Improving Image Contrastive Clustering Through Self-Learning Pairwise Constraints

DOI:
10.1109/TNNLS.2023.3329494

Citation for published version (APA):
Guo, Y., Bai, L., Yang, X., & Liang, J. (2023). Improving Image Contrastive Clustering Through Self-Learning Pairwise Constraints. *IEEE Transactions on NEural Networks and Learning Systems*, 1-13. https://doi.org/10.1109/TNNLS.2023.3329494

Published in:
IEEE Transactions on NEural Networks and Learning Systems

Improving Image Contrastive Clustering through Self-Learning Pairwise Constraints

Yecheng Guo, Liang Bai, Xian Yang, Jiye Liang *Senior Member, IEEE*

***Abstract*—In this paper, a new unsupervised contrastive clustering model is introduced, namely, Image Contrastive Clustering with Self-Learning Pairwise Constraints (ICC-SPC). This model is designed to integrate pairwise constraints into the contrastive clustering process, enhancing the latent representation learning and improving clustering results for image data. The incorporation of pairwise constraints helps reduce the impact of false negatives and false positives in contrastive learning, while maintaining robust cluster discrimination. However, obtaining prior pairwise constraints from unlabeled data directly is quite challenging in unsupervised scenarios. To address this issue, ICC-SPC designs a pairwise learning module. This module autonomously learns pairwise constraints among data samples by leveraging consensus information between latent representation and pseudo labels, which are generated by the clustering algorithm. Consequently, there is no requirement for labeled images, offering a practical resolution to the challenge posed by the lack of sufficient supervised information in unsupervised clustering tasks. ICC-SPC's effectiveness is validated through evaluations on multiple benchmark datasets. This contribution is significant, as we present a novel framework for unsupervised clustering by integrating contrastive learning with self-learning pairwise constraints.**

***Index Terms*—Contrastive clustering, Self-learning pairwise constraints, Latent representation learning, Unsupervised clustering**

I. Introduction

CLUSTERING is an essential area of study within machine learning and data mining, which seeks to categorize unlabeled images into distinct groups such that images within the same group share high similarity, while objects in separate groups display dissimilarity. Various clustering algorithms [1], [2], [3], [4] have been proposed, many of which have been successfully applied in real-world applications. One primary challenge in clustering involves determining an effective similarity measure between objects. With the rise in data dimensionality, "the curse of dimensionality" [5] poses challenges, hindering the efficiency of many clustering algorithms when handling high-dimensional images [6].

To tackle this challenge, some clustering algorithms have been developed that simultaneously learn representations and clustering assignments. These algorithms can be classified into two groups: those that are not based on neural networks and those that utilize deep learning techniques. The former includes spectral clustering [7], subspace clustering [8], [9], and NMF clustering [2], which use traditional feature extraction methods to obtain low-dimensional representations. However, deep neural networks can learn better representations for complex datasets in certain contexts than traditional embedding methods[10]. Therefore, deep neural network based clustering algorithms, known as deep clustering, have been designed, such as DEC [11], DAC [12] and JULE [13]. These algorithms improve existing deep neural networks to obtain latent representations of images that are more suitable for clustering tasks.

Over the past few years, contrastive learning has emerged as a successful strategy for deep neural networks to learn unsupervised data representation [14], [15]. To simultaneously learn data representations and clustering results, the integration of contrastive learning and clustering algorithms, known as contrastive clustering, has become increasingly popular [16], [17], [10]. While contrastive learning treats the argumentations of an object as its positive samples and selects other objects as its negative samples, its performance is limited by the fact that the cluster structure is rarely considered. This makes the quality of the acquired latent representation sensitive to the selection of negatives, with false negatives often being selected for objects that belong to the same cluster, as observed in many cases [14], [15]. As a result, objects in the same clusters might not have comparable latent representations.

To tackle the aforementioned issue, contrastive clustering algorithms incorporate clustering assumptions into the training process of contrastive neural networks. The objective is to guarantee that similar data in the original space have comparable latent representations. However, while clustering assumptions can mitigate the impact of selecting false negatives, it also introduces a significant degree of false consistency uncertainty. In some contrastive clustering algorithms [17], [18], [19], an object's neighbors are regarded as its positives. However, since the data representation in the original feature space is often raw and unprocessed, an object and its neighbors may belong to different clusters, leading to the use of false positives. Therefore, it is evident that although integrating the clustering hypothesis can improve data representation consistency, cluster discrimination remains inadequate in contrastive clustering.

To improve the consistency of data representation and the discrimination of cluster structure in contrastive clustering, supplementary tags or pairwise constraints can be used to drive the training process. Pairwise constraints are a type of wide-used supervised information in semi-supervised clustering

Y. Guo, L. Bai and J. Liang are with Institute of Intelligent Information Processing, Shanxi University, Taiyuan, 030006, China (Corresponding author: Liang Bai)
Email: gyc15536823479@163.com, bailiang@sxu.edu.cn, ljy@sxu.edu.cn
X. Yang is with Alliance Manchester Business School, The University of Manchester, Manchester, M13 9PL, UK
Email: xian.yang@manchester.ac.uk

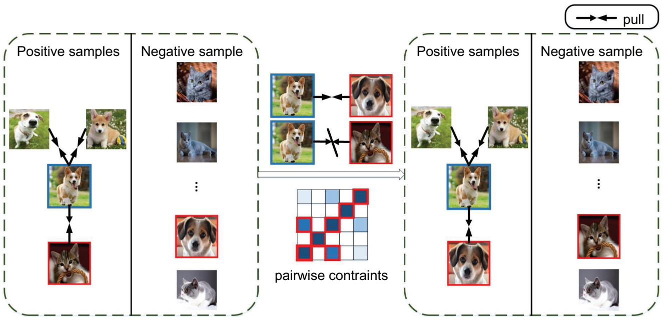


Fig. 1: The illustration of the effectiveness of incorporating pairwise constraints in contrastive clustering. The blue boxes indicate the anchor samples, whereas the red boxes illustrate the error samples (false negatives and positives). After adding pairwise constraints, samples of the same class can be better discriminated, which can alleviate the problems of false negatives and false positives.

