel clustering strategy of gradual community detection for deep clustering. Incorporating the new perspective of cluster network analysis, it can effectively leverage the concepts of complex network to improve self-supervised clustering.
2. We present a novel self-supervised solution of DCvCD for deep clustering, which performs cluster assignment via gradual community expansion. To enhance pseudo-label purity, the algorithm of community merging factors in global structural metrics as well as the traditional distance metrics.
3. We have implemented the DCvCD solution based on the popular clustering approaches and their RL backbones, and validated its efficacy on benchmark image datasets. Our extensive experiments demonstrate that DCvCD can effectively improve the SOTA performance. Our ablation study has also shown that the introduced perspective of cluster network analysis can effectively improve pseudo-label purity.

2 Related Work

Traditional clustering algorithms often perform poorly on complex high-dimensional data, such as images [21–23]. Therefore, deep clustering has established itself as a dominant approach [8]. The typical roadmap of deep clustering is two-stage. It first leverages deep neural networks to transform complex data into low-dimensional feature

representations, and then applies clustering objectives to generate final clusters [1–3, 24, 25].

A lot of effort has been devoted to designing increasingly advanced DNN backbones for single-stage deep clustering. For instance, Yang et al. [26] proposed the Deep Clustering Network (DCN), which clusters latent features produced by an autoencoder using K-means. Xie et al. [27] used a pretrained autoencoder and iteratively refines clusters by KL-divergence clustering loss assignments. Yang et al. [28] proposed JULE, which combines an agglomerative clustering process with deep learning in a recurrent framework. Guo et al. [6] presented the Improved Deep Embedded Clustering (IDEC) method, which optimizes clustering label assignments and feature representations by local structure of data distribution. Dizaji et al. [29] incorporated cross-entropy loss and cluster-size regularization into deep clustering. Huang et al. [30] proposed a deep clustering method called Partition Confidence Maximization (PICA), which seeks to maximize the global partition confidence of clustering solutions. More recently, Huang et al. [20] proposed the ProPos framework, which balances contrastive and non-contrastive learning through prototype scattering and positive sampling alignment. Most recently, the proposal of GeCC [31] aimed to enhance clustering robustness of contrastive learning by explicitly modeling domain shifts. In parallel, Chang et al. [32] introduced an uncertainty-aware contrastive framework that models co-cluster probabilities with a Dirichlet distribution to downweight ambiguous pairs, and Wu et al. [33] recast deep clustering as a semi-supervised problem by fine-tuning a pretrained clustering network with unified positive and negative pseudo-labels. Different from these sample-level reweighting and pseudo-labeling schemes, our DCvCD focuses on refining cluster structures by gradually restructuring communities on similarity graphs.

To improve representation learning, some two-stage solutions have adopted an over-clustering strategy. For instance, the Deep Adaptive Clustering method proposed by Chang et al. [34] implements a flexible cluster merging mechanism through dynamic adaptive adjustment of features and clusters. Similarly, Huang et al. [30] proposed a method of deep semantic clustering by partition confidence maximization for image clustering, ensuring that semantically similar clusters are prioritized for merging.

Orthogonal to the two-stage and single-stage work, the research community has also investigated graph-based deep clustering, which aims to leverage the underlying structure of data for improved performance [10, 11, 35]. For instance, Chiang et al. [35] proposed a fast and memory-efficient method based on Graph Convolutional Networks (GCNs). Bo and Wang [10] presented the Structural Deep Clustering Network (SDCN), combining GCNs with the DEC framework to integrate structural information into deep clustering. Peng et al. [11] proposed the Attention-driven Graph Clustering Network (AGCN), which dynamically aggregates node attribute features and topological features, and adaptively fuses multi-scale features embedded in different layers. Li et al. [9] introduced Twin Contrastive Learning (TCL), which simultaneously conducts contrastive learning at both instance and cluster level. More recently, Huang et al. [36] proposed a deep clustering framework, DeepCluE, which integrates feature representations from multiple network layers, overcoming the limitations of traditional methods that mainly rely on single-level features. Yu et al. [12] also introduced Contextually Affinitive Neighborhood Refinery (CoNR), which refines

cluster assignments by online semantic re-ranking. Moreover, it employed a progressive boundary filtering mechanism to mitigate noisy neighbors.

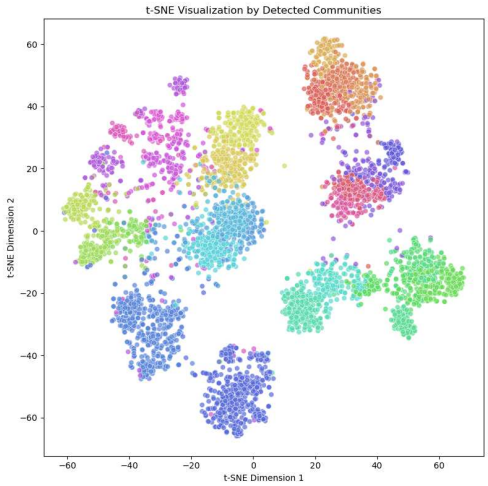


Fig. 2: Community visualization after partitioning an initial cluster with the Leiden algorithm: a few major communities and many smaller communities.

The above-mentioned works primarily operate in a single-view setting. Orthogonal to this, another emerging research direction is multi-view clustering, where data is represented by multiple feature sets. Recent work in the multi-view domain has tackled challenges such as noisy pseudo-labels (ROLL [37]) and fairness [38]. On the other hand, integrating textual information into image clustering to enhance semantic consistency and interpretability has become a research frontier in recent years. Approaches such as TeDeSC [13] explore how to discover semantic partitions through text-driven frameworks, while TAC [14] optimizes clustering decisions using external knowledge (e.g., WordNet). Similarly, Stephan et al. [15, 39] employ generated captions and VQA-style descriptions as an intermediate text space for clustering, and Cai et al. [40] fuse caption and visual features to obtain more semantically meaningful clusters. In a more controllable setting, $(IC\|TC)$ [41] conditions clustering on user-specified text criteria, and recent cross-modal methods such as SEIC [42] refine CLIP-based encoders by enforcing consistency between image and text semantics.

