Soft Neighbors Supported Contrastive Clustering

Yu Duan, Huimin Chen, Runxin Zhang, Rong Wang, Feiping Nie*, *Senior Member, IEEE*,
Xuelong Li, *Fellow, IEEE*

***Abstract*—Existing deep clustering methods leverage contrastive or non-contrastive learning to facilitate downstream tasks. Most contrastive-based methods typically learn representations by comparing positive pairs (two views of the same sample) against negative pairs (views of different samples). However, we spot that this *hard treatment* of samples inadequately models inter-sample relationships, leading to class collision and degraded clustering performance. In this paper, we propose a soft neighbor supported contrastive clustering method to address this issue. Specifically, we propose the *perception radius* concept to quantify similarity confidence between a sample and its neighbors. Building on this insight, we design a two-level soft neighbor loss that captures both local and global neighborhood relationships. Additionally, a cluster-level loss enforces compact and well-separated cluster distributions. Finally, we introduce a pseudo-label refinement strategy to mitigate false negative samples. Extensive experiments on benchmark datasets demonstrate the superiority of our method. The code is available at https://github.com/DuannYu/soft-neighbors–supported-clustering.**

***Index Terms*—Deep Clustering, Contrastive Learning, Soft Neighbors, Unsupervised Learning**

I. Introduction

THE exponential growth of unlabeled data, particularly visual data, has created an urgent need for unsupervised grouping techniques. As a fundamental unsupervised learning method, clustering naturally addresses this challenge through its intrinsic grouping mechanism. Consequently, diverse clustering methods have emerged from various perspectives, including K-means [1], fuzzy c-means [2], spectral clustering, subspace clustering [3], affinity propagation [4], non-negative matrix factorization [5], [6], Gaussian mixture [7], and other graph-based methods [8], [9].

From the data acquisition perspective, clustering has been extended to multi-view settings, where cluster assignments are made by integrating information from different views of the same data [10], [11]. Recent studies have explored more challenging scenarios involving heterogeneous data across views. A typical example is Electronic Health Records (EHR), comprising both structured demographic attributes (e.g., age, gender) and unstructured diagnostic code sets. This inherent heterogeneity between single-valued attributes and set-valued attributes presents significant challenges for conventional multi-view clustering methods. To overcome this limitation, Zhong *et al.* [12], [13] developed a novel clustering method that effectively handles mixed demographic and diagnostic code data. However, these conventional methods mainly operate on low-dimensional or raw features and lack representation learning capability, leading to suboptimal performance.

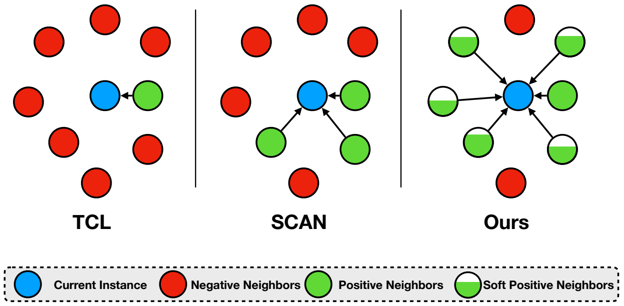


Fig. 1. Illustration of three contrastive clustering strategies. For a current sample, TCL [22] only considers its augmentation as positive sample that is also the nearest neighbor. SCAN [23] lets its several nearest neighbors as positive ones. In this paper, our basic idea is that we regard neighbors as soft neighbors, which are partialy correlated to the current sample, and contribute to contrastive learning limitly.

In recent years, deep learning has gained considerable attention owing to its powerful representation learning capability. Numerous deep clustering methods, which integrate deep learning with clustering, have been developed [14]–[18]. These methods non-linearly transform data into a latent feature space, thereby enhancing cluster analysis performance. Most deep clustering methods typically comprise two key components: representation learning and cluster assignment. Among the pioneering deep clustering methods, Xie *et al.* [19] introduced Deep Embedded Clustering (DEC), which employs a KL divergence-based objective. Building upon DEC, IDEC [20] jointly optimizes feature representation and cluster assignment while preserving local structures. Ji *et al.* [21] developed the Invariant Information Clustering (IIC) method, which maximizes mutual information between images and their augmented versions.

Nowadays, contrastive learning has significantly improved the performances of unsupervised learning. Motivated by this, several contrastive-based clustering methods have emerged, achieving state-of-the-art performance [23]–[27]. For example, Li *et al.* [22] jointly learned the feature and cluster assignment

Yu Duan is with the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China. (E-mail: duanyuee@gmail.com)

Huimin Chen and Feiping Nie are with the School of Computer Science, School of Artificial Intelligence, Optics and Electronics (iOPEN), and the Key Laboratory of Intelligent Interaction and Applications (Ministry of Industry and Information Technology), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China. (E-mail: duanyuee@gmail.com; chenhuimin@mail.nwpu.edu.cn; feipingnie@gmail.com).

Runxin Zhang, Rong Wang and Xuelong Li are with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P.R. China, and also with the Key Laboratory of Intelligent Interaction and Applications (Northwestern Polytechnical University), Ministry of Industry and Information Technology, Xi'an 710072, P.R. China. (email: zhangrunxin66@gmail.com; wangrong07@tsinghua.org.cn; email: li@nwpu.edu.cn).

* is corresponding author.

at the same time, without any explicit clustering process. Another contrastive-based method is called SCAN [23], which uses the learned representations from a pre-text task, and finally obtained cluster assignment by using nearest neighbors. On the other hand, Huang [28] divided contrastive learning into two steps: *alignment* and *uniformity*. The *alignment* step can make positive pairs closer. The *Uniformity* step encourages negative pairs to uniformly scatter on the unit hyper-sphere as much as possible, to achieve the purpose of pushing away negative pairs. Despite their success, it is inevitable that the samples belonging to the same class maybe wrongly pushed away, leading to sub-optimal performance.

To address these limitations, we propose Soft Neighbor-Supported Contrastive Clustering, a novel method that intelligently incorporates soft neighborhood relationships into contrastive learning. As illustrated in Fig. 1, our method introduces adaptive soft positive neighbors for more robust contrastive loss computation. Specifically, we develop:

- A neighbor positiveness measurement strategy that dynamically evaluates sample relationships, effectively mitigating false negative effects.
- A consistency loss that simultaneously enhances intra-class compactness and inter-class separation.
- A pseudo-label guided contrastive module that progressively refines neighbor selection through self-supervised learning

The complete framework generates cluster assignments end-to-end upon convergence. Our key contributions include:

- We propose a new clustering method called soft neighbors supported contrastive clustering, which could consider positive and negative pairs more properly, and learn more cluster-favorable features.
- We also design a boosting strategy to improve clustering performance based on pseudo-labels. Experimental results show that these strategy could fine-tune our models and enhance clustering results.
- Extensive experimental results on several benchmark datasets demonstrate that our proposed method outperforms the existing state-of-the-art methods by a significant margin.

The rest of this paper is organized as follows: Section II briefly review the related works. We discuss proposed method and its optimization in Section III. Experimental results are reported in Section IV. Finally, we conclude the whole paper in Section V.

II. Related Works

A. Contrastive Learning

Contrastive learning, as a metric learning method, has been successfully applied in unsupervised, semi-supervised and supervised learning tasks [29]–[31]. It firstly builds positive and negative pairs for each sample, then maps them into feature spaces, maximizing similarities between positive pairs and minimizing the negative ones. Hence, the selection of positive/negative pairs is crucial for contrastive learning. For example, in SimCLR [32], the positive sample is obtained through image augmentation, while negatives are randomly sampled within mini-batches. It has been demonstrated that larger batch sizes (more negative pairs) can yield better performance. However, excessively large batches leads expensive storage and computational requirements. To consider more negative samples, MoCo [33] treats contrastive learning as a dictionary lookup process, by utilizing an memory bank and a moving-averaged encoder. On the other hand, to avoid the side-effects from building negative pairs, BYOL [34] uses the teacher-student network to replace choosing negative samples, updating the network in a moving-average manner and avoiding trivial solutions. Even without using negative pairs, large batch size, and momentum encoders, SimSiam [35] still shows that simple siamese networks can learn meaningful representations.

B. Deep Clustering

In contrast to contrastive learning that treats each sample as an independent class, deep clustering aims to group similar samples into the same category. For example, DEC [19] and IDEC [20], as typical deep clustering methods, use auto-encoders for representation learnings, apply k-means to initialize cluster centers, and then compute KL divergence to train networks. Peng *et al.* [36] proposed a new subspace clustering method, which solves the drawback of dealing with non-linear structures. JULE [14] learns CNN for representation learning and hierarchical clustering in a recurrent manner. DAC [15], DDC [37] and DCCM [18] alternately optimize the inter-sample relationships and clustering assignment during training.

Furthermore, contrastive-based clustering methods have achieved great improvements. Some of them use pretext tasks learning discriminative features to assist downstream clustering tasks [22], [23], [38]. Others combine representation learning and clustering assignments together, jointly optimize networks until convergence, and finally obtain the cluster predictions [25], [27], [39]–[41]. To name a few, IDFD [27] proposes to perform both sample discrimination and feature decorrelation. Peng *et al.* [22] proposed TCL, which considers instance-level and cluster-level features together, adopts the features as a prior, and fine-tunes results in a supervised manner. SCAN [23] is similar to TCL, but it mainly uses nearest neighbors to learn cluster-favorable representations.