[20], [21], [22]. These constraints can be expressed as pairwise similarities, dissimilarities, or partial orderings between data points. If some pairwise constraints are obtained, we can use them to reduce the effect of false positives and negatives in contrastive clustering. As shown in Fig. 1, incorporating pairwise constraints can help distinguish between samples from the same and different categories, which can alleviate the effects of false positives and negatives. By pulling samples in the same clusters closer together, the clustering performance can be further improved. Unfortunately, obtaining prior knowledge about pairwise constraints for an unlabeled dataset can be challenging for users.

Hence, in this paper, we introduce a novel approach, named **I**mage **C**ontrastive **C**lustering with **S**elf-learning **P**airwise **C**onstraints (**ICC-SPC**), which integrates a pairwise constraints learning module into the contrastive learning-based clustering framework. This approach is fully unsupervised, eliminating the need for labeled data, and thus reducing the cost and effort associated with acquiring such data. Our framework includes a pairwise constraints learning module, which comprises two parts: the first part learns pairwise constraints using consensus information among pseudo labels, latent representations, and their augmented samples, while the second part ensures that samples that are likely to belong to the same group have similar latent representations and pseudo labels using the learned pairwise constraints. Our approach leverages pairwise constraints to facilitate model training and alleviate challenges such as the presence of false negative samples in contrastive learning and the lack of adequate constraints on pseudo labels. The primary contributions of this paper include:

- A pairwise constraints learning module is developed, designed to learn pairwise constraints among data samples effectively. This innovation mitigates the inherent limitations of contrastive clustering by reducing the influence of both false negatives and false positives. The quality of the latent representations obtained is enhanced, while preserving robust cluster discrimination.
- The proposed pairwise constraints learning module can be integrated into any existing contrastive clustering model to enhance its performance by learning pairwise constraints among data samples.
- Our method is fully unsupervised and eliminates the need for labeled data, making it a viable and efficient solution for handling false negative samples in clustering.
- We have conducted extensive experiments on multiple benchmark datasets to demonstrate the efficacy of our proposed ICC-SPC method.

II. Related work

A. *Image Contrastive Learning*

Image Contrastive learning has recently made significant contributions in characterizing unlabeled data, as exemplified in [23]. The primary concept behind contrastive learning is to transform the input data into a latent space where the similarity of positive sample pairs is maximized, while the similarity of negative sample pairs is minimized [24]. As an example, SimCLR [14] introduces a straightforward contrastive learning framework for visual representation. It treats samples within a batch as negative samples and does not require any memory banks. Meanwhile, MOCO [15] utilizes a moving-averaged encoder and a queue. However, the false negatives problem can negatively affect the performance of contrastive learning in these methods.

To address this issue, several approaches have been proposed. FNC [25] presents an unsupervised method for identifying false negative samples, enhancing contrastive learning performance by removing such samples from the contrastive loss. IFND [26] proposes a novel self-supervised contrastive learning framework that leverages k-means to obtain pseudo labels and incrementally detects and eliminates false negative samples. SMoG [27] introduces the momentum grouping scheme, which synchronously conducts feature grouping with representation learning and reduces the false negatives of instance contrastive methods. PGCL [28] captures the core semantic structure inherent in graph data by grouping semantically analogous graphs together. This approach ensures consistency in clustering across varying augmentations of the same graph. The method proposed in [29] introduces a denoising supervision mechanism that emphasizes structured learning through supervised means. Furthermore, the approach outlined in [30] advocates for preserving mutual information between the representations and inputs, a strategy that effectively minimizes the semantic information loss associated with false negative samples.

In this paper, we address false positive and negative samples in contrastive learning by imposing constraints on sample pairs. Instead of relying solely on pseudo-labels to acquire class information and learn pairwise constraints, our method utilizes a combination of a pseudo-labeling matrix, latent representation similarity matrix, and pairwise constraints for augmented views during the training of the pairwise constraints network. This comprehensive approach yields superior pairwise constraints.

B. Image Contrastive Clustering

Image contrastive clustering is a technique that simultaneously learns data representations and clustering results, resulting in impressive clustering performance. Here, we first introduce instance-level contrastive clustering models. One such model is SCAN [19], which employs a two-stage learning strategy. The first stage leverages contrastive learning to extract features, whereas the second stage pulls in the semantic features of anchor points and nearest neighbors to perform clustering. Another model is IDFD [31], which uses instance discrimination to learn data similarities and feature decorrelation to eliminate superfluous correlations between features. Various end-to-end contrastive clustering methods have also been developed. MICE [32] integrates the discriminative representations acquired through contrastive learning with the semantic structures captured by a latent mixture model to create a unified probabilistic clustering framework. In SCL [33], negative samples are limited by semantic memory, allowing for the further distinction of samples from different clusters.

Lately, there has been growing attention on incorporating cluster-level contrastive learning to enhance the efficacy of contrastive clustering. One such approach is CC [10], which involves two levels of contrastive learning: instance-level and cluster-level, with the goal of enhancing the similarity of positive samples and reducing the similarity of negative samples for contrastive clustering. Another method is DRC [16], which explores the connection between contrastive learning and mutual information, and provides an approach for translating any maximized mutual information into minimal contrastive learning loss. GCC [17] introduces a clustering assumption that there should be similarity in representation and clustering assignments between an image and its randomly augmented nearest neighbors. TCL [34] uses the CC method for clustering, but it enhances both instance-level and cluster-level contrastive learning by utilizing a confidence-based criterion to select pseudo labels. TCC [35] broadens the scope of contrastive learning to encompass a cluster-level mechanism, where each data point in the same cluster contributes to a comprehensive representation conveying the contextual information of every data group.

However, the performance of all these contrastive learning-based clustering methods is limited by their ability to tackle the problem of false negative samples in contrastive learning.

C. Pairwise Constraints in Clustering

Semi-supervised clustering commonly integrates pairwise constraints to assimilate supervised data. For instance, NLPPC [20] leverages relationships between labels as constraints, enhancing label propagation optimization. AIPC [21] innovatively employs pairwise data points and neighboring constraints, known as constraint neighborhood projections, adept at resolving conflicts. The study in [36] presents a joint PCP model tailored for constrained spectral clustering, which simultaneously refines both propagation and affinity matrices. Meanwhile, [37] proposes a method that unifies diverse constraint sources to identify well-structured clusters. Extending the spectral clustering's objective function, Self-CSC [22] encompasses both pairwise and label self-constrained elements.