3 Methodology

As shown in Figure 1, in the initialization phase, DCvCD performs community detection within each initial cluster, and divides its samples into multiple pseudo-communities, the largest of which is selected as the main community. In the iterative phase, DCvCD first exploits the pseudo-label information of samples in main communities to fine-tune an RL backbone, and then performs community detection on all the unlabeled samples; finally, it merges pseudo-communities with main communities.

Since the DCvCD solution can in principle be built upon any existing RL backbone, and its initial clustering can leverage any clustering algorithm, the rest of this section focuses on the technical solutions of community detection & merging and backbone fine-tuning.

3.1 Community Detection & Merging

Community Detection: to enable community detection, DCvCD constructs a weighted network of samples within each initial cluster based on their representation similarity, in which edge weights are measured by the cosine similarity between samples. Specifically, it connects two samples by an edge if and only if their representation similarity exceeds a threshold (e.g., 0.5 in our implementation).

Then, DCvCD applies a community detection algorithm on each initial cluster to partition the samples into many pseudo-communities. Our implementation employs the Leiden algorithm. As illustrated in Figure **??**, the core principle of the Leiden algorithm is to ensure that the resulting communities are internally cohesive, typically yielding only a few large communities and a limited number of small ones. DCvCD selects the largest pseudo-community as the initial main community. It is noteworthy that the core advantage of the Leiden algorithm is highly aligned with DCvCD's goal of improving pseudo-label purity through global structure. Increasing modularity can directly enhance the correlation among samples within communities.

Technically, the Leiden algorithm identifies communities by maximizing modularity, which reflects the density of connections within communities versus the sparsity of connections between them. It is noteworthy that, as a more advanced algorithm than the traditional Louvain algorithm, the Leiden introduces a crucial refinement phase in its optimization process. This phase specifically addresses the inherent shortcoming of the Louvain algorithm, which can produce badly-connected communities that are not internally connected. Consequently, the Leiden algorithm theoretically guarantees that all identified communities are internally well-connected. This improvement makes the Leiden algorithm more reliable and robust for the clustering task.

Formally, the objective function optimized by the Leiden algorithm, modularity Q, is defined as:

$$Q = \frac{1}{2W} \sum_{i,j} \left[w_{ij} - \frac{k_i k_j}{2W} \right] \delta(x_i, x_j) \tag{1}$$

where W represents the sum of all edge weights in the network, i.e., $W = \sum_{i,j} w_{ij}$; w_{ij} is the weight of the edge between nodes i and j, with $w_{ij} = 0$ if no edge exists; k_i and k_j are the weighted degrees of nodes i and j respectively, with $k_i = \sum_j w_{ij}$; and $\delta(x_i, x_j)$ denotes the community indicator function, where $\delta(x_i, x_j) = 1$ if nodes i and j belong to the same community, and 0 otherwise.

To ensure high pseudo-label purity in initial main communities, DCvCD applies risk screening to filter out highly risky samples. Intuitively speaking, a sample is considered as risky if its degree of deviation from the community centroid is beyond a pre-specified confidence interval (e.g., 0.9 in our implementation) in terms of distance.

Community Merging: to effectively reduce the accumulation of pseudo-label errors during the iterative process of community merging, we present an ensemble metric,

consisting of both network structural indicators and traditional distance indicators, to optimize the merging of isolated communities with main communities.

Intuitively speaking, our solution aims to ensure that a merging operation maintains community structure consistency while minimizing the negative impact of misclassification. Towards this aim, our merging strategy not only considers the representation similarity between communities, but also incorporates network structural metrics such as modularity increment and average degree change to ensure the structural coherence of merged communities. Specifically, we define the guiding ensemble metric by

$$L = \frac{\Delta Q}{\max(\Delta Q)} + \frac{\Delta k}{\max(\Delta k)} - \frac{t}{\max(t)} \tag{2}$$

where L denotes the ensemble score, ΔQ denotes modularity increment after community merging, Δk denotes average degree change, and t denotes the measured distance between two communities. We elaborate each measure as follows:

- **Modularity Increment ΔQ.** Modularity increment is used to assess the overall optimization effect of merging on the structure of the main community. The modularity increment, ΔQ , measures the "quality" improvement of community structure after merging. A higher ΔQ indicates that the dense connectivity within a community after merging is significantly higher than expected in a random graph, implying that two communities have strong global similarity.
- **Average Degree Change Δk.** Besides the global quality improvement as measured by ΔQ, we also introduce the average degree increment, Δk, to measure the change in local connectivity, which can indicate the structural balance of a community after merging. A larger Δk indicates stronger node connectivity between two communities, suggesting higher similarity in their local structures.
- **Community Distance t.** As usual, the ensemble metric also uses the node distances between communities to assess positional similarity in the feature space, aiming to avoid the merging of communities that are far apart. Formally, the metric of t is defined as:

$$t = \frac{1}{|C_1| \cdot |C_2|} \sum_{x_i \in C_1} \sum_{y_j \in C_2} d(x_i, y_j) \tag{3}$$

  where C_1 and C_2 denote two communities, and $d(x_i, y_j)$ denotes the Euclidean distance between two nodes. It can be observed that a smaller t value indicates that the nodes are closer in feature space, which helps to avoid structural instability caused by long-distance merging.

In each iteration, for each main community, C_i, DCvCD selects an isolated community with the highest L metric value w.r.t. C_i and merges it with C_i. After community merging, DCvCD re-trains the RL backbone on the updated main communities, reconstructs a network on the nodes in isolated communities, and re-executes the Leiden algorithm.