Moreover, researchers have extended contrastive-based clustering to a multi-view manner. For example, Xu *et al.* [42] proposed a framework for multi-view clustering that incorporates multi-level representation learning. Specifically, it learns multiple levels of features for each view, including low-level features, high-level features, and semantic labels in a fusion-free manner. Pan *et al.* [43] proposed a Multi-view Contrastive Graph Clustering (MCGC) to learn a consensus graph by exploiting not only attribute content but also graph structure information. Yang *et al.* propose a novel end-to-end deep multi-view clustering framework, which has multiple single-view clustering tasks and one multi-view clustering task. Therefore, it is employed to harvest the complementary and consistent information of multi-view data.

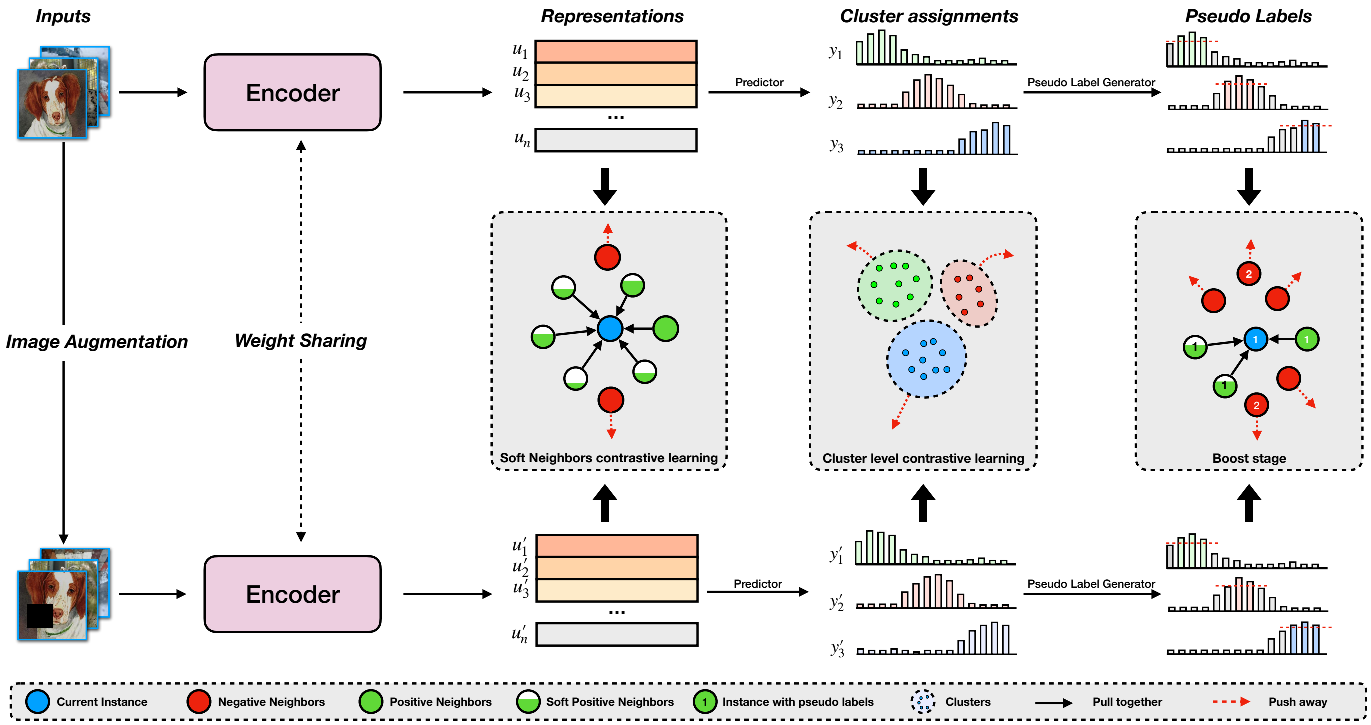


Fig. 2. The overview of our proposed method. In `Pre-train Stage`, we simply ultilize self-supervised learning to pre-train the encoder (Section III-A). In `Train Stage`, We leverage soft neighbors supported contrastive loss to learning cluster-favorable representations and use cluster-level contrastive learning to obtain the cluster assignments (Section III-B). In `Boost Stage`, we propose a strategy to generate pseudo labels, and finally fine-tune the cluster predictions (Section III-C).

III. Methods

Our method significantly differs from the aforementioned methods. It incorporates the soft neighbor strategy and considers more potential positive samples in contrastive learning. It finally obtains cluster-favorable representations and improve clustering performances.

In sum, our proposed method is mainly consist of following three parts. As illustrated in Fig. 2, we first pre-train the encoder in Section III-A with an unsupervised manner. Then we discuss the soft neighbors supported contrastive clustering in Section III-B. At last, we provide a boosting strategy based on pseudo-labels to further improve the clustering performance in Section III-C.

A. Self-supervised Model Pre-training

As for unsupervised learning task, we empirically know that the initialization of the network is important to the final performance. In order to get more cluster-favorable features from raw images, here we use the contrastive learning to pre-train the network. Given an input sample $\mathbf{x}_i$ and its augmented $\mathbf{x}_i'$ from batch $\mathcal{B}$, we obtain their semantic features $\mathbf{u}$ and $\mathbf{u}'$ after encoders respectively, the contrastive loss can be generally written as:

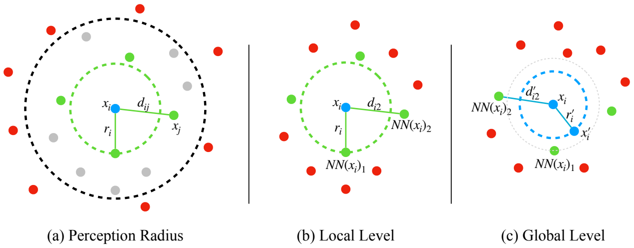
$$\mathcal{L}_{pre-train} = -\log \frac{\exp(sim(\mathbf{u}_i, \mathbf{u}_i')/\tau)}{\sum_{j=1}^{2|\mathcal{B}|} \mathbb{1}_{j\neq i} \exp(sim(\mathbf{u}_i, \mathbf{u}_j)/\tau)}, \tag{1}$$

where τ is a temperature parameter, $\mathbf{u}_i$ and $\mathbf{u}_i'$ are positive pairs, $\mathbb{1}_{j\neq i}$ is a conditional funciton, when $j \neq i$, it returns 1 and 0 otherwise. The $sim(\cdot,\cdot)$ is the function for similarity measurement like cosine similarity. After computing the loss, we use gradient descent to train the whole networks.

B. Soft Neighbors Supported Contrastive Clustering

Before introducing the proposed method, we first give an observation that most contrastive learning often regards the nearest features as the positive sample, and all the others are negative. It only pulls two samples together and pushes all the others away. However, this *hard negativity* often ignores the fact that the sample itself and neighbors potentially belong to the same category, degrading downstream tasks, e.g. classification and clustering [44]. A very naive idea is considering more neighbors as positive pairs. This intuition is well demonstrated in SCAN. In this method, neighbors are regarded as either positive or not to contribute to the current sample. We observe that this is incorrect because the neighbors are partially correlated for the current sample. In this paper, we propose a concept of *soft positive neighbors* based on the neighbor relationship. And we give an independent confidence strategy to measure the correlations between sample and its neighbors. Now, there are two problems before us: *a) how to choose suitable neighbors*, and *b) how to measure the positiveness of neighbors*.

To choose suitable neighbors, we simply feed all samples from batches into the encoder to obtain features. Specifically, let the current sample be $\mathbf{x}_i$, we select top-K nearest neighbors in this feature space, denoted as $NN(\mathbf{u}_i)_k, \forall k \in \{1, 2, ..., K\}$.



(a) Perception Radius (b) Local Level (c) Global Level

Fig. 3. The key idea of the perception radius. (a) Given a sample $\mathbf{x}_i$, the distance between $\mathbf{x}_i$ and its nearest sample is defined the *perception radius*. Only the distance is ranging from $\mathbf{r}_i$ to $2\mathbf{r}_i$ could be regarded as soft neighbors. (b) At Local Level, the distance between $\mathbf{x}_i$ and its nearest neighbor is called its *local perception radius*. (c) At Global Level, the distance between $\mathbf{x}_i$ and its augmented view is defined as *global perception radius*.

As for k-th nearest neighbors $NN(\mathbf{u}_i)_k$, we define its positiveness confidence $\mathbf{s}_{ik} \in [0, 1]$, guiding the training process for contrastive learning.

Now, we will discuss how to calculate the positivity confidence of neighbors. Firstly, we give a definition for the sample's *perception radius*.

Definition 1: For a sample $\mathbf{x}_i$, the distance to its nearest neighbor $NN(\mathbf{u}_i)_1$ is its *perception radius*, denoted as $\mathbf{r}_i$.