In contrast, our paper introduces a novel and fully unsupervised approach to alleviate the issue of false negative samples in contrastive learning-based clustering. Unlike existing approaches that rely on labeled data, our method offers an effective and efficient solution that does not require any supervised information. This represents a significant contribution to the field of clustering, as it offers a way to handle false negative samples without incurring the cost and effort of acquiring labeled data.

III. METHOD

This section provides a detailed introduction to our developed method, which is versatile and can be employed with various contrastive clustering techniques. As illustrated in Fig. 2, our ICC-SPC model is comprised of the contrastive clustering module and the pairwise constraints learning module. We will begin by introducing the contrastive clustering module, covering its underlying concepts and how it works. Then, we will introduce our pairwise constraints learning module, which allows us to learn and utilize pairwise constraints among data samples to enhance the clustering outcomes. We will provide a thorough explanation of how the pairwise constraints learning module works and how it can be integrated with the contrastive clustering module to achieve better clustering performance.

A. Preliminary

1) Definition of Notations: The subsequent notations are employed in this paper. Let $\boldsymbol{I} = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_M\}$ represent a set of the original images, $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_M\}$ and $\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{x}}_1, ..., \hat{\boldsymbol{x}}_M\}$ represent two random augmentations of the original images, where $\boldsymbol{x}_i$ (or $\hat{\boldsymbol{x}}_i$) indicates the ith row of $\boldsymbol{X}$ (or $\hat{\boldsymbol{X}}$) and M is sample size. $\boldsymbol{Z} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_M\}$ and $\hat{\boldsymbol{Z}} = \{\hat{\boldsymbol{z}}_1, ..., \hat{\boldsymbol{z}}_M\}$ are the corresponding latent representations of $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$, where $\boldsymbol{Z}$ and $\hat{\boldsymbol{Z}}$ are both matrices with dimensions $\mathbb{R}^{M \times D}$, D indicates the dimension of latent representations. $\boldsymbol{z}_i$ (or $\hat{\boldsymbol{z}}_i$) $\in \mathbb{R}^D$ represents the ith row of $\boldsymbol{Z}$ (or $\hat{\boldsymbol{Z}}$). We define $\boldsymbol{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_M\}$ and $\hat{\boldsymbol{Y}} = \{\hat{\boldsymbol{y}}_1, ..., \hat{\boldsymbol{y}}_M\}$ as the pseudo labels of $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ respectively, where $\boldsymbol{Y}$ (or $\hat{\boldsymbol{Y}}$) is $\mathbb{R}^{M \times K}$ with K being the number of clusters. $\boldsymbol{y}_i$(or $\hat{\boldsymbol{y}}_i$) $\in \mathbb{R}^K$ represent the ith row of $\boldsymbol{Y}$(or $\hat{\boldsymbol{Y}}$), indicating the cluster assignment of the i-th sample. $\boldsymbol{y}^i$(or $\hat{\boldsymbol{y}}^i$) $\in \mathbb{R}^M$ represent the ith column of $\boldsymbol{Y}$(or $\hat{\boldsymbol{Y}}$), indicating the probabilities of each of the M samples falling into cluster i.

In contrastive clustering, $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ need to be converted into an initial representation $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ by the neural network $f(\cdot)$. Furthermore, two neural networks $g_z(.)$ and $g_y(.)$ are utilized to convert $\boldsymbol{H}$ (or $\hat{\boldsymbol{H}}$) into $\boldsymbol{Z}$ (or $\hat{\boldsymbol{Z}}$) and $\boldsymbol{Y}$ (or $\hat{\boldsymbol{Y}}$), respectively. Before introducing our model in detail, we summarize the essential notations utilized throughout this paper in Table I.

2) Contrastive Clustering Module: This subsection provides an introduction to contrastive clustering, laying the foundation for our proposed model [10] [17]. The core idea behind contrastive learning is to acquire representations that group similar or positive samples together while separating dissimilar

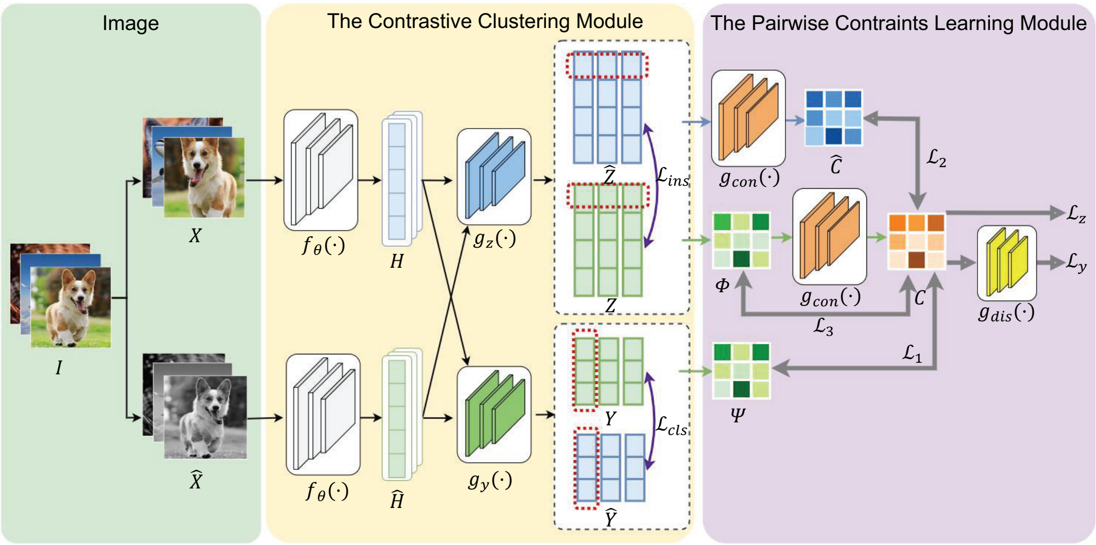


Fig. 2: The illustration of our ICC-SPC model. The first part of ICC-SPC is the contrastive clustering module, including the instance-level contrastive loss $\mathcal{L}_{ins}$ and the cluster-level contrastive loss $\mathcal{L}_{cls}$. The second part of ICC-SPC is the pairwise constraints learning module, using $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ together to train the network $g_{con}(\cdot)$ for obtaining pairwise constraints. The pairwise constraints are utilized to ensure that samples likely to belong to the same group have similar latent representations and pseudo labels, as indicated by the auxiliary losses $\mathcal{L}_z$ and $\mathcal{L}_y$. The $g_{dis}(\cdot)$ network is to determine samples with high-confidence pseudo labels.