Efficiency Analysis: theoretically, given a network with n nodes and m edges, the average-case time complexity of the Leiden algorithm is approximately $O(m+n\log n)$, making it highly efficient for large-scale networks. Furthermore, it can be observed

that after the first iteration, a majority of samples would be assigned to main communities; subsequent iterations would operate on a dramatically reduced number of unlabeled samples. As a result, the overhead of community detection would decrease dramatically after the first iteration. On the other hand, the computational complexity of community merging is approximately $O(n_i \cdot n_j)$, where n_i and n_j denote the numbers of nodes in the two communities respectively. In practice, since the size of most isolated communities is usually very small (i.e. in single digits), its overhead is usually negligible compared with that of community detection. Our efficiency evaluation results presented in the section of empirical study show that most of the DCvCD overhead is spent on backbone fine-tuning, and its total overhead is comparable to that of existing clustering approaches.

3.2 Backbone Fine-tuning

During each iteration, since the samples in the main community are regarded as reliable cluster members, DCvCD assigns pseudo-labels to these samples to guide RL backbone fine-tuning. The essential objective of fine-tuning is to make the generated embeddings better reflect the current cluster structure. To this end, as usual, DCvCD uses a clustering-aware loss function. Specifically, it adopts an adapted form of the InfoNCE (Noise-Contrastive Estimation) loss, which can be formally defined as follows:

$$\mathcal{L} = -\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(z_i, z_{p_i})/\tau)}{\exp(\text{sim}(z_i, z_{p_i})/\tau) + \sum_{j \in \mathcal{N}_i} \exp(\text{sim}(z_i, z_j)/\tau)} \tag{4}$$

Table 1: Comparative results of image-only clustering approaches on five image benchmarks, with the best results highlighted. "—" indicates that the experimental results or source code of these methods cannot be obtained.

Datasets	CIFAR-10(%)			CIFAR-100(%)			STL-10(%)			ImageNet-10(%)			ImageNet-Dogs(%)		
Metrics	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
DCGAN (2015)	26.5	31.5	17.6	12.0	15.1	4.50	21.0	29.8	13.9	22.5	34.6	15.7	12.1	17.4	7.8
JULE (2016)	19.2	27.2	13.8	7.50	13.7	3.30	18.2	27.7	16.4	17.5	30.0	13.8	5.4	13.8	2.8
DCCM (2019)	49.6	62.3	40.8	28.5	32.7	17.3	37.6	48.2	26.2	60.8	87.1	55.5	32.1	38.3	18.2
PICA (2020)	59.1	69.6	51.2	31.0	33.7	17.1	61.1	71.3	53.1	60.8	90.1	82.2	33.6	32.4	17.9
CC (2021)	70.5	79.0	63.7	43.1	42.9	26.6	76.4	85.0	72.6	80.2	87.0	76.1	44.5	42.9	27.4
CRLC (2021)	67.9	79.9	66.4	—	—	—	72.9	81.8	68.2	83.1	85.4	75.9	—	—	—
MICE (2021)	73.5	83.4	69.5	43.0	42.2	27.7	61.3	72.0	53.2	61.3	84.2	82.2	42.3	43.9	28.6
GCC (2021)	76.4	85.6	72.8	47.2	47.2	30.5	68.4	78.8	63.1	84.2	90.1	82.2	49.0	52.6	36.2
EDESC (2022)	46.4	62.7	—	37.0	38.5	—	68.7	74.5	—	—	—	—	—	—	—
DFVC (2022)	64.3	75.6	61.5	43.5	47.2	26.1	64.2	73.1	59.8	75.3	85.7	73.6	—	—	—
DCSC (2022)	70.4	79.6	64.4	45.2	46.9	29.3	79.2	86.5	74.9	—	—	—	—	—	—
TCL (2022)	81.9	88.7	78.0	50.8	51.2	36.7	79.9	86.8	75.7	87.5	89.5	83.7	62.3	64.4	51.6
SPICE (2022)	73.4	83.8	70.5	—	—	—	81.7	90.8	81.2	82.8	92.1	83.6	62.7	67.5	52.6
CCES (2023)	72.4	81.2	69.4	44.2	43.6	30.1	77.5	84.7	73.1	—	—	—	—	—	—
SeCu (2023)	79.9	88.5	78.2	51.6	51.6	36.0	70.7	81.4	65.7	—	—	—	—	—	—
DivClust (2023)	71.0	81.5	67.5	44.0	43.7	28.3	65.1	74.4	64.0	85.0	90.0	81.9	51.6	52.9	37.6
ProPos (2023)	88.6	94.3	88.4	51.5	53.2	37.6	75.8	86.7	73.7	89.6	95.6	90.6	69.2	74.5	62.7
CoNR (2023)	86.7	93.2	86.1	51.3	53.6	37.5	85.2	92.6	84.6	91.1	96.4	92.2	74.4	79.4	66.7
DeepCluE (2024)	72.7	76.4	64.6	47.2	45.7	28.8	—	—	—	88.2	92.4	85.6	44.8	41.6	27.3
GeCC (2025)	75.5	84.4	71.0	44.1	45.1	28.7	—	—	—	90.9	96.1	91.6	54.9	60.3	42.1
UACL (2025)	82.5	89.9	79.4	53.6	53.9	37.3	80.2	87.1	75.8	91.3	95.3	89.6	64.0	64.6	52.9
DCvCD (DivClust)	81.8	90.1	80.2	51.3	52.2	36.8	77.6	84.1	73.9	89.8	93.4	87.4	56.9	59.3	41.5
DCvCD (ProPos)	**90.7**	**95.3**	**89.6**	52.8	54.9	**39.1**	78.1	88.9	75.2	90.3	96.0	90.7	71.1	75.8	**68.3**
DCvCD (CoNR)	88.7	94.2	87.8	**53.6**	**55.8**	39.0	**87.2**	**94.0**	**85.8**	**92.0**	**97.5**	**93.1**	**76.2**	**81.0**	67.4

Where N denotes the number of training samples (i.e., pseudo-labeled samples), z_i denotes the embedding vector of node i, z_{p_i} denotes the embedding of a positive sample sharing the same cluster with z_i, $\mathcal{N}_i$ denotes the set of negative samples, $\text{sim}(\cdot, \cdot)$ denotes a

similarity function, e.g. cosine similarity, and τ denotes the temperature parameter scaling the similarity.