Perception radius describes the ability of each sample to perceive its local information. Fig. 3 shows an intuitive examples of *perception radius*. According to Definition 1, we could define $\mathbf{s}_{ik} \in [0, 1]$ as the positiveness confidence of k-th nearest neighbors to $\mathbf{x}_i$, which can be calculated as

$$\mathbf{s}_{ik} = \text{clip}(1 - \frac{\mathbf{d}_{ik} - \mathbf{r}_i}{\mathbf{r}_i}, 0, 1), \tag{2}$$

where $\mathbf{d}_{ik}$ is the distance between $\mathbf{u}_i$ and $NN(\mathbf{u}_i)_k$, and clip$(\cdot)$ removes the incentive for moving $\mathbf{s}_{ik}$ outside of the interval $[0, 1]$. We could use this value to adaptively adjust the contribution of neighbors to the current sample during contrastive learning.

The mined neighbors are very important supervised information for the current sample. It directly determines the radius of the sample and further impact the positiveness. Next, we will introduce the neighbors' search and the objective function from both local and global perspectives.

1) Local Soft Positive Neighbors: For the local-level, we calculate the pairwise distance of all embeddings in a batch, to obtain the radius $\mathbf{r}_i$ of each sample $\mathbf{u}_i$ and its neighbors $NN(\mathbf{u}_i)_k$ with positiveness $\mathbf{s}_{ik}$. Then we obtain the corresponding clustering assignment $\mathbf{p}_i \in \mathbb{R}^C$, $NN(\mathbf{p}_i)_j \in \mathbb{R}^c$ by feeding features into the predictor respectively, where C is predefined cluster numbers. Finally, the local-level soft contrastive loss can be written as

$$\mathcal{L}_{local} = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{K} \mathbf{s}_{ij} \log \mathbf{O}_{ij} + (1 - \mathbf{s}_{ij}) \log (1 - \mathbf{O}_{ij}), \tag{3}$$

where $\mathbf{O}_{ij} = \langle \mathbf{p}_i, NN(\mathbf{p}_i)_j \rangle$ and $\langle \cdot, \cdot \rangle$ is matrix inner product to measure the similarity.

2) Global-Level Soft Positive Neighbors: The local level neighbors only consider the pairwise distance in a batch. According to the previous works [22], [26], an important observation is that the nearest neighbors of a sample are its augmented features. Under this context, the true or global *perception radius* of any sample is determined by its own augmentations. Formally, the global *perception radius* of the sample is defined as

$$\mathbf{r}'_i = ||\mathbf{u}_i - \mathbf{u}'_i||, \tag{4}$$

where $\mathbf{u}'_i$ is the feature encoded from augmented sample $\mathbf{x}'_i$. Different from local-level loss, we calculate the pairwise distance between original and augmented samples, and further obtain the global-level positiveness confidence $\mathbf{s}'_{ik} \in [0, 1]$ as the positiveness confidence of k-th nearest neighbors to $\mathbf{x}_i$, which can be written as

$$\mathbf{s}'_{ik} = \text{clip}(1 - \frac{\mathbf{d}'_{ik} - \mathbf{r}'_i}{\mathbf{r}'_i}, 0, 1), \tag{5}$$

Based on this, assume that the cluster assignment of $\mathbf{x}'$ is $\mathbf{p}' \in \mathbb{R}^{B \times C}$ we can obtain pairwise distance $\mathbf{p}$ and $\mathbf{p}'$, then further get global soft contrastive loss

$$\mathcal{L}_{global} = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{K} \mathbf{s}'_{ij} \log \mathbf{O}'_{ij} + (1 - \mathbf{s}'_{ij}) \log (1 - \mathbf{O}'_{ij}), \tag{6}$$

where $\mathbf{O}' \in \mathbb{R}^{|\mathcal{B}| \times |\mathcal{B}|}$, and its (i, j)-th elements is also $\langle \mathbf{p}_i, NN(\mathbf{p}_i)_j \rangle$.

3) Cluster-Level Loss: Once obtaining the sample assignment, we need to discuss the relationship between clusters from another perspective. Different from local and global levels, it is intuitive to see that each cluster cannot belong to the same category. In other words, they regard other clusters as negative samples. Formally, define $\mathbf{q} = \mathbf{p}^T \in \mathbb{R}^{C \times B}$ which can be seen as a special distribution of clusters. Our task is to maximize the difference between each cluster. With this insight, we could write a cluster-level contrastive loss as follow

$$\mathcal{L}_{clu\_cont} = -\log \frac{\exp(sim(\mathbf{q}_i, \mathbf{q}'_i)/\tau)}{\sum_{j=1}^{2C} \mathbb{1}_{j \neq i} \exp(sim(\mathbf{q}_i, \mathbf{q}_j)/\tau)}, \tag{7}$$

where τ is a temperature parameter. Besides above contrastive loss, we also introduce the widely used entropy loss to avoid trivial solutions of the model. That is, all samples belong to the same category. The entropy loss is written as follow:

$$\mathcal{L}_{entropy} = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \mathbf{p}_i \log(\mathbf{p}_i). \tag{8}$$

Thus, the cluster-level contrastive loss is finally defined as

$$\mathcal{L}_{clu} = \mathcal{L}_{clu\_cont} - \epsilon \mathcal{L}_{entropy}. \tag{9}$$

Finally, The entire loss at `Train Stage` can be computed as

$$\mathcal{L}_{train} = \mathcal{L}_{clu} + \alpha_1 \mathcal{L}_{local} + \alpha_2 \mathcal{L}_{global}, \tag{10}$$

where α_1 and α_2 are trade-off parameters.

C. Boost Stage

We experimentally observed that as the training goes on, the clustering assignments will become clear, which are always correct. Based on this observation, we select assignments with high confidence as pseudo-labels to fine-tune our prediction results. We call the above process as `Boost Stage`. Specifically, we first select samples whose clustering assignment is higher than a certain threshold γ as their candidate pseudo-labels $\hat{\mathbf{y}}$. Among all these candidate pseudo-labels, we could leverage following strategy to obtain the final pseudo-labels $\mathbf{y}$,

$$\mathbf{y}_i = \text{TOP}(\hat{\mathbf{y}}_i, \sigma\%), \quad s.t.\ \hat{\mathbf{y}}_i \in \{\mathbf{p}_i | \mathbf{p}_i \geq \gamma\}, \tag{11}$$

where $\text{TOP}(\mathbf{p}_i, \sigma\%)$ is a selection function that outputs the largest the most $\sigma\%$ confident samples in each class.

Once the pseudo-labels $\mathbf{y}_i$ are obtained, we are able to give a soft contrastive loss between samples in the `Boost Stage`. Unlike Eq. (3) and Eq. (6), soft neighbors must be selected from the same category according to the pseudo-labels. Formally, for each sample $\mathbf{x}_i$, the boosting soft contrastive loss is

$$\mathcal{L}_{bs} = -\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\sum_{j\in\{j|\mathbf{y}_i=\mathbf{y}_j\}} \mathbf{s}_{ij}\log\mathbf{O}_{ij} + (1-\mathbf{s}_{ij})\log(1-\mathbf{O}_{ij}). \tag{12}$$

At the same time, using the obtained pseudo-labels, we could train the network in a supervised manner. Given $\mathbf{p}_i$ be the i-th sample's clustering assignment, we could define the boosting loss as follow,

$$\mathcal{L}_{pesudo} = -\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\texttt{CrossEntropy}(\mathbf{y}_i, \mathbf{p}_i), \tag{13}$$

where $\texttt{CrossEntropy}(\cdot,\cdot)$ is cross entropy loss. Finally, the entire loss in this `Boost Stage` can be computed as

$$\mathcal{L}_{boost} = \beta_1\mathcal{L}_{bs} + \beta_2\mathcal{L}_{pesudo}, \tag{14}$$

where β_1 and β_2 are trade-off parameters. Finally, the whole processes of the proposed method are summarized in Algorithm 1.

Algorithm 1 Soft Positive Neighbors Contrastive Clustering

```
Input : Dataset $X$; clusters $c$; Pre-train/Train/Boost iterations,
        Pre-train/Train/Boost batch size; cross-entropy hyper-
        parameter $\epsilon$, threshold $\gamma$; image augmentation strat-
        egy.
Output : Clustering assignments.
/* Pre-train Stage                                              */
Pre-train the model using loss $\mathcal{L}_{pre-train}$.
/* Train Stage                                                  */
for $epoch = 1$ to $MAX\_EPOCH$ do
    Sample a mini-batch from whole datasets.
    Feed samples into encoders get representations.
    Compute local and global level soft neighbors loss accord-
      ing to Eq. (3), Eq. (6).
    Compute cluster-level loss by Eq. (9).
    Compute training loss by Eq. (10).
    Update network weight to minimize $\mathcal{L}_{train}$.
/* Boost Stage                                                  */
for $epoch = 1$ to $MAX\_EPOCH$ do
    Sample a mini-batch from whole datasets.
    Feed samples into encoders get representations and cluster
      assignments.
    Generate pseudo-labels and compute pseudo-labels based
      soft neighbors loss according to Eq. (12).
    Compute supervised loss by Eq. (13).
    Update network weight to minimize $\mathcal{L}_{boost}$.
```

D. Complexity Analysis and Discussions

This section focuses on analyzing the time complexity per batch. For clarity, let the batch size be denoted as n. In the `Pre-train Stage`, the time complexity of representation learning is $\mathcal{O}(n^2)$. In the `Train Stage`, the complexity of obtaining the similarity between the sample and the nearest neighbor is $\mathcal{O}(n^2)$. Next, the complexity of calculating $\mathcal{L}_{local}$ and $\mathcal{L}_{glocal}$ is $\mathcal{O}(n^2)$. Similarly, the complexity of calculating $\mathcal{L}_{clu\_cont}$ is $\mathcal{O}(C^2)$. Since $C \ll n$ is generally true, the complexity of calculating $\mathcal{L}_{train}$ is $\mathcal{O}(n^2)$. In the `Boost Stage`, the time complexity of $\mathcal{L}_{boost}$ is $\mathcal{O}(n^2 + n\log n)$. $n\log n$ is for obtaining the most $\sigma\%$ confident samples in each class.