TABLE I: Symbol and Description

Symbol	Description
$\boldsymbol{I}$	Set of original images.
$\boldsymbol{X}$	The first augmented view of the original images $\boldsymbol{I}$.
$\hat{\boldsymbol{X}}$	The second augmented view of the original images $\boldsymbol{I}$.
$\boldsymbol{H}$ ($\hat{\boldsymbol{H}}$)	The initial representation of $\boldsymbol{X}$ ($\hat{\boldsymbol{X}}$) generated by $f(.)$.
$\boldsymbol{Z}$ ($\hat{\boldsymbol{Z}}$)	The latent representation of $\boldsymbol{X}$ ($\hat{\boldsymbol{X}}$) generated by $g_z(.)$.
$\boldsymbol{z}_i(\hat{\boldsymbol{z}}_i)$	The ith row of $\boldsymbol{Z}$ ($\hat{\boldsymbol{Z}}$).
$\boldsymbol{Y}$ ($\hat{\boldsymbol{Y}}$)	The pseudo label of $\boldsymbol{X}$ ($\hat{\boldsymbol{X}}$) generated by $g_y(.)$.
$\boldsymbol{y}_i(\hat{\boldsymbol{y}}_i)$	The ith row of $\boldsymbol{Y}$ ($\hat{\boldsymbol{Y}}$).
$\boldsymbol{y}^i(\hat{\boldsymbol{y}}^i)$	The ith column of $\boldsymbol{Y}$ ($\hat{\boldsymbol{Y}}$).
$\overline{\boldsymbol{y}}_i$	The discretized form of $\boldsymbol{y}^i$.
$\boldsymbol{C}$ ($\hat{\boldsymbol{C}}$)	The pairwise constraints of $\boldsymbol{X}$ ($\hat{\boldsymbol{X}}$).
$c_{ij}(\hat{c}_{ij})$	The element in the ith row and jth column of $\boldsymbol{C}(\hat{\boldsymbol{C}})$.
$\overline{\boldsymbol{C}}$	The variant of $\boldsymbol{C}$ only containing high-confidence constraints.
$\overline{c}_{ij}$	The element in the ith row and jth column of $\overline{\boldsymbol{C}}$.
$\boldsymbol{\Psi}$ ($\hat{\boldsymbol{\Psi}}$)	The similarity matrix calculated from $\boldsymbol{Y}\boldsymbol{Y}^T$ ($\hat{\boldsymbol{Y}}\hat{\boldsymbol{Y}}^T$).
$\boldsymbol{\Phi}$ ($\hat{\boldsymbol{\Phi}}$)	The similarity matrix calculated from $\boldsymbol{Z}\boldsymbol{Z}^T$ ($\hat{\boldsymbol{Z}}\hat{\boldsymbol{Z}}^T$).
$\boldsymbol{\theta}$	The confidence levels of pseudo labels.
θ_i	The ith element of $\boldsymbol{\theta}$.
M	The sample size.
D	The dimension of latent representation.
K	The number of classes.
δ_1,δ_2	Hyperparameters used in the pairwise constraints learning.
$f(\cdot)$	The neural network.
$g_z(\cdot)$	The instance-level contrastive head.
$g_y(\cdot)$	The cluster-level contrastive head.
$g_{con}(\cdot)$	The pairwise constraints learning network.
$g_{dis}(\cdot)$	The discriminator to assist the genreation of $\boldsymbol{\theta}$.
τ_z	The instance-level temperature parameter.
τ_y	The cluster-level temperature parameter.

or negative samples from each other. This is achieved through both instance-level and cluster-level contrastive learning.

Instance-level contrastive learning aims to optimize a similarity metric between sample pairs. It encourages representations of similar samples to be more similar to each other than dissimilar samples. The objective is to decrease the distance between positive instances and increase the distance between negative instances. Conversely, cluster-level contrastive learning adopts a cluster-level projection head to learn the cluster assignments. It aims to group similar examples into clusters and subsequently learn representations that capture the similarities and differences between the clusters. Together, instance-level and cluster-level contrastive learning help to capture both within-class and between-class variations, enabling better representation learning for clustering tasks. The training objectives for contrastive clustering can be summarized as follows:

$$\mathcal{L} = \mathcal{L}_{ins} + \mathcal{L}_{cls}, \tag{1}$$

where $\mathcal{L}_{ins}$ refers to the instance-level contrastive loss and $\mathcal{L}_{cls}$ refers to the cluster-level contrastive loss.

To demonstrate the general applicability of our approach, we will evaluate it on two representative contrastive clustering methods, CC [10] and GCC [17], in our experiments. These methods differ slightly in their contrastive losses $\mathcal{L}$, mainly due to their different strategies for producing the second argumentation $\hat{\boldsymbol{X}}$. In CC, both $\hat{\boldsymbol{X}}$ and $\boldsymbol{X}$ are produced by random transformation of the original images. In GCC, $\boldsymbol{X}$ is made by random transformation of the original images and each $\boldsymbol{x}_i$ in $\hat{\boldsymbol{X}}$ is a randomly selected neighbor of $\boldsymbol{x}_i$. Next, we will provide detailed descriptions of $\mathcal{L}_{ins}$ and $\mathcal{L}_{cls}$ in CC and GCC.