After the fine-tuning, DCvCD leverages the updated backbone to refresh the embeddings of the samples in the entire dataset. Then, it invokes a new iteration of community detection, community merging and backbone fine-tuning.

4 Empirical Study

In this section, we empirically evaluate the performance of our proposed approach on benchmark image datasets.

4.1 Experimental Setup

We have conducted experiments using five benchmark image datasets: CIFAR-10 [43], CIFAR-100 [43], STL-10 [44], ImageNet-10 [45] and ImageNet-Dogs [45], whose brief descriptions are as follows: 1) **CIFAR-10**: containing 60,000 color images in 10 classes, CIFAR-10 is widely used for image clustering evaluation; 2) **CIFAR-100**: containing 20 superclasses and 100 subclasses, CIFAR-100 is an extension of CIFAR-10. Due to its greater number of categories, CIFAR-100 is more challenging, often used for the evaluation of fine-grained clustering; 3) **STL-10**: containing 13,000 images across 10 classes, STL-10 is a popular benchmark derived from ImageNet, often used for unsupervised and self-supervised learning; 4) **ImageNet-10**: as a subset of ImageNet, ImageNet-10 contains 13,000 color images from 10 classes. It is usually used for quick image classification evaluation without requiring full training on the large ImageNet dataset; 5) **ImageNet-Dogs**: a subset of ImageNet used for fine-grained clustering, this dataset consists of 19,500 images from 15 different breeds of dogs, presenting a significant challenge due to the subtle inter-class variations.

Since traditional approaches, e.g., K-Means [46], SC [47], AC [48] and NMF [49], are not competitive compared with deep clustering methods, we compare our proposed DCvCD approach with 21 mainstream deep clustering alternatives, nearly two decades. The compared methods include DCGAN [50], JULE [28], DCCM [51], PICA [30], CC [7], CRLC [52], MICE [53], GCC [54], EDESC [55], DFVC [56], DCSC [57], and the most recently proposed SeCu [58], DivClust [59], CCES [60], DeepCluE [36], TCL [9], SPICE [25], ProPos [20], CoNR [12], GeCC [31] and UACL [32]. For a supplementary comparison against text-enhanced methods, please see Appendix A. As usual, we measure the performance of deep clustering by three metrics, Clustering Accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI).

It is noteworthy that for fair comparison, we do not directly compare DCvCD with the existing text-enhanced approaches, e.g., CLIP [16], SIC [61] and TAC [14], since they leverage additional external knowledge to facilitate clustering. However, we also present their performance results for reference purposes in the appendix due to space limit. The experimental results show that the performance of DCvCD is competitive compared with these text-enhanced approaches.

In the comparative study, our implementations of DCvCD employ DivClust, which is a primitive but popular algorithm, and ProPos and CoNR, which have reported SOTA results on the test datasets, for initial clustering, and their RL backbones for representation fine-tuning. ProPos and CoNR use the ResNet-34 and the ResNet-18 as their RL backbones respectively. For model fine-tuning, we use the SGD optimizer with an initial learning rate of 0.0001 to balance training stability and convergence speed. We set the batch size at 64, and the number of epochs at 100. In community detection and merging, the threshold of representation similarity for edge addition is set to the default value of 0.5. Our experimental

Table 2: The ablation evaluation results of the clustering strategy w.r.t. various baseline clustering approaches, in which the better results are highlighted

Method	CIFAR-10(%)	CIFAR-100(%)	STL-10(%)	ImageNet-10(%)	ImageNet-Dogs(%)
DivClust	81.5	43.7	74.4	90.0	52.9
DCvCD (DivClust)	**90.1 ± 1.2**	**52.2 ± 1.8**	**84.1 ± 1.0**	**93.4 ± 0.4**	**59.3 ± 0.7**
CC	79.0	42.9	85.0	87.0	42.9
DCvCD (CC)	**87.8 ± 0.6**	**50.5 ± 0.3**	**93.1 ± 0.6**	**90.1 ± 0.3**	**50.2 ± 0.5**
ProPos	94.3	53.2	86.7	95.6	74.5
DCvCD (ProPos)	**95.3 ± 0.2**	**54.9 ± 0.9**	**88.9 ± 0.3**	**96.0 ± 0.1**	**75.8 ± 0.3**
CoNR	93.2	53.6	92.6	96.4	79.4
DCvCD (CoNR)	**94.2 ± 0.3**	**55.8 ± 0.6**	**94.0 ± 0.2**	**97.5 ± 0.1**	**81.0 ± 0.1**
TCL	88.7	51.2	86.8	89.5	64.4
DCvCD (TCL)	**90.9 ± 1.1**	**53.7 ± 1.5**	**88.6 ± 0.7**	**92.5 ± 0.6**	**66.2 ± 0.9**

Table 3: The ablation evaluation results of the community merging metrics. The top block shows results on CIFAR datasets, while the bottom block shows results on STL and ImageNet datasets.

Method	CIFAR-10(%)			CIFAR-100(%)		
	NMI	ACC	ARI	NMI	ACC	ARI
DCvCD (w/o ΔQ)	85.5	91.5	84.0	50.1	52.1	35.0
DCvCD (w/o Δk)	87.1	92.8	86.1	52.6	54.7	37.8
DCvCD (w/o t)	86.5	92.4	85.3	51.9	53.8	36.9
DCvCD (Full)	**88.7**	**94.2**	**87.8**	**53.6**	**55.8**	**39.0**

Method	STL-10(%)			ImageNet-10(%)			ImageNet-Dogs(%)		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
DCvCD (w/o ΔQ)	84.0	90.8	82.1	88.8	93.6	88.4	72.5	77.0	63.7
DCvCD (w/o Δk)	86.2	92.9	84.5	91.0	95.9	91.5	75.2	79.9	65.9
DCvCD (w/o t)	85.5	92.1	83.7	90.5	95.5	91.0	75.0	79.3	65.1
DCvCD (Full)	**87.2**	**94.0**	**85.8**	**92.0**	**97.5**	**93.1**	**76.2**	**81.0**	**67.4**

Table 4: Runtime comparison between DCvCD and ProPos: averages and variances over 10 runs.