Finally, assume that the total number of iterations is T, the time complexity of the proposed method is $\mathcal{O}(n^2T)$. It should be noted that even if the time complexity in each batch is $\mathcal{O}(n^2)$, the batch size is always much smaller than the dataset size, so it does not cause excessive time complexity. Each time complexity is summarized in Table I.

TABLE I
THE TIME COMPLEXITY OF EACH STEPS DURING TRAINING MODEL

Stage	Term	Complexity
Pre-train Stage	Compute $\mathcal{L}_{pre-train}$	$\mathcal{O}(n^2)$
Train Stage	Obtain positive confidence S	$\mathcal{O}(n^2)$
	Compute $\mathcal{L}_{local}$	$\mathcal{O}(n^2)$
	Compute $\mathcal{L}_{global}$	$\mathcal{O}(n^2)$
	Compute $\mathcal{L}_{clu\_cont}$	$\mathcal{O}(C^2)$
	Compute $\mathcal{L}_{train}$	$\mathcal{O}(n^2)$
Boost Stage	Select most $\sigma\%$ confident samples	$\mathcal{O}(n\log n)$
	Compute $\mathcal{L}_{boost}$	$\mathcal{O}(n^2)$

On the other hand, this paper mainly regards the nearest neighbors as potential positive pairs, which improves the discriminability of representation learning. As an extension to contrastive learning, the proposed method can be well extended to different tasks, such as few-shot learning, zero-shot learning, anomaly detection, etc. Meanwhile, for mixed data types, proposed method can still serve as a good representation learning paradigm, providing important technical support without adding additional computational costs. For example, for structured data (such as age), the perception radius can be calculated by Euclidean distance; for set data such as diagnosis codes, Jaccard similarity can be used instead. This flexibility shows that the soft neighbor strategy has the potential to handle multi-modal data.

TABLE II
DETAILED INFORMATION ABOUT FIVE BENCHMARK DATASETS.

Datasets	Classes	Split	Samples
CIFAR-10	10	Train+Test	60000
CIFAR-100	20	Train+Test	60000
STL-10	10	Train+Test	13000
ImageNet-10	10	Train	13000
ImageNet-Dogs	15	Train	19500

IV. EXPERIMENTS

In this section, we perform experimental validation on our proposed method. We first discuss the experimental settings including implementation details, dataset descriptions, and evaluation metrics. Then we give a series of visualized experiments, comparison experiments and ablation studies to help better understand the methods.

A. Experimental Settings

1) Datasets & Metrics: In this paper, we assess our proposed method on five widely-used image datasets: CIFAR-10, CIFAR-100 [45], STL-10 [46], ImageNet-10, and ImageNet-Dogs [15]. For CIFAR-100, we took its 20 super-classes as the ground truth instead of 100 fine-grained classes. The detail of the datasets is summarized in Table II.

We use three widely-used clustering metrics, namely normalized mutual information (NMI) [47], clustering accuracy (ACC) [48] and adjusted Rand index (ARI) [49]. All metrics with higher values demonstrate better clustering results.

2) Network Architectures: For a fair comparison with previous works [50] [51], we adopt ResNet18 as encoder on CIFAR-10, CIFAR-100 and STL-10, and ResNet50 in ImageNet-10 and ImageNet-Dogs. As ResNet is designed for squared images with size 224×224, we follow previous works [23], [52] to modify the standard ResNet to help the backbones work in the small size images, such as CIFAR-10 and CIFAR-100. As for the predictor, we use a fully-connected layer with ReLU squeezing features to a low-dimensional space, whose dimension is equal to the number of clusters.

3) Image Processing: First of all, we normalize all datasets with different means and standard deviations (std). Following Cutout [53], the augmentation was randomly selected four transformations from RandAugment [54], whose parameters were uniformly sampled between fixed ranges. We list whole parameters in Table III.

TABLE III
AUGMENTATIONS INFORMATION FOR FIVE BENCHMARK DATASETS.

Datasets	Mean	Std
CIFAR-10	[0.491, 0.482, 0.447]	[0.202, 0.199, 0.201]
CIFAR-100	[0.507, 0.487, 0.441]	[0.268, 0.257, 0.276]
STL-10	[0.485, 0.456, 0.406]	[0.229, 0.224, 0.225]
ImageNet-10	[0.485, 0.456, 0.406]	[0.229, 0.224, 0.225]
ImageNet-Dogs	[0.485, 0.456, 0.406]	[0.229, 0.224, 0.225]
Transformation	Parameter	Range
Brightness	B	[0.05,0.95]
Color	C	[0.05,0.95]
Contrast	C	[0.05,0.95]
Posterize	B	[4,8]
Rotate	θ	[-30,30]
Sharpness	S	[0.05,0.95]
Shear X,Y	R	[-0.1,0.1
Solarize	T	[0,256]
Translation X, Y	λ	[-0.1,0.1]

4) Implementation Details: At the `Pre-train Stage`, we follow SCAN and NNM to train our encoder for each dataset respectively. At the `Train Stage`, we use Adam optimizer with momentum 1e-4 learning rate and 1e-4 weight decay. and train models for 500 epochs. We experimentally set the hyper-parameter for cross-entropy loss ϵ to be 5, and $\alpha_1 = \alpha_2 = 1$, respectively. The number of soft neighbors K is selected as 10. At the `Boost Stage`, we fixed the learning rate to 1e-4 and trained the model 200 epochs. we select the $\sigma = 70\%$ most confident samples, set threshold γ is set to 0.99 and the number of soft neighbors K is also selected as 10. Here, we also empirically set $\beta_1 = \beta_2 = 1$ for all datasets. All experiments are implemented on RTX 3090Ti with CUDA 11.0 and PyTorch 1.6.0 [55].

B. Visualization

1) Illustration for positiveness confidence of soft neighbors: In this sub-section, we directly illustrate the positiveness confidence of soft neighbors through experiments on ImageNet-10. As shown in Fig. 5, we randomly selected four samples and calculated the positiveness confidence of its top-10 neighbors. The most intuitive one is Airliner (the second row in the figure), whose local nearest neighbor is almost a flip of the images. Another is Airship (the third row in the figure). Even though the staff appears in the second picture, our model still regards it as the nearest neighbor of the current sample. It sidely reflects that our model can well extract important feature information from raw image.

2) The evolution of features learning and cluster assignment during training process: Next we experimentally illustrate the convergence of the model. It can be known that our proposed method is mainly divided into three major steps: `Pre-train Stage`, `Train Stage`, and `Boost Stage`. Therefore, after completing each stage, we perform T-SNE [56] to visualize obtained features after the encoder. At the same time, the clustering assignment is represented by different colors. As shown in Fig. 6-(a), after the `Pre-train Stage`, we can see the obscure distribution of each cluster. In other words, we get a rough neighborhood structure for downstream clustering learning. Then after `Train Stage`, we can clearly observe the clear distribution of features and the balance of cluster assignment in Fig. 6-(b). At the `Boost Stage`, as Fig. 6-(c) displays, we obtained more compact and well-separated clusters, which illustrates the contribution of a pseudo-labels based learning strategy.

C. Comparson Results

In the comparisons, we adopt both traditional methods, including K-Means [1], SC [57], AC [58], NMF [59], and

(a) CIFAR-10 (b) CIFAR-100 (c) STL-10 (d) ImageNet-10 (e) ImageNet-Dogs

Fig. 4. Visualization of some image samples from the five benchmark datasets.