For instance-level contrastive learning, the latent representations are used to compute the instance-level loss:

$$\mathcal{L}_{ins} = -\frac{1}{M} log \sum_{i=1}^{M} \frac{\sum_{\boldsymbol{z}_j \in \boldsymbol{P}_i} e^{(s(\boldsymbol{z}_i,\boldsymbol{z}_j)/\tau_z)}}{\sum_{\boldsymbol{z}_j \in \boldsymbol{P}_i \cup \boldsymbol{N}_i} e^{(s(\boldsymbol{z}_i,\boldsymbol{z}_j)/\tau_z)}}, \tag{2}$$

where $s(\cdot)$ is the similarity measure, can be cosine similarity or dot product, τ_z is the temperature parameter at the instance level, $\boldsymbol{P}_i$ and $\boldsymbol{N}_i$ are two sets including all the positive and negative examples of image $\boldsymbol{z}_i$, respectively. In CC, $\boldsymbol{P}_i = \{\hat{\boldsymbol{z}}_i\}$ and $\boldsymbol{N}_i$ includes other M random augmented samples. However, in GCC, the positive instances refer to the neighbors of $\boldsymbol{z}_i$ and the negative samples are defined as the non-neighbors of $\boldsymbol{z}_i$, i.e., $\boldsymbol{P}_i = \{\boldsymbol{z}_j | \boldsymbol{z}_j \in knn(\boldsymbol{z}_i)\}$ and $\boldsymbol{N}_i = \{\boldsymbol{z}_j | \boldsymbol{z}_j \notin knn(\boldsymbol{z}_i)\}$, where $knn(\boldsymbol{z}_i)$ is a set including the first k nearest neighbors of $\boldsymbol{z}_i$.

To implement cluster-level contrastive learning, we adopt pseudo labels and define the cluster-level contrastive loss as follows:

$$\mathcal{L}_{cls} = -\frac{1}{K}\sum_{i=1}^{K} log \frac{e^{(s(\boldsymbol{y}^i,\hat{\boldsymbol{y}}^i)/\tau_y)}}{\sum_{j=1}^{K} e^{(s(\boldsymbol{y}^i,\hat{\boldsymbol{y}}^j)/\tau_y)}} + \mathcal{L}_{au}, \qquad (3)$$

Where τ_y is the cluster-level temperature parameter. Additionally, $\mathcal{L}_{au}$ is an auxiliary loss to prevent the model from assigning most instances to the same cluster. Its definition can be seen in [10], [17].

B. Pairwise Constraints Learning Module

Compared to conventional contrastive clustering models, our framework includes a pairwise constraints learning module that learns pairwise constraints among data samples. As illustrated in Fig. 3, we assert that pairwise constraints should adhere to two foundational principles. Firstly, consistency between the pairwise constraints and the similarity matrices of pseudo-labels and latent representations should exist. Secondly, the pairwise constraints derived from various augmentation views should be consistent.

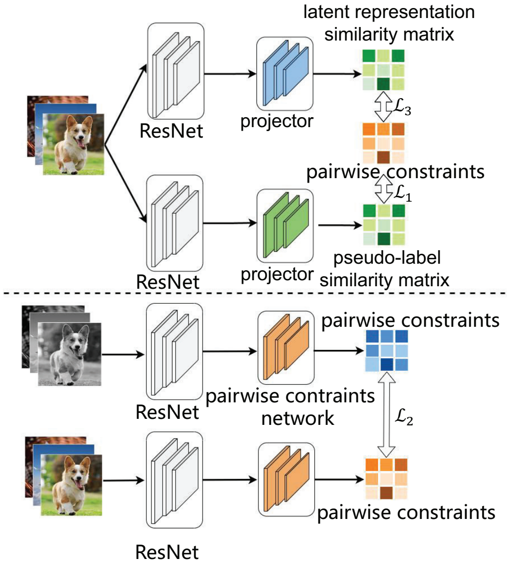


Fig. 3: Illustration of the pairwise constraints learning module within our framework, highlighting the adherence to two core principles: consistency between pairwise constraints and similarity matrices of pseudo-labels and latent representations, and consistency of pairwise constraints across various augmentation views.

This module comprises two parts. The first part utilizes consensus information among pseudo labels, latent representations, and their augmented samples to learn pairwise constraints. During the training phase, only the parameters of the network $g_{con}(\cdot)$ are modified, while the other network parameters remain fixed. The second part of our approach leverages the learned pairwise constraints to guarantee that samples which are likely to belong to the same group have comparable latent representations and pseudo labels. These pairwise constraints are applied at both the instance and class levels to facilitate model training and alleviate challenges such as the presence of false negative samples in contrastive learning and the lack of adequate constraints on pseudo labels. In the following parts, we delve into the details of this module.

1) Learning pairwise constraints of samples:

The first phase of our pairwise constraints learning module is focused on training a pairwise constraints network and extracting pairwise constraints. Let $\boldsymbol{\Phi}$ represent the pairwise similarity matrix derived from latent representations, with each element in the ith row and jth column given by $\phi_{ij} = s(\boldsymbol{z_i}, \boldsymbol{z_j})$. Moreover, we define $\boldsymbol{\Psi} = \boldsymbol{Y}\boldsymbol{Y}^T$, which is the similarity matrix calculated using pseudo labels $\boldsymbol{Y}$. The dimensions of these matrices are $\mathbb{R}^{M \times M}$. For the similarity measure, we choose cosine similarity in the subsequent text, that is:

$$s(\boldsymbol{z}_i, \boldsymbol{z}_j) = \frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_j}{\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|}. \qquad (4)$$

In this module, the $g_{con}(\cdot)$ network is specifically designed to learn pairwise constraints in a self-supervised manner. The network's final layer incorporates a Sigmoid layer to ensure the output is bounded between 0 and 1. The output of $g_{con}(\cdot)$, represented as $\boldsymbol{C} = g_{con}(\boldsymbol{\Phi})$ and $\hat{\boldsymbol{C}} = g_{con}(\hat{\boldsymbol{\Phi}})$, serves as the pairwise constraints. It should be noted that only the parameters of $g_{con}(\cdot)$ are updated during training, while the parameters of $f(\cdot)$, $g_z(\cdot)$, and $g_y(\cdot)$, which are used in the conventional contrastive clustering model, are fixed. We define three loss functions to train $g_{con}(\cdot)$. The first loss aims to ensure that $\boldsymbol{C}$ captures the information contained in the pseudo labels, which is defined as:

$$\mathcal{L}_1 = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}(c_{ij} - \psi_{ij})^2, \qquad (5)$$

where c_{ij} and ψ_{ij} denote the elements in the ith row and jth column of $\boldsymbol{C}$ and $\boldsymbol{\Psi}$, respectively.