Method	CIFAR-10(h)	CIFAR-100(h)	STL-10(h)	ImageNet-10(h)
DCvCD	9.3 ± 0.9	10.2 ± 1.7	3.1 ± 0.7	4.0 ± 1.1
ProPos	10.2 ± 1.7	10.4 ± 1.8	14.2 ± 0.9	2.5 ± 1.4

results, which are put in the appendix due to space limit, show that the performance of DCvCD is robust w.r.t. the similarity threshold provided that it is set within a reasonable range. We have conducted all the experiments on an NVIDIA RTX A6000 GPU. As in previous work, we report the performance results averaged over 10 runs in the experiment.

4.2 Comparative Evaluation

The detailed comparative results have been reported in Table 1, in which the best results are **highlighted**. It can be observed that our proposed approach of DCvCD improves the SOTA performance on all the five datasets. Among the existing alternatives without leveraging external knowledge, ProPos and CoNR achieve the overall best performance. The corresponding DCvCD solutions still manage to improve performance. This observation clearly demonstrates that the proposed clustering strategy is complementary to the existing SOTA approaches.

Notably, compared to the more primitive approach of DivClust, the corresponding DCvCD(DivClust) solution shows considerable improvement across all the metrics. For instance, in terms of ACC, DCvCD(DivClust) outperforms DivClust by more than 8% on CIFAR-10, CIFAR-100 and STL-10.

It is interesting to point out that the improvement margins of DCvCD are not uniform across datasets: they tend to be larger when the initial clustering is relatively weak and more modest when the baseline is already very strong. Overall, the evaluation results clearly demonstrate that DCvCD, which introduces the perspective of network analysis into the clustering strategy, can effectively improve image clustering performance.

Table 5: Performance comparison of different community detection algorithms implemented in DCvCD(CoNR).

Algorithm	CIFAR-10(%)			CIFAR-100(%)		
	NMI	ACC	ARI	NMI	ACC	ARI
CoNR (Baseline)	86.7	93.2	86.1	51.3	53.6	37.5
Louvain	87.0	93.6	86.5	51.9	54.5	37.9
GN	86.8	93.4	86.2	51.5	54.0	37.6
DBSCAN	88.0	93.9	87.0	52.7	55.1	38.2
Leiden (Ours)	**88.7**	**94.2**	**87.8**	**53.6**	**55.8**	**39.0**

Algorithm	STL-10(%)			ImageNet-10(%)			ImageNet-Dogs(%)		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
CoNR (Baseline)	85.2	92.6	84.6	91.1	96.4	92.2	74.4	79.4	66.7
Louvain	85.7	93.4	85.0	91.3	96.9	92.5	74.8	80.2	67.0
GN	85.3	92.9	84.7	91.2	96.6	92.3	74.5	79.8	66.8
DBSCAN	86.9	93.6	85.5	91.8	97.2	92.8	75.5	80.6	67.3
Leiden (Ours)	**87.2**	**94.0**	**85.8**	**92.0**	**97.5**	**93.1**	**76.2**	**81.0**	**67.4**

4.3 Ablation Study

Clustering Strategy: to verify the benefit of the community-oriented clustering strategy, we implement the DCvCD solution based on various representative clustering methods, including CC (Contrastive Clustering) [7] and TCL (Twin Contrastive Learning) [9], besides DivClust, ProPos and CoNR as reported in Table 1, and compare their performance with its corresponding baselines in terms of ACC. To verify the statistical significance of DCvCD's superior performance, we also report its performance variances besides the averages over 10 runs. W.r.t. the compared baselines, since only the ProPos reports performance variances, we report both averages and variances on ProPos but only averages on other baselines.

The detailed comparison and evaluation results are presented in Table 2. It can be observed that DCvCD consistently outperforms its corresponding baseline on all the datasets. It is noteworthy that its margins in terms of averages comfortably exceed variances, validating statistical significance of DCvCD advantage. Furthermore, it is interesting to point out that the improvement margins are overall more considerable if the baseline underperforms. On three of the five datasets, CIFAR-10, CIFAR-100 and ImageNet-10, even beginning with the suboptimal clustering results of DivClust, DCvCD(DivClust) achieves competitive performance compared with DCvCD(ProPos) and DCvCD(CoNR). Our experimental results clearly demonstrate that the proposed community-oriented clustering strategy can effectively improve the performance of existing RL approaches. They bode well for its application in real scenarios.

Community Merging Metrics: to verify the efficacy of the three metrics used in community merging, we conduct an ablation study on their impact on the performance of DCvCD. The detailed evaluation results on DCvCD(CoNR) have been presented in Table **??**. The results on other DCvCD solutions are similar, thus omitted here. It can be observed that removing any of the three metrics, i.e., the modularity, the average degree and the distance, the performance of DCvCD would drop, with the modularity having the biggest adverse impact. These observations clearly demonstrate that the three metrics are complementary to each other and their ensemble achieves the best performance.

4.4 Efficiency Evaluation

To evaluate the efficiency of DCvCD clustering strategy, we compare its overhead with ProPos, one of only a few SOTA works that report runtime results. The detailed comparison is presented in Table 4, in which the overhead refers to the runtime of DCvCD after initial clustering is completed. The results are averages and variances over 10 runs.