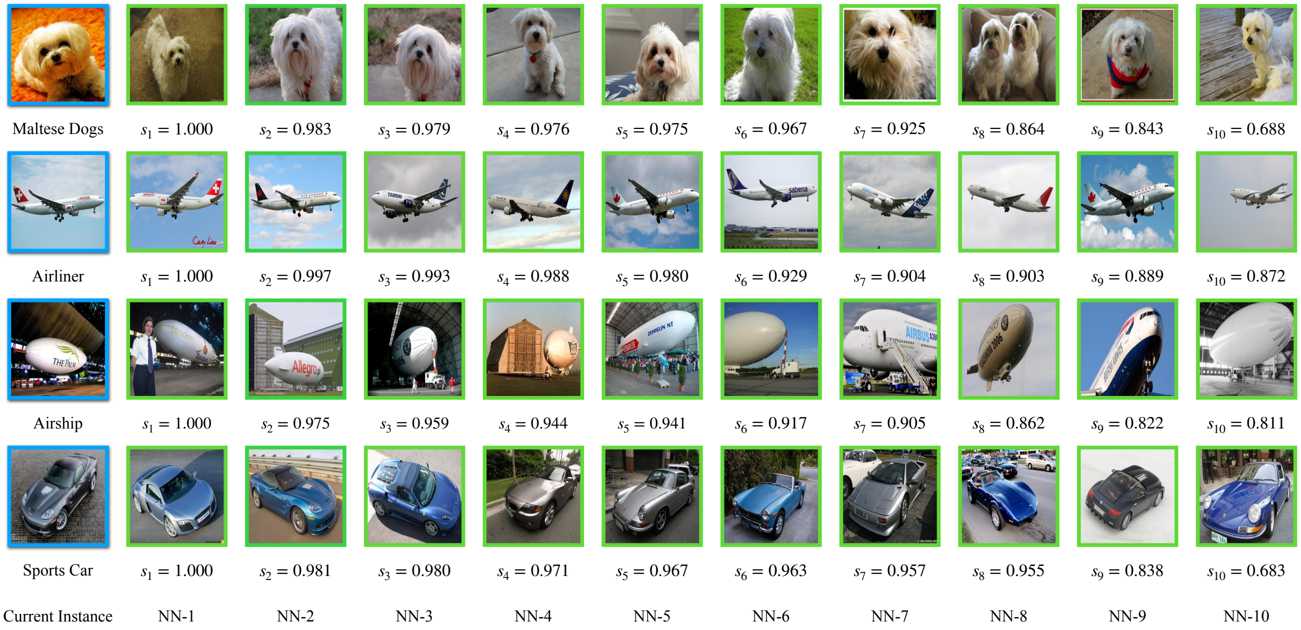
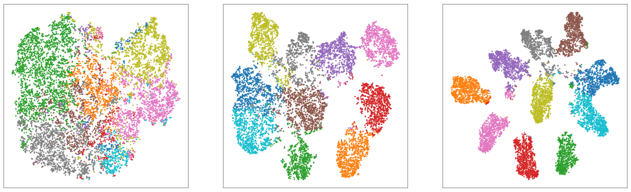


Fig. 5. Nearest neighbors (NN) of the randomly selected samples on ImageNet-10.



(a) Pre-train Stage (b) Train Stage (c) Boost Stage

Fig. 6. Feature space visualization on CIFAR-10. We apply T-SNE on features spaces and use different colors to indicate variant cluster assignments.

deep clustering methods, such as AE [60], DAE [61], DCGAN [62] DeCNN [63], VAE [64], JULE [14], DEC [19], DAC [15], ADC [65], DDC [37], DCCM [18], IIC [21], PICA [50], DCDC [66], CC [26], Pretext [32]+K-means, SCAN [23], NNM [52], DFVC [67], EDESC [68], DeepDPM [69], SPICE [38] and TCL [22].

We report the clustering performance of the above methods on five datasets in Table IV. It can be seen that our proposed method can achieve good performance in most cases. It is worth noting that in the `Train Stage`, our method has been able to match or even better than most comparisons. For example, on ImageNet-10, our method is already 1.0% higher than the second-best method SPICE in ACC. It can be seen that compared with the latest clustering method TCL, our method is still competitive and achieves better performance on CIFAR-10/100 and ImageNet-10. It is worth noting that CC [26] and TCL [22] use cluster-level contrastive learning to obtain the final clustering assignment, which have been widely used in many unsupervised tasks. Our proposed method can still obtain comparable results on different datasets, which shows that our method has potential application value in unsupervised learning. Finally, the results show that the proposed method achieves promising performance even on all datasets.

D. Ablation Study

In this section, several ablation experiments are proposed to verify the importance of each module. From top to bottom, soft neighbors are the core of this article, and we first analyze their important role in the entire training process including in `Train Stage` and `Boost Stage`.

1) Effectiveness of the modules in Train Stage: Recall the Section III-B, `Train Stage` mainly includes local and global level losses related to neighbors and a cluster-level loss.

TABLE IV
CLUSTERING PERFORMANCE ON FIVE BENCHMARK DATASETS. THE FIRST (GREEN) AND SECOND (BLUE) BEST RESULTS ARE HIGHLIGHTED.

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**
K-Means [1]	0.087	0.229	0.049	0.084	0.130	0.028	0.125	0.192	0.061	0.119	0.241	0.057	0.055	0.105	0.020
SC [57]	0.103	0.247	0.085	0.090	0.136	0.022	0.098	0.159	0.048	0.151	0.274	0.076	0.038	0.111	0.013
AC [58]	0.105	0.228	0.065	0.098	0.138	0.034	0.239	0.332	0.140	0.138	0.242	0.067	0.037	0.139	0.021
NMF [59]	0.081	0.190	0.034	0.079	0.118	0.026	0.096	0.180	0.046	0.132	0.230	0.065	0.044	0.118	0.016
AE [60]	0.239	0.314	0.169	0.100	0.165	0.048	0.250	0.303	0.161	0.210	0.317	0.152	0.104	0.185	0.073
DAE [61]	0.251	0.297	0.163	0.111	0.151	0.046	0.224	0.302	0.152	0.206	0.304	0.138	0.104	0.190	0.078
DCGAN [62]	0.265	0.315	0.176	0.120	0.151	0.045	0.210	0.298	0.139	0.225	0.346	0.157	0.121	0.174	0.078
DeCNN [63]	0.240	0.282	0.174	0.092	0.133	0.038	0.227	0.299	0.162	0.186	0.313	0.142	0.098	0.175	0.073
VAE [64]	0.245	0.291	0.167	0.108	0.152	0.040	0.200	0.282	0.146	0.193	0.334	0.168	0.107	0.179	0.079
JULE [14]	0.192	0.272	0.138	0.103	0.137	0.033	0.182	0.277	0.164	0.175	0.300	0.138	0.054	0.138	0.028
DEC [19]	0.257	0.301	0.161	0.136	0.185	0.050	0.276	0.359	0.186	0.282	0.381	0.203	0.122	0.195	0.079
DAC [15]	0.396	0.522	0.306	0.185	0.238	0.088	0.366	0.470	0.257	0.394	0.527	0.302	0.219	0.275	0.111
ADC [65]	-	0.325	-	-	0.160	-	-	0.530	-	-	-	-	-	-	-
DDC [37]	0.424	0.524	0.329	-	-	-	0.371	0.489	0.267	0.433	0.577	0.345	-	-	-
DCCM [18]	0.496	0.623	0.408	0.285	0.327	0.173	0.376	0.482	0.262	0.608	0.710	0.555	0.321	0.383	0.182
IIC [21]	-	0.617	-	-	0.257	-	-	0.610	-	-	-	-	-	-	-
PICA [50]	0.591	0.696	0.512	0.310	0.337	0.171	0.611	0.713	0.531	0.802	0.870	0.761	0.352	0.352	0.201
DCDC [66]	0.585	0.699	0.506	0.310	0.349	0.179	0.621	0.734	0.547	0.817	0.879	0.787	0.360	0.365	0.207
CC [26]	0.705	0.790	0.637	0.431	0.429	0.266	0.746	0.850	0.726	0.859	0.893	0.822	0.445	0.429	0.274
Pretext [32]+K-means	0.598	0.659	0.509	0.402	0.395	0.239	0.604	0.658	0.506	-	-	-	-	-	-
SCAN [23]	0.715	0.816	0.665	0.449	0.440	0.283	0.673	0.792	0.618	-	-	-	-	-	-
NNM [52]	0.748	0.843	0.709	0.484	0.477	0.316	0.694	0.808	0.650	-	-	-	-	-	-
DFVC [67]	0.643	0.756	0.615	0.435	0.472	0.261	0.643	0.731	0.598	0.753	0.847	0.736	0.375	0.391	0.184
EDESC [68]	0.464	0.627	-	0.370	0.385	-	0.687	0.745	-	-	-	-	-	-	-
DeepDPM [69]	-	-	-	-	-	-	0.740	0.840	0.700	-	-	-	-	-	-
SPICE [38]	0.734	0.838	0.705	0.448	0.468	0.294	**0.817**	**0.908**	**0.812**	0.812	0.921	0.836	-	-	-
TCL [22]	0.792	0.867	0.737	**0.522**	**0.517**	0.337	0.732	0.792	0.564	**0.869**	0.891	0.823	**0.624**	**0.639**	**0.503**
Train Stage	0.743	0.842	0.706	0.473	0.475	0.305	0.609	0.733	0.552	0.837	0.918	0.832	0.417	0.428	0.271
Boost Stage	**0.795**	**0.886**	**0.777**	0.502	0.508	**0.347**	0.612	0.736	0.559	0.855	**0.931**	**0.856**	0.445	0.463	0.308

TABLE V
ABLATION ANALYSIS AT TRAIN STAGE ON CIFAR-10, CIFAR-100 AND IMAGENET-10. ✓ DENOTES THE LOSS IN USE.

	Losses			CIFAR-10		CIFAR-100		ImageNet-10	
No.	**Local Level**	**Global Level**	**Cluster Level**	**NMI**	**ACC**	**NMI**	**ACC**	**NMI**	**ACC**
1	✓			0.722	0.827	0.449	0.443	0.829	0.908
2		✓		0.727	0.827	0.454	0.449	0.831	0.909
3			✓	0.713	0.816	0.452	0.420	0.826	0.905
4	✓	✓		0.731	0.831	0.462	0.456	0.835	0.913
5	✓	✓	✓	0.743	0.842	0.473	0.475	0.837	0.918

In order to verify their role in the training process, we conduct ablation studies by removing one or two of the losses and report the results in Table V. Here, we use the local level as the baseline to observe different losses function. Noting that in this section we mainly discuss the `Train Stage`, so we report clustering results without boost strategy for simplicity. It can be seen that both global level and cluster level loss have significantly improved performance since the former makes the model more accurate in finding soft neighbors and enhances representation learning, which can make the entire cluster distribution more distinct and compact.