In addition to the aforementioned loss function, we introduce a second loss that enforces consistency between pairwise constraints derived from two augmented views, represented as:

$$\mathcal{L}_2 = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}(c_{ij} - \hat{c}_{ij})^2, \qquad (6)$$

where $\hat{c}_{ij}$ denote the elements in the ith row and jth column of $\hat{\boldsymbol{C}}$.

In conjunction with the previous two losses, we define a third loss to ensure that $\boldsymbol{C}$ reflects the similarities observed in the latent space represented by $\boldsymbol{Z}$, as defined by:

$$\mathcal{L}_3 = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}(1-c_{ij})\log(1+\max\{\phi_{ij},0\}). \quad (7)$$

According to ROUL [38], the intent behind this loss is to ensure that when two samples show high similarity in the cosine similarity matrix of their latent representations, the corresponding value in $\boldsymbol{C}$ should also be high. This loss is crafted to ensure that higher values in $\boldsymbol{\Phi}$ match up with larger values in $\boldsymbol{C}$. However, due to the inherent inaccuracies in $\boldsymbol{\Phi}$, values that are negative or slightly above zero in $\boldsymbol{\Phi}$ do not significantly correspond to a rise in $\boldsymbol{C}$. The complete loss function utilized to train $g_{con}(\cdot)$ is defined as follows:

$$\mathcal{L}_{g_{con}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3. \quad (8)$$

2) Enhancing contrastive clustering using learned pairwise constraints:

In the second phase of our pairwise constraints learning module, pairwise constraints are used to encourage samples of the same class to have similar latent representations and pseudo labels. Specifically, after obtaining the pairwise constraints from the first part of our pairwise constraints learning module, we can use these constraints to identify samples of the same category. That is, samples with pairwise constraints larger than a given threshold δ_1 should be considered as samples of the same category. We define the following matrix $\overline{\boldsymbol{C}} \in \mathbb{R}^{M\times M}$, which is a parse version of $\overline{\boldsymbol{C}}$ and the entry in the ith row and jth column is determined by:

$$\bar{c}_{ij} = \begin{cases} c_{ij} & c_{ij} \geq \delta_1 \\ 0 & c_{ij} < \delta_1 \end{cases} \quad (9)$$

To enforce samples from potentially the same group, as indicated by $\overline{\boldsymbol{C}}$, having similar latent representations from augmented views, we define the following loss:

$$\mathcal{L}_z = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\bar{c}_{ij}e^{-s(\boldsymbol{z}_i,\hat{\boldsymbol{z}}_j)}. \quad (10)$$

Additionally, we introduce a similar loss to constrain the pseudo labels as follows:

$$\mathcal{L}_y = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\theta_i\bar{c}_{ij}e^{-s(\boldsymbol{y}_i,\hat{\boldsymbol{y}}_j)}, \quad (11)$$

where θ_i is an indicator that represents the level of confidence in the pseudo label assigned to sample i. The introduction of θ_i is intended to address the potential bias in the pseudo labels, which are not ground-truth labels and may contain misleading information. By determining the contributions of individual pseudo labels based on their level of confidence, our approach can mitigate the impact of such biases and improve the overall accuracy and reliability of the clustering results.

High-confidence pseudo label selection (HCPL): To automatically learn θ_i, we proposed a high-confidence pseudo label selection approach. We design a discriminator $g_{dis}(\cdot)$ to generate θ_i for each sample $\boldsymbol{x}_i$. The inputs to $g_{dis}(\cdot)$ are the concatenation of latent representations and pseudo labels. The output of $g_{dis}(\cdot)$ is to determine whether the input pair comes from the same sample. The positive examples are created by concatenating the latent representations $\boldsymbol{z}_i$ of $\boldsymbol{x}_i$ and the one-hot codes of the pseudo label $\overline{\boldsymbol{y}}_i$. Instead of directly using $\boldsymbol{y}_i$, we first discretize it into $\overline{\boldsymbol{y}}_i$, where the element of $\overline{\boldsymbol{y}}_i$ is set to 1 if the corresponding element of $\boldsymbol{y}_i$ is within a small margin of 1, typically less than 0.01, and 0 otherwise. In contrast, the negative examples combine the latent representations of $\boldsymbol{x}_i$ and the other K-1 one-hot codes. These instances are input into the discriminator $g_{dis}(\cdot)$ during training. When $\boldsymbol{z}_i$ and $\boldsymbol{y}_i$ belong to the same sample $\boldsymbol{x}_i$, the aim is to maximize the output from the discriminator $g_{dis}(\cdot)$, and conversely, minimize it when they do not belong to the same sample.

With the other parts of the model frozen, the discriminator $g_{dis}(\cdot)$ trained using the following loss:

$$\mathcal{L}_{dis} = \frac{1}{MK}\sum_{i=1}^{M}(\log g_{dis}(\boldsymbol{z}_i \oplus \overline{\boldsymbol{y}}_i) + \sum_{j=1}^{K-1}\log(1-g_{dis}(\boldsymbol{z}_i \oplus \overline{\boldsymbol{y}}_j))), \quad (12)$$

where $\overline{\boldsymbol{y}}_j$ is a one-hot vector where ones appear at different locations from $\overline{\boldsymbol{y}}_i$ representing negative samples, and $\oplus$ is the vector concatenation operator. The output of $g_{dis}(\cdot)$ is then transformed into the discrete value θ_i to indicate the confidence level of pseudo labels:

$$\theta_i = \mathbb{I}(g_{dis}(\boldsymbol{z}_i \oplus \overline{\boldsymbol{y}}_i) > \delta_2), \quad (13)$$

where the indicator $\mathbb{I}$ equals 1 if the condition is satisfied and 0 otherwise. The reason for introducing a discriminator is that directly using the value of pseudo labels to indicate label confidence may lead to the exclusion of some challenging samples. These hard samples have a significant impact on the model performance, and we aim to improve contrastive learning by identifying reliable hard samples. The discriminator is helpful in this regard since it can identify trustworthy hard samples, even when their pseudo labels values are not very high.