It can be observed that: 1) on CIFAR-10, CIFAR-100, and ImageNet-10, the runtime of DCvCD is comparable to that of ProPos; 2) on the STL-10 dataset, DCvCD is significantly more efficient than ProPos. The main reason is that ProPos uses an additional 100,000 unlabeled images to optimize representation learning, whereas community detection and merging does not involve these extra data. These results demonstrate that the overhead of DCvCD is comparable to that of the existing clustering algorithms.

4.5 Choices of Community Detection Algorithm

To verify the efficacy of DCvCD w.r.t. different community detection algorithms and the superiority of the Leiden algorithm, we have conducted a comparative study using four representative algorithms: Leiden (the default), Louvain [62] (the traditional modularity-based algorithm), GN [63] (a graph-splitting-based algorithm), and DBSCAN [64] (a density-based algorithm). All the implementations are based on the CoNR, since CoNR achieves the best baseline performance.

We have presented the detailed evaluation results in Table 5. It can be observed that:

- DCvCD can effectively improve performance using any tested community detection algorithm. This result clearly validates the value of gradual community detection to cluster optimization, demonstrating the generality and flexibility of our proposed approach.
- Our choice of Leiden algorithm consistently achieves the best performance on clustering quality. Leiden's superior performance can be attributed to its explicit refinement of partitions and enforcement of well-connected communities, which can

effectively reduce poorly connected or fragmented clusters. Consequently, the communities found by Leiden can usually better align with semantic classes, and thus provide a stronger starting point for gradual community merging.

5 Conclusion and Future Work

In this paper, we propose a novel approach of deep clustering via gradual community detection, whose clustering strategy factors in cluster network analysis. Our extensive experiments on benchmark image datasets have validated its efficacy. It is noteworthy that the proposed DCvCD, as a clustering strategy, is a flexible approach. It can be easily built upon the existing representation learning approaches for deep clustering. The flexibility of DCvCD bodes well for its application in real scenarios.

On future work, it is worthy to extend DCvCD beyond the image-only setting by applying gradual community detection on multimodal similarity graphs, so that unsupervised clustering can benefit from complementary visual and textual information. Our current work separates the stages of representation learning and community detection. This strategy has the obvious benefit that it can easily leverage the existing RL backbones. However, in future work, it is interesting to investigate how to effectively align RL backbone design and training with community characteristics.

Declarations

Author contributions. Tianyu Cheng: Methodology, Validation, Writing – original draft, Writing – review & editing. Na Chen: Validation, Writing – review & editing. Siyi Yang: Supervision, Validation. Chuyi Fan: Supervision, Validation. Qun Chen: Methodology, Writing – review & editing.

Data availability. The datasets we used in the experiment, including CIFAR-10, CIFAR-100, STL-10, ImageNet-10, and ImageNet-Dogs, can be obtained from the references [43] [43] [44] [45] [45]. The aforementioned datasets are publicly available from the sources cited above. We use them in accordance with their respective licenses and usage policies.

Conflict of interests. The authors declare that they have no known competing financial interests or personal relationships that could have influenced the research reported in this paper.

Appendix A Comparison with Text-Enhanced Approaches

For completeness, we provide a comparison with text-enhanced clustering methods. It is noteworthy that, for fair comparison in our main evaluation, we do not directly compare DCvCD with these approaches, e.g., CLIP [16], SIC [61] and TAC [14], since they leverage additional external knowledge to facilitate clustering. However, we present their performance results alongside our best-performing model in Table A1 for reference. Since many of the existing approaches did not open-source their implementations, we directly use the performance results presented in their original papers for comparison.

Table A1: Comparative results between DCvCD and text-enhanced approaches. Note that despite the advantage of external knowledge in these approaches, DCvCD demonstrates competitive performance using only image data.

Datasets	CIFAR-10			STL-10			ImageNet-10		
Metrics	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
SIC (2023 AAAI)	84.7	92.6	84.4	95.3	98.1	95.9	97.0	98.2	96.1
CLIP (k-means)	70.3	74.2	61.6	91.7	94.3	89.1	96.9	98.2	96.0
TAC (no train)	80.8	90.1	79.8	92.3	94.5	89.5	97.5	98.6	97.0
TAC (2024 ICML)	83.3	91.9	83.1	**95.5**	**98.2**	**96.1**	**98.5**	**99.2**	**98.3**
DCvCD (CoNR)	**88.7**	**94.2**	**87.8**	87.2	94.0	85.8	92.0	97.5	93.1

Table A2: Performance of DCvCD on STL-10 with different similarity thresholds τ.

τ	ACC (%)	NMI (%)	ARI (%)	Efficiency(s)	Purity (%)
0.3	93.2	86.5	84.8	2.2	90.5
0.4	93.8	87.0	85.3	1.6	91.2
0.5	94.0	87.2	85.8	0.8	92.3
0.6	93.5	86.8	85.2	0.5	91.5
0.7	92.8	86.0	84.5	0.3	90.0

Table A3: Simulated performance of DCvCD on CIFAR-10 with different similarity thresholds τ.

τ	ACC (%)	NMI (%)	ARI (%)	Efficiency (s)	Purity (%)
0.3	92.8	87.0	85.8	14.5	89.5
0.4	93.7	88.0	86.9	10.5	90.8
0.5	94.2	88.7	87.8	7.2	92.0
0.6	93.6	88.3	87.3	5.8	91.3
0.7	92.9	87.5	86.5	5.0	90.2

Appendix B Hyperparameter Sensitivity Analysis

B.1 Parameter Sensitivity to Similarity Threshold

To evaluate how the similarity threshold τ impacts DCvCD's performance, we conduct experiments on STL-10 and CIFAR-10 datasets. We vary τ from 0.3 to 0.7 with a step of 0.1, while fixing other components: the community detection algorithm is Leiden, the RL backbone is CoNR, and the main community selection rule remains unchanged. We assess performance across three dimensions: **Clustering Performance**: ACC, NMI, and ARI; **Efficiency**: Time for a single community detection (seconds); **Pseudo-label Quality**: Purity of the main community (proportion of truly homogeneous samples).