2) Effectiveness of the modules in Boost Stage: In the following experiments, all ablations only affect at `Boost Stage`, thus we perform all experiments on the same model trained in the `Train Stage` for convenience. Similarly, we first perform an ablation analysis on the loss that appears in the `Boost Stage`. The results of the ablation are shown in Fig. 7. Among them, the red bar represents the result at `Train Stage`. It can be seen that only using the pseudo-label loss can slightly improve the clustering performance. On the contrary, when only using soft clustering loss based on pseudo-label, it has a side effect. One possible reason is that it destroys the distribution of clusters during training. The joint use of the two losses, the soft neighbor loss is more like a fine-tuned item, to correct the supervised learning based on the pseudo-label to achieve better performance.

3) Train Stage v.s. Boost Stage: To verify the effectiveness of `Boost Stage`, we provide an ablation analysis as follows. Specifically, we give two different experimental settings: 1) `Train Stage` (700 epochs) and 2) `Train Stage` (500 epochs) + `Boost Stage` (200 epochs). Meanwhile, the remain settings are the same as the main manuscripts. Fig. 8 shows the performance of different training strategies. For better readability, we fill in red before 500 epochs and blue afterwards.

As shown in Fig. 8, it can be observed that the performance tends to be stable after 500 epochs when using only the train stage (light-colored lines). However, after adding `Boost`

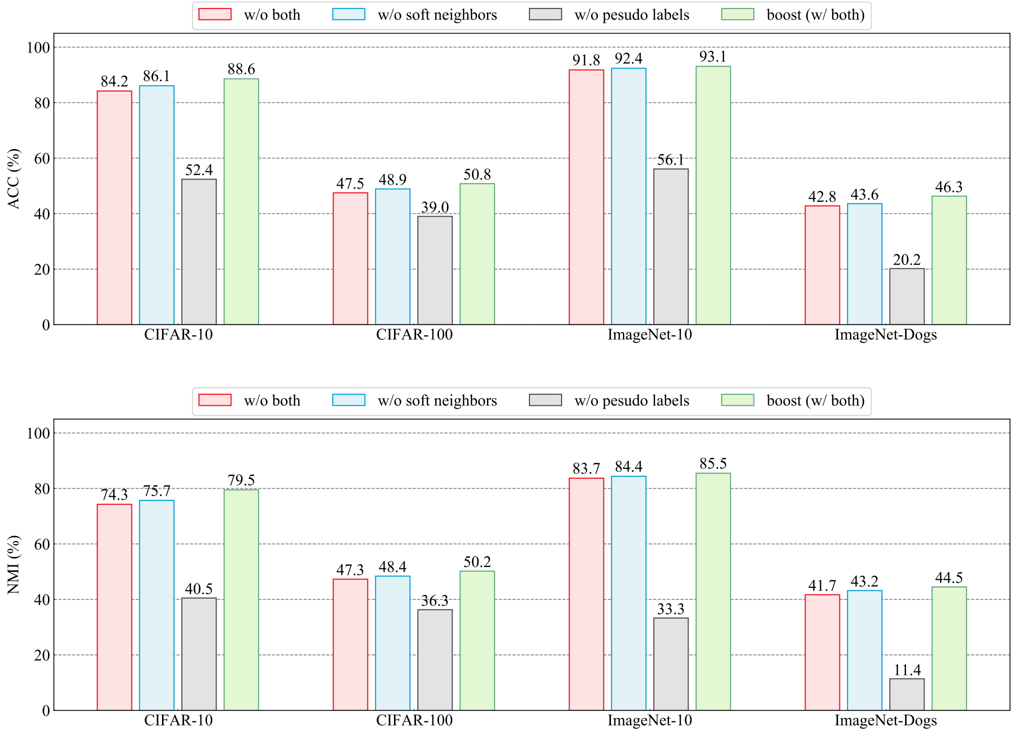
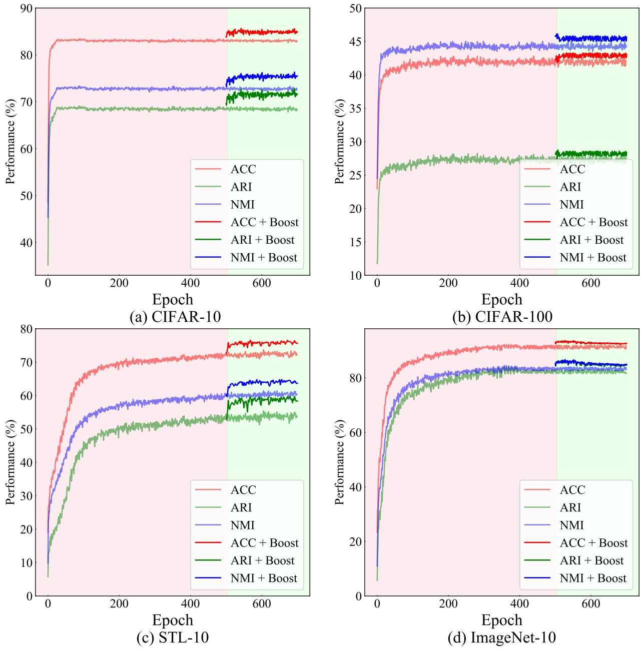


Fig. 7. Ablation analysis at Boost Stage on four datasets in ACC (top) and NMI (bottom). Models at Train Stage are adopted as baselines and boosted with different losses.



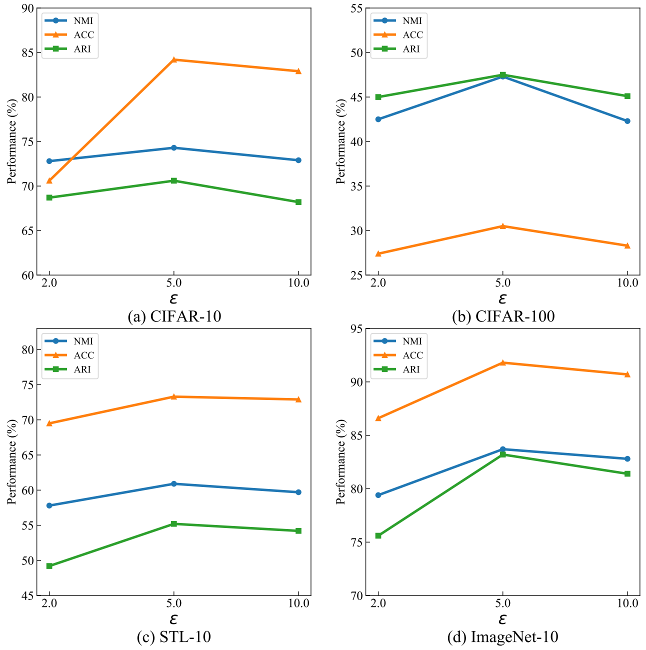
(a) CIFAR-10

(b) CIFAR-100

(c) STL-10

(d) ImageNet-10

Fig. 8. The effectiveness of Boost Stage on four datasets.



(a) CIFAR-10

(b) CIFAR-100

(c) STL-10

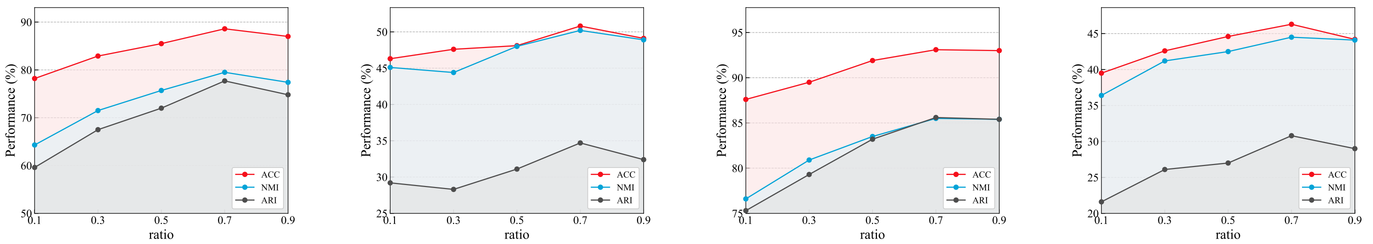
(d) ImageNet-10

Fig. 9. The impacts of ϵ at Train Stage on four datasets.

`Stage`, the performance will be significantly improved, especially on CIFAR-10 and STL-10. As a result, longer training does not make the performance better, which directly reflects the effectiveness of the `Boost Stage`.

E. Analysis on Parameter Sensitivity

In the above sections, we discuss the impact of different modules on method performances. Next, we will discuss how hyper-parameters effects the performances.



(a) CIFAR-10 (b) CIFAR-100 (c) ImageNet-10 (d) ImageNet-Dogs

Fig. 10. Clustering performance under different ratio σ of pseudo labels on four datasets.