C. Model Training Process

The model undergoes training through the following two phases. During the first phase, which is the pre-training phase, the contrastive clustering module's networks $f(\cdot)$, $g_z(\cdot)$, and $g_y(\cdot)$ are trained using the loss functions $\mathcal{L}_{ins}$ and $\mathcal{L}_{cls}$. In the second phase, the pairwise constraints generation network $g_{con}(\cdot)$ is first trained independently using the loss function $\mathcal{L}_{g_{con}}$ while keeping $f(\cdot)$, $g_z(\cdot)$, and $g_y(\cdot)$ fixed. Next, the network $g_{dis}(\cdot)$, which returns the confidence levels of pseudo labels, is trained using the loss function $\mathcal{L}_{dis}$ while fixing the other networks. Finally, the entire contrastive clustering model with pairwise constraints is trained, where the networks $f(\cdot)$, $g_z(\cdot)$, and $g_y(\cdot)$ are updated using the following loss function:

$$\mathcal{L} = \mathcal{L}_{ins} + \mathcal{L}_{cls} + \mathcal{L}_z + \mathcal{L}_y. \quad (14)$$

The details are shown in Algorithm 1.

IV. Experiments

This section presents the experimental results that illustrate the effectiveness of our proposed method. To illustrate the adaptability of our model, we evaluated its performance by

Algorithm 1 The Algorithm of ICC-SPC.

```
Input: Data samples $\boldsymbol{I}$; the number of pre-training epochs $E_1$; the number of training epochs $E_2$; sample size $M$; temperature hyperparameters $\tau_z$ and $\tau_y$; number of classes $K$; thresholds $\delta_1$, $\delta_2$; networks $f(\cdot)$, $g_z(\cdot)$, $g_y(\cdot)$, $g_{dis}(\cdot)$, and $g_{con}(\cdot)$.
// Phase 1: pre-training the contrastive clustering module
for epoch = 1 to $E_1$ do
    Compute the augmented views of latent representations and pseudo labels;
    Compute instance-level contrastive loss as defined in Eq.(2);
    Compute cluster-level contrastive loss as defined in Eq.(3);
    Update parameters of $f(\cdot), g_z(\cdot), g_y(\cdot)$ using the loss functions $\mathcal{L}_{ins}$ and $\mathcal{L}_{cls}$;
end
// Phase 2: Training the pairwise constraints learning module
for epoch = $E_1$ to $E_2$ do
    // Learning pairwise constraints
    Train $g_{con}(\cdot)$ using the loss defined in Eq. (8), while the parameters of $f(\cdot)$,$g_z(\cdot)$,$g_y(\cdot)$ are frozen;
    Generate pairwise constraints using $g_{con}(\cdot)$;
    // Utilizing pairwise constraints to enhance contrastive learning.
    Train $g_{dis}(\cdot)$ using the loss defined in Eq.(12), while fixing parameters of other networks in the model;
    Use $g_{dis}(\cdot)$ to get the confidence level of pseudo labels through Eq. (13);
    Compute the losses defined in Eq. (10) and Eq. (11) for constraining latent representations and pseudo labels, respectively.
    Train $f(\cdot)$, $g_z(\cdot)$, and $g_y(\cdot)$ using the loss defined in Eq. (14).
end
```

integrating the proposed self-learning pairwise constraints into two widely used contrastive clustering methods, namely CC and GCC.

A. Dataset

We performed experiments on five popular benchmark datasets: CIFAR-10 [39], CIFAR-100 [39], STL-10 [40], ImageNet-10 [12], and ImageNet-Dogs [12]. CIFAR-10 comprises of 60,000 32x32 colour images of 10 distinct categories, while CIFAR-100 comprises the same number of images but grouped into 20 broad classes and 100 sub-classes. For the sake of fairness in our experiment, we selected the 20 broad classes as our labeling scheme. STL-10 is an image classification dataset that also includes unlabeled data. ImageNet-10 and ImageNet-Dogs are subsets of the large-scale ImageNet dataset [41], containing 13,000 and 19,500 images, respectively. ImageNet-10 consists of 10 classes, while ImageNet-Dogs comprises 15 classes of dog breeds. We selected these datasets due to their diversity in size, complexity, and number of classes, which enabled us to comprehensively assess the efficacy of our proposed method.

To ensure a fair comparison, we employed the identical dataset settings as those used in the original versions of CC and GCC, respectively. This involved using the same training-test data splits, preprocessing steps, and hyperparameters. An overview of each dataset is presented in Table II. The models named ICC-SPC and IGCC-SPC indicate that our proposed self-learning pairwise constraints have been integrated into CC and GCC, respectively.

TABLE II: List of datasets used in our experiments.

Dataset	Image size(ICC-SPC)	Image size(IGCC-SPC)	#Samples	#Classes
CIFAR-10	224×224	32×32	60000	10
CIFAR-100	224×224	32×32	60000	20
STL-10	224×224	96×96	113000	10
ImageNet-10	224×224	96×96	13000	10
ImageNet-Dogs	224×224	96×96	19500	15

B. Baseline Methods

The baseline methods can be divided into three categories as follows:

- Conventional clustering models: Traditional clustering methods are techniques to partition objects or samples within a dataset into different groups or clusters. Representative methods include K-means [42], Agglomerative clustering(AC) [3], Spectral clustering (SC) [43], and Nonnegative Matrix Factorization (NMF) based clustering [2].
- Deep clustering models: Deep clustering methods are a category of techniques that combine deep learning with traditional clustering approaches. They leverage neural network models to learn data representations and apply these representations to clustering tasks. Representative methods include VAE [44], JULE [13], DAC [12], PICA [45] and DCCM [46].
- Contrastive clustering models: Contrastive clustering is a machine learning technique that combines elements of contrastive learning and clustering. Representative methods include DRC [16], NNCC [47], MICE [32], CC [10] and GCC [17].