B.1.1 Performance Trends Across Thresholds

Tables A2 and A3 summarize the results on STL-10 and CIFAR-10, respectively. Theoretically, the threshold τ controls the sparsity of the similarity graph. A value of $\tau = 0.5$ serves as a critical point that filters out noisy edges (which often correspond to inter-class connections with low cosine similarity) while preserving the intra-class manifold structure. Below this threshold, the Leiden algorithm may struggle to distinguish boundaries due to the 'resolution limit' problem in modularity maximization; above it, the graph becomes overly fragmented, preventing the formation of complete semantic clusters. Several key observations emerge: - **Peak Performance at Moderate** τ: Both datasets exhibit optimal clustering performance at $\tau = 0.5$. For example, on STL-10, ACC reaches 94.0%, NMI 87.2%, and ARI 85.8% — all the highest across all thresholds. This suggests $\tau = 0.5$ balances network connectivity and community granularity, enabling Leiden to detect cohesive communities with high pseudo-label purity (92.3% on STL-10).
- **Impact of Overly Loose/Tight Thresholds**: When τ is too low (e.g., 0.3), the network becomes overly dense, leading to coarse community partitions and lower pseudo-label purity (90.5%). Conversely, a too high τ (e.g., 0.7) results in a sparse network, causing community fragmentation and reduced clustering performance (ACC drops to 92.8% on STL-10).
- **High Computational Efficiency** It is worth noting that although the community detection step introduces computational overhead, it scales strictly linearly with the number of edges ($O(m)$) in the sparse graph constructed by the thresholding operation. As shown in Table 2, the time cost (e.g., 0.8s at $\tau = 0.5$) is negligible compared to the backbone fine-tuning stage, confirming that our topological refinement strategy is highly efficient.
- **Robustness Within a Range**: DCvCD maintains competitive performance for $\tau \in [0.4, 0.6]$ on both datasets. For instance, on CIFAR-10, ACC changes by less than 1 percentage point for $\tau \in [0.4, 0.6]$, indicating tolerance to moderate variations in τ.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant No. 2023YFB4503600).

References

[1] Dang, Z., Deng, C., Yang, X., Wei, K., Huang, H.: Nearest neighbor matching for deep clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13693–13702 (2021)

[2] Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., Van Gool, L.: Scan: Learning to classify images without labels. In: European Conference on Computer Vision, pp. 268–285 (2020). Springer

[3] Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 132–149 (2018)

[4] Choudhury, S.J., Pal, N.R.: Deep and structure-preserving autoencoders for clustering data with missing information. IEEE Transactions on Emerging Topics in Computational Intelligence **5**(4), 639–650 (2019)

[5] Fard, M.M., Thonet, T., Gaussier, E.: Deep k-means: Jointly clustering with k-means and learning representations. Pattern Recognition Letters **138**, 185–192 (2020)

[6] Guo, X., Gao, L., Liu, X., Yin, J.: Improved deep embedded clustering with local structure preservation. In: Ijcai, vol. 17, pp. 1753–1759 (2017)

[7] Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J.T., Peng, X.: Contrastive clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 8547–8555 (2021)

[8] Ren, Y., Pu, J., Yang, Z., Xu, J., Li, G., Pu, X., Philip, S.Y., He, L.: Deep clustering: A comprehensive survey. IEEE Transactions on Neural Networks and Learning Systems (2024)

[9] Li, Y., Yang, M., Peng, D., Li, T., Huang, J., Peng, X.: Twin contrastive learning for online clustering. International Journal of Computer Vision **130**(9), 2205–2221 (2022)

[10] Bo, D., Wang, X., Shi, C., Zhu, M., Lu, E., Cui, P.: Structural deep clustering network. In: Proceedings of the Web Conference 2020, pp. 1400–1410 (2020)

[11] Peng, Z., Liu, H., Jia, Y., Hou, J.: Attention-driven graph clustering network. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 935–943 (2021)

[12] Yu, C., Shi, Y., Wang, J.: Contextually affinitive neighborhood refinery for deep clustering. Advances in Neural Information Processing Systems **36** (2024)

[13] Liu, M., Zhong, Z., Li, J., Franchi, G., Roy, S., Ricci, E.: Organizing unstructured image collections using natural language. arXiv preprint arXiv:2410.05217 (2024)

[14] Li, Y., Hu, P., Peng, D., Lv, J., Fan, J., Peng, X.: Image clustering with external guidance. arXiv preprint arXiv:2310.11989 (2023)

[15] Stephan, A., Miklautz, L., Sidak, K., Wahle, J.P., Gipp, B., Plant, C., Roth, B.: Text-guided image clustering. arXiv preprint arXiv:2402.02996 (2024)

[16] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., *et al.*: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021). PmLR

[17] Jiang, Z., Wang, Y., Davis, L., Andrews, W., Rozgic, V.: Learning discriminative features via label consistent neural network. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 207–216 (2017). IEEE

[18] Misra, I., Maaten, L.v.d.: Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6707–6717 (2020)

[19] Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Scientific reports **9**(1), 1–12 (2019)

[20] Huang, Z., Chen, J., Zhang, J., Shan, H.: Learning representation for clustering via prototype scattering and positive sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(6), 7509–7524 (2022)

[21] Cai, D., He, X., Wang, X., Bao, H., Han, J.: Locality preserving nonnegative matrix factorization. In: IJCAI, vol. 9, pp. 1010–1015 (2009). Citeseer

[22] Yang, J., Lin, C.-T.: Multi-view adjacency-constrained hierarchical clustering. IEEE Transactions on Emerging Topics in Computational Intelligence **7**(4), 1126–1138 (2022)

[23] Fang, S.-G., Huang, D., Wang, C.-D., Tang, Y.: Joint multi-view unsupervised feature selection and graph learning. IEEE Transactions on Emerging Topics in Computational Intelligence (2023)