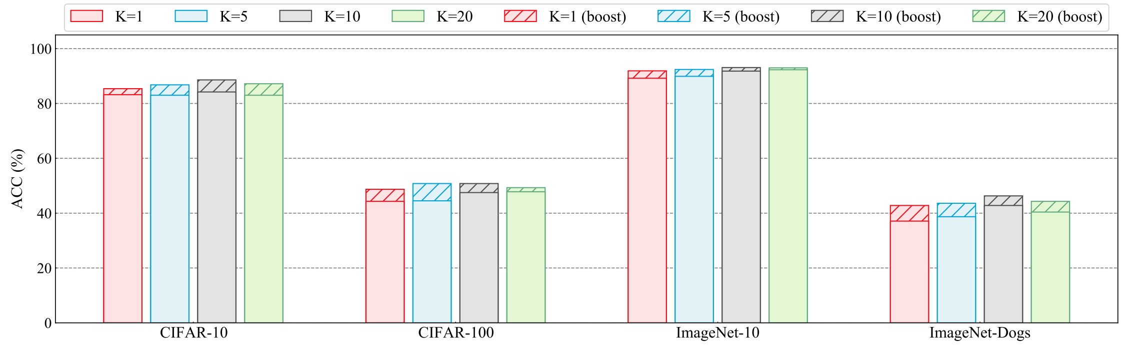


Fig. 11. Clustering performance (ACC) under various selected top-K neighbors in both Train and Boost stage.

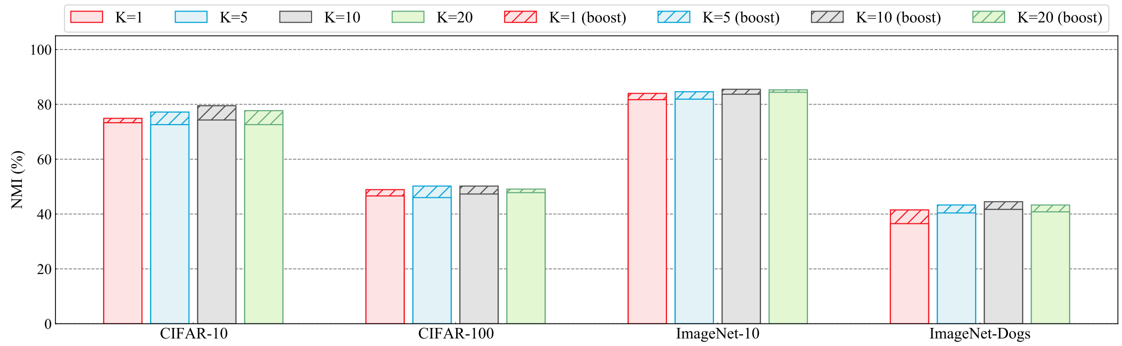


Fig. 12. Clustering performance (NMI) under various selected top-K neighbors in both Train and Boost stage.

1) Effectiveness of the entropy loss: In this subsection, we will discuss the how impact of entropy loss term on clustering performance. We experimentally find that the model obtains trivial solutions and all samples are predicted into one category without entropy loss. When other parameters are fixed, we set the ϵ in the range of $\mathcal{S}_\epsilon = \{2.0, 5.0, 10.0\}$, and conduct experiments on four different datasets. Fig. 9 summarizes the impact of different ϵ values on performance.

As shown in Fig. 9, our method achieves the optimal results when ϵ is set to 5. We believe that when the value of ϵ is too small, the method assigns all samples to the same cluster. On the contrary, when ϵ is excessively large, the entropy loss will result in uniformly distributed predictions. In other words, each sample tends to have an equal probability to be assigned to different categories, thereby degrading prediction accuracy. Therefore, in all experiments, we fixed $\epsilon = 5$ so that the model can get the best results.

2) The quality of pseudo-labels: The above experiment shows that the generation of pseudo-label is the most important part of the `Boost Stage`. For a pseudo-labels generation, the basic keys are threshold γ and the ratio σ selected from candidate pseudo-labels.

Here, we set γ to $\mathcal{S}_\gamma = \{0.9, 0.95, 0.99\}$ respectively to observe its impact on the clustering effect. It is worth mentioning that we do not set γ too low. The reason is that we find that after `Train Stage`, the prediction confidence of the cluster assignment is almost above 0.9 during the experiment. Table VI summarizes the experimental results under the different values of γ. We can see that when the threshold is higher, better results can be obtained. This also

TABLE VI
CLUSTERING PERFORMANCE UNDER DIFFERENT VALUE OF THRESHOLD γ ON FOUR DATASETS. **BOLDFACE** DENOTES THE BEST RESULTS.

Datasets	**CIFAR-10**			**CIFAR-100**		
threshold γ	**NMI**	**ACC**	**ARI**	**NMI**	**ACC**	**ARI**
0.90	0.781	0.876	0.761	0.489	0.492	0.330
0.95	0.788	0.879	0.764	0.499	0.501	0.342
0.99	**0.795**	**0.886**	**0.777**	**0.502**	**0.508**	**0.347**
Datasets	**ImageNet-10**			**ImageNet-Dogs**		
threshold γ	**NMI**	**ACC**	**ARI**	**NMI**	**ACC**	**ARI**
0.90	0.846	0.926	0.847	0.423	0.422	0.269
0.95	0.850	0.928	0.850	0.425	0.430	0.286
0.99	**0.855**	**0.931**	**0.856**	**0.445**	**0.463**	**0.308**

TABLE VII
THE PERFORMANCE ON DIFFERENT TRAINING STRATEGIES WITH SOFT NEIGHBORS (SN) ON IMAGENET-10 AND IMAGENET-DOGS.

		ImageNet-10		**ImageNet-Dogs**	
Stages	**Strategy**	**NMI**	**ACC**	**NMI**	**ACC**
Train Stage	SimCLR	0.821	0.898	0.401	0.410
	SimCLR+SN	0.833	0.903	0.404	0.416
	MoCo	0.826	0.905	0.408	0.421
	MoCo+SN	0.837	0.918	0.417	0.428
Boost Stage	SimCLR	0.839	0.918	0.410	0.429
	SimCLR+SN	0.848	0.936	0.419	0.452
	MoCo	0.844	0.924	0.432	0.436
	MoCo+SN	0.855	0.931	0.445	0.463

directly shows that a higher threshold can make the selected pseudo-label more representative and closer to the ground truth.

Now we discuss the choice of the number of pseudo-labels. We set the selected ratios to $\mathcal{S}_\sigma = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ to observe their impact on the model, and report the results in Fig. 10. There is a general trend from Fig. 10-(a) to Fig. 10-(d) that the clustering results increase and then decrease corresponding to the ratios, and the optimal ratio value is taken at 0.7. The reason for this is that if the number of pseudo-labels is too small, then in the process of supervised learning, the fine-tuning ability of the model is limited. Conversely, if the number of labels obtained is too large, wrong classification samples may be introduced, hurting the clustering performance.

3) Effectiveness of the soft neighbors strategy: To verify the importance of soft neighbors throughout the whole training process, we conduct ablation studies by changing the number of considered nearest neighbors, namely, the value of K. Fig. 11 and Fig. 12 shows the clustering performance of the algorithm under different K values. It can be seen that when $K = 1$, the clustering performance in the two steps is not as good as $K = 5$ and $K = 10$, which directly proves that the soft neighbor strategy can consider more effective positive samples and improve the model encoder's ability for representation learning, which in turn affects the final performance. However, we can observe that if we consider enough neighbors, that is, $K = 20$, the model cannot achieve the best results. It shows that limited positive samples can improve the performance well and too much soft neighbors can inevitably introduce negative samples, messing up the training process and hurting the clustering performance. Our results show that the best ACC and NMI can be obtained when $K = 10$, so we simply adopt $K = 10$ in all other experiments.

4) Generalization of soft neighbors strategy: Moreover, we tested the generalization of the proposed soft neighbors. SimCLR and MoCo are both well-known contrastive learning methods. It should be noted that both methods are directly used to pre-train the encoder. In the `Train` and `Boost Stage`, we mainly train linear-prob to cluster assignment. Therefore, we adopt the MoCo-v2 implementation on ImageNet-10 and ImageNet-Dogs in `Pre-train Stage`, and then apply the soft positive neighbors to see the linear-prob performance. In fairness, we performed 500 iterations in the `Train` stage and 200 iterations in the `Boost Stage` here. All other parameters remain the same as in the original SimCLR and MoCo-v2 papers. Table VII shows the performance under the different training strategies.

As shown in Table VII, we can see that under the different pre-training strategies, the models can obtain relatively good results. Among them, the performance of MoCo-v2 is slightly better than SimCLR. Since MoCo introduces more negative pairs through memory bank during training, which enables encoder to obtain better representation.

V. CONCLUSIONS

Based on the observation that contrastive learning will push away false negative samples during the training process, in this paper, we propose a contrastive clustering method based on a soft neighbor strategy. Different from previous works that use hard positive and negative samples, we introduce the concept of *perception radius* to measure the positiveness confidence of neighbors. According to these adaptive weights, we propose local and global level soft neighbor losses to partially support the current sample. At the same time, we also use cluster level loss to make the cluster distribution more separated. In addition, in order to further reduce the impact of false negative samples, we also propose a soft neighbor strategy based on pseudo-labels to fine-tune the network and improve clustering performance. Extensive experiments on image clustering demonstrate the effectiveness of our proposed method.

In the future, we plan to extend this method to open-world semi-supervised learning tasks. Meanwhile, beyond image learning tasks, we also intend to investigate more complex scenarios, such as multi-view clustering and heterogeneous data clustering.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101902, in part by the Natural Science Basic Research Program of Shaanxi (Program No. 2021JM-071), in part by the National Natural Science Foundation of China under Grants 62176203, Grant 62176212, Grant 61936014 and Grant 61772427.

REFERENCES

[1] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*, 1967, pp. 281–297.

[2] J. Xue, F. Nie, R. Wang, and X. Li, "Iteratively re-weighted algorithm for fuzzy c-means," *IEEE Transactions on Fuzzy Systems*, 2022.

[3] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[4] K. Wang, J. Zhang, D. Li, X. Zhang, and T. Guo, "Adaptive affinity propagation clustering," *arXiv preprint arXiv:0805.1096*, 2008.

[5] Q. Gu and J. Zhou, "Local learning regularized nonnegative matrix factorization," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[6] D. Wang, S. Han, Q. Wang, L. He, Y. Tian, and X. Gao, "Pseudo-label guided collective matrix factorization for multiview clustering," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 8681–8691, 2021.

[7] M. Ouyang, W. J. Welsh, and P. Georgopoulos, "Gaussian mixture clustering and imputation of microarray data," *Bioinformatics*, vol. 20, no. 6, pp. 917–923, 2004.

[8] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 977–986.

[9] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[10] D. Wu, Z. Yang, J. Lu, J. Xu, X. Xu, and F. Nie, "Ebmgc-gnf: Efficient balanced multi-view graph clustering via good neighbor fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[11] J. Lu, F. Nie, R. Wang, and X. Li, "Fast multiview clustering by optimal graph mining," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[12] H. Zhong, G. Loukides, and R. Gwadera, "Clustering datasets with demographics and diagnosis codes," *Journal of biomedical informatics*, vol. 102, p. 103360, 2020.

[13] H. Zhong, G. Loukides, and S. P. Pissis, "Clustering demographics and sequences of diagnosis codes," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2351–2359, 2022.

[14] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5147–5156.

[15] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5879–5887.

[16] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 132–149.

[17] E. Tzoreff, O. Kogan, and Y. Choukroun, "Deep discriminative latent space for clustering," *arXiv preprint arXiv:1805.10795*, 2018.

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

[19] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*. PMLR, 2016, pp. 478–487.

[20] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation." in *Ijcai*, 2017, pp. 1753–1759.

[21] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9865–9874.

[22] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng, "Twin contrastive learning for online clustering," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2205–2221, 2022.

[23] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Scan: Learning to classify images without labels," in *European conference on computer vision*. Springer, 2020, pp. 268–285.

[24] X. Wang, Z. Liu, and S. X. Yu, "Unsupervised feature learning by cross-level instance-group discrimination," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 586–12 595.

[25] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," *arXiv preprint arXiv:2005.04966*, 2020.

[26] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 8547–8555.

[27] Y. Tao, K. Takagi, and K. Nakata, "Clustering-friendly representation learning via instance discrimination and feature decorrelation," *arXiv preprint arXiv:2106.00131*, 2021.

[28] Z. Huang, J. Chen, J. Zhang, and H. Shan, "Learning representation for clustering via prototype scattering and positive sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[29] W. Xia, T. Wang, Q. Gao, M. Yang, and X. Gao, "Graph embedding contrastive multi-modal representation learning for clustering," *IEEE Transactions on Image Processing*, vol. 32, pp. 1170–1183, 2023.

[30] S. Zhang, S. Khan, Z. Shen, M. Naseer, G. Chen, and F. S. Khan, "Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3479–3488.

[31] X. Wen, B. Zhao, and X. Qi, "Parametric classification for generalized category discovery: A baseline study," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 16 590–16 600.

[32] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[33] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[34] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[35] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[36] X. Peng, J. Feng, J. Lu, W.-Y. Yau, and Z. Yi, "Cascade subspace clustering," in *Thirty-First AAAI conference on artificial intelligence*, 2017.

[37] J. Chang, Y. Guo, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep discriminative clustering analysis," *arXiv preprint arXiv:1905.01681*, 2019.

[38] C. Niu, H. Shan, and G. Wang, "Spice: Semantic pseudo-labeling for image clustering," *IEEE Transactions on Image Processing*, vol. 31, pp. 7264–7278, 2022.

[39] T. W. Tsai, C. Li, and J. Zhu, "Mice: Mixture of contrastive experts for unsupervised image clustering," in *International Conference on Learning Representations*, 2020.

[40] Y. Shen, Z. Shen, M. Wang, J. Qin, P. Torr, and L. Shao, "You never cluster alone," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 734–27 746, 2021.

[41] Y. Guo, M. Xu, J. Li, B. Ni, X. Zhu, Z. Sun, and Y. Xu, "Hcsc: Hierarchical contrastive selective coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9706–9715.

[42] J. Xu, H. Tang, Y. Ren, L. Peng, X. Zhu, and L. He, "Multi-level feature learning for contrastive multi-view clustering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 051–16 060.

[43] E. Pan and Z. Kang, "Multi-view contrastive graph clustering," *Advances in neural information processing systems*, vol. 34, pp. 2148–2159, 2021.

[44] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9588–9597.

[45] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[46] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[47] S. Wang, X. Liu, X. Zhu, P. Zhang, Y. Zhang, F. Gao, and E. Zhu, "Fast parameter-free multi-view subspace clustering with consensus anchor guidance," *IEEE Transactions on Image Processing*, vol. 31, pp. 556–568, 2021.

[48] R. Wang, J. Lu, Y. Lu, F. Nie, and X. Li, "Discrete and parameter-free multiple kernel k-means," *IEEE Transactions on Image Processing*, vol. 31, pp. 2796–2808, 2022.

[49] D. Wu, F. Nie, J. Lu, R. Wang, and X. Li, "Balanced graph cut with exponential inter-cluster compactness," *IEEE Transactions on Artificial Intelligence*, 2021.

[50] J. Huang, S. Gong, and X. Zhu, "Deep semantic clustering by partition confidence maximisation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8849–8858.

[51] D. Zhang, F. Nan, X. Wei, S. Li, H. Zhu, K. McKeown, R. Nallapati, A. Arnold, and B. Xiang, "Supporting clustering with contrastive learning," *arXiv preprint arXiv:2103.12953*, 2021.

[52] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang, "Nearest neighbor matching for deep clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 693–13 702.

[53] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[54] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[55] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[56] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[57] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," *Advances in neural information processing systems*, vol. 17, 2004.

[58] K. C. Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," *Pattern recognition*, vol. 10, no. 2, pp. 105–112, 1978.

[59] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Twenty-first international joint conference on artificial intelligence*, 2009.

[60] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, 2006.

[61] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[62] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[63] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2528–2535.

[64] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[65] P. Haeusser, J. Plapp, V. Golkov, E. Aljalbout, and D. Cremers, "Associative deep clustering: Training a classification network with no labels," in *German Conference on Pattern Recognition*. Springer, 2018, pp. 18–32.

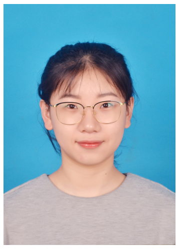
[66] Z. Dang, C. Deng, X. Yang, and H. Huang, "Doubly contrastive deep clustering," *arXiv preprint arXiv:2103.05484*, 2021.

[67] Q. Ji, Y. Sun, J. Gao, Y. Hu, and B. Yin, "A decoder-free variational deep embedding for unsupervised clustering," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[68] J. Cai, J. Fan, W. Guo, S. Wang, Y. Zhang, and Z. Zhang, "Efficient deep embedded subspace clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1–10.

[69] M. Ronen, S. E. Finder, and O. Freifeld, "Deepdpm: Deep clustering with an unknown number of clusters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9861–9870.

Yu Duan received the MS and Ph.D. degree from Northwestern Polytechnical University in 2021 and 2024. He is currently working as a Post Doctor with Xidian University, Xi'an. His research interests focus on clustering, semi-supervised learning, representation learning.



Huimin Chen received the B.S. degree in software engineering and the M.S. degree in computer technology from Northwestern Polytechnical University, Xi'an, China, in 2019 and 2022, respectively. She is currently working toward the Ph.D. degree in computer science with the School of Computer Science. Her research interests include machine learning and its applications, and her main subject is graph-based clustering methods.

Runxin Zhang received the B.S. degree in Software Engineering from Northwestern Polytechnical University, Xi'an, China. He is currently pursuing the Ph.D. degree with the School of School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University. His research interests include clustering and its applications.

Rong Wang received the B.S. degree in information engineering, the M.S. degree in signal and information processing, and the Ph.D. degree in computer science from Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004, 2007 and 2013, respectively. During 2007 and 2013, he also studied in the Department of Automation, Tsinghua University, Beijing, China for his Ph.D. degree. He is currently an associate professor at the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests focus on machine learning and its applications.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2025.3583194

Feiping Nie received the Ph.D. degree in Computer Science from Tsinghua University, China in 2009, and is currently a full professor in Northwestern Polytechnical University, China. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing and information retrieval. He has published more than 100 papers in the following journals and conferences: TPAMI, IJCV, TIP, TNNLS, TKDE, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, ACM MM. His papers have been cited more than 20000 times and the H-index is 84. He is now serving as Associate Editor or PC member for several prestigious journals and conferences in the related fields.

Xuelong Li (Fellow, IEEE) is currently a full professor at the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an P.R. China.