C. Experimental Settings

Experimental setup. We implemented all experiments using PyTorch [48]. Our code is available at the following URL[1]. To ensure a fair comparison, we followed the data augmentation techniques proposed in [10] and [17]. We utilized the Adam optimizer with an initial learning rate of 0.0003 to optimize the MLP $g_{con}(\cdot)$ and the discriminator $g_{dis}(\cdot)$. The middle hidden layer of $g_{con}(\cdot)$ has a dimension of 4096. $g_{dis}(\cdot)$ has an output dimension of 1. The fully connected neural network $g_z(\cdot)$ is set to have a row space dimensionality of 128 to retain more information on images. Meanwhile, the

[1] https://github.com/gyc126/ICC-SPC

output dimension of $g_y(\cdot)$ is equal to the number of clusters. The training process consists of a total of $E_2 = 1000$ epochs.

In ICC-SPC, we utilized ResNet-34 [49] as the backbone for our experiments. The batch size was set to 256, and the temperature parameter τ_z was fixed at 0.5 for all experiments. For all datasets, we used a cluster-level temperature parameter τ_y of 1.0. The Adam optimizer with an initial learning rate of 0.0003 was utilized to simultaneously optimize $g_z(\cdot)$ and $g_y(\cdot)$. The overall model was trained after pre-training $E_1 = 200$ epochs. In IGCC-SPC, ResNet-18 [49] is used as the backbone for the experiments. The batch size used for training is 256 on CIFAR-10 and CIFAR-100, and 96 on STL-10 and ImageNet-10/Dogs. The temperature parameter τ_z is set to 0.1 for all datasets, and the cluster-level temperature parameter τ_y is fixed to 1.0. The optimizer used is SGD with an initial learning rate of 0.4, a weight decay of 0.0001, and a momentum coefficient of 0.9. The learning rate is decayed using a cosine scheduler with a decay rate of 0.1. The overall model is trained after pre-training $E_1 = 80$ epochs. We will discuss the settings of hyperparameters δ_1 and δ_2 in the sensitivity analysis, which will be presented later in this section. Additionally, the experiments are conducted on Nvidia A100. Training on CIFAR-10 takes approximately 25 hours, while ICC-SPC, GCC, and IGCC-SPC require 40 hours, 16 hours, and 27 hours, respectively.

Evaluation Metrics. To quantitatively evaluate the effectiveness of our proposed ICC-SPC and IGCC-SPC methods, we adopted three commonly used evaluation metrics for clustering, namely Accuracy (ACC) [11], Normalized Mutual Information (NMI) [50], and Adjusted Rand Index (ARI) [51]. ACC is a widely recognized evaluation metric in deep clustering. It measures the percentage of correctly assigned data points to their true clusters. NMI is a popular metric for measuring the quality of clusters obtained through deep clustering. It assesses the degree of agreement between the true cluster assignments and the predicted ones while accounting for chance. ARI is another widely used metric in deep clustering. It quantifies the similarity between the true and predicted cluster assignments, correcting for chance agreement. ARI values range from -1 to 1. The three metrics provide a comprehensive assessment of the clustering performance in terms of both accuracy and structure. A greater value of the metrics implies superior clustering performance.

D. Analysis of Clustering Performance

1) Baseline Comparison: Table III presents the evaluation results of several representative deep clustering methods, with results of other methods taken directly from their corresponding papers and publicly available code, and "-" denoting missing results. Our proposed deep clustering method outperforms conventional clustering methods across all five image datasets due to its ability to leverage the capabilities of deep neural networks in learning highly representative feature embeddings. These embeddings capture the intrinsic characteristics of the images, leading to enhanced clustering performance. Our contrastive clustering-based model outperforms other deep clustering methods on the same five datasets, which can be attributed to the efficacy of contrastive learning in producing high-quality latent representations of the data.

To show the effectiveness of our model, we compared it with other contrastive clustering models. Our comparison revealed that models with the inclusion of our self-learning pairwise constraints, ICC-SPC and IGCC-SPC, showed better performance than their original versions, CC and GCC, in most datasets. Additionally, IGCC-SPC showed the best performance in four out of five datasets, further validating the effectiveness of our approach. We observed that for the STL-10 dataset, ICC-SPC did not improve the performance of CC. This could be due to the fact that the STL-10 dataset only uses a small fraction of the whole dataset for testing, where cluster-level contrastive learning may not play an essential role, and pseudo labels are not generated to facilitate pairwise constraints learning. Overall, our proposed method outperforms conventional clustering techniques and other deep contrastive clustering models. Our self-learning pairwise constraints improve clustering accuracy, making it state-of-the-art in this area.

2) Clustering performance with self-labeling boost: Recently, TCL [34] adopted a confidence-based criterion to choose pseudo labels, enhancing instance-level and cluster-level contrastive learning. Specifically, in TCL, once the contrastive clustering model training is completed, a confidence-based boosting strategy is applied to fine-tune the model. This involves selecting confident samples by setting a threshold on the pseudo label obtained from the cluster-level contrastive head. The cross-entropy loss is used to fine-tune the model by aligning the pseudo labels of substantial data augmentation with the weak data augmentation of confident samples, a process known as self-labeling. Here, we aim to compare our model with this self-labeling boosted baseline model. For a fair comparison, we also fine-tune the IGCC-SPC model by adding the same number of epochs of the self-labeling strategy according to SCAN [19]. The comparison results are shown in Table IV, where IGCC-SPC* indicates that our IGCC-SPC model has been further fine-tuned exclusively using the self-labeling boosting strategy from TCL. We find that our IGCC-SPC* model generally shows better performance than TCL in 4 out of 5 datasets. As discussed earlier, the suboptimal performance of our model on the STL-10 dataset can be attributed to the fact that it only utilizes a small fraction of the entire dataset for testing. In this case, cluster-level contrastive learning may not be as crucial, and pseudo labels are not generated to aid in the learning of pairwise constraints.

E. Impact on Sample Selection in Contrastive Learning

1) Evaluation of Learned Pairwise Constraints: To illustrate the performance of learning pairwise constraints, we first represent the ground-truth pairwise relationships between samples using the one-hot encoding of the ground-truth labels denoted by