[24] Deng, X., Huang, D., Chen, D.-H., Wang, C.-D., Lai, J.-H.: Strongly augmented contrastive clustering. Pattern Recognition **139**, 109470 (2023)

[25] Niu, C., Shan, H., Wang, G.: Spice: Semantic pseudo-labeling for image clustering. IEEE Transactions on Image Processing **31**, 7264–7278 (2022)

[26] Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In: International Conference on Machine Learning, pp. 3861–3870 (2017). PMLR

[27] Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487 (2016). PMLR

[28] Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5147–5156 (2016)

[29] Ghasedi Dizaji, K., Herandi, A., Deng, C., Cai, W., Huang, H.: Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5736–5745 (2017)

[30] Huang, J., Gong, S., Zhu, X.: Deep semantic clustering by partition confidence maximisation. In: Proceedings of the IEEE/CVF Conference on Computer Vision

and Pattern Recognition, pp. 8849–8858 (2020)

[31] Chen, Y., Wu, W., Ou-Yang, L., Wang, R., Wang, D.D.: Gecc: Generalized contrastive clustering with domain shifts modeling. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 15966–15974 (2025)

[32] Chang, L., Chen, L., Zhou, C.: Uncertainty-aware contrastive learning for deep clustering. Neurocomputing, 130568 (2025)

[33] Wu, J., Li, Z., Sun, W., Yin, J., Nie, L., Lin, Z.: Clusmatch: Improving deep clustering by unified positive and negative pseudo-label learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)

[34] Chang, J., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep adaptive image clustering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5879–5887 (2017)

[35] Chiang, W.-L., Liu, X., Si, S., Li, Y., Bengio, S., Hsieh, C.-J.: Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 257–266 (2019)

[36] Huang, D., Chen, D.-H., Chen, X., Wang, C.-D., Lai, J.-H.: Deepclue: Enhanced deep clustering via multi-layer ensembles in neural networks. IEEE Transactions on Emerging Topics in Computational Intelligence (2024)

[37] Sun, Y., Li, Y., Ren, Z., Duan, G., Peng, D., Hu, P.: Roll: Robust noisy pseudo-label learning for multi-view clustering with noisy correspondence. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 30732–30741 (2025)

[38] Xu, H., Wang, Q., Wang, B., Gao, Q.: Deep fair multi-view clustering with attention kan. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 5061–5070 (2025)

[39] Stephan, A., Miklautz, L., Leiber, C., De Araujo, P.H.L., Repas, D., Plant, C., Roth, B.: Text-guided alternative image clustering. arXiv preprint arXiv:2406.18589 (2024)

[40] Cai, Y., Kosugi, S., Funakoshi, K., Okumura, M.: Enhancing image clustering with captions. In: Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation, pp. 246–255 (2024)

[41] Kwon, S., Park, J., Kim, M., Cho, J., Ryu, E.K., Lee, K.: Image clustering conditioned on text criteria. arXiv preprint arXiv:2310.18297 (2023)

[42] Li, Z., Sun, W., Hu, J., Yin, J., Wu, J., Nie, L.: Self-enhanced image clustering

with cross-modal semantic consistency. arXiv preprint arXiv:2508.01254 (2025)

[43] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[44] Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 215–223 (2011). JMLR Workshop and Conference Proceedings

[45] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255 (2009)

[46] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press (1967)

[47] Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. Advances in neural information processing systems **17** (2004)

[48] Gowda, K.C., Krishna, G.: Agglomerative clustering using the concept of mutual nearest neighbourhood. Pattern recognition **10**(2), 105–112 (1978)

[49] Zheng, C.-H., Huang, D.-S., Zhang, L., Kong, X.-Z.: Tumor clustering using nonnegative matrix factorization with gene selection. IEEE Transactions on Information Technology in Biomedicine **13**(4), 599–607 (2009)

[50] Radford, A.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)

[51] Wu, J., Long, K., Wang, F., Qian, C., Li, C., Lin, Z., Zha, H.: Deep comprehensive correlation mining for image clustering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8150–8159 (2019)

[52] Do, K., Tran, T., Venkatesh, S.: Clustering by maximizing mutual information across views. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9928–9938 (2021)

[53] Tsai, T.W., Li, C., Zhu, J.: Mice: Mixture of contrastive experts for unsupervised image clustering. In: International Conference on Learning Representations (2020)

[54] Zhong, H., Wu, J., Chen, C., Huang, J., Deng, M., Nie, L., Lin, Z., Hua, X.-S.: Graph contrastive clustering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9224–9233 (2021)

[55] Cai, J., Fan, J., Guo, W., Wang, S., Zhang, Y., Zhang, Z.: Efficient deep embedded

subspace clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1–10 (2022)

[56] Ji, Q., Sun, Y., Gao, J., Hu, Y., Yin, B.: A decoder-free variational deep embedding for unsupervised clustering. IEEE Transactions on Neural Networks and Learning Systems **33**(10), 5681–5693 (2021)

[57] Zhang, F., Li, L., Hua, Q., Dong, C.-R., Lim, B.-H.: Improved deep clustering model based on semantic consistency for image clustering. Knowledge-Based Systems **253**, 109507 (2022)

[58] Qian, Q.: Stable cluster discrimination for deep clustering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16645–16654 (2023)

[59] Metaxas, I.M., Tzimiropoulos, G., Patras, I.: Divclust: Controlling diversity in deep clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3418–3428 (2023)

[60] Yin, J., Wu, H., Sun, S.: Effective sample pairs based contrastive learning for clustering. Information Fusion **99**, 101899 (2023)

[61] Cai, S., Qiu, L., Chen, X., Zhang, Q., Chen, L.: Semantic-enhanced image clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 6869–6878 (2023)

[62] Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment **2008**(10), 10008 (2008)

[63] Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proceedings of the national academy of sciences **99**(12), 7821–7826 (2002)

[64] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., *et al.*: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96, pp. 226–231 (1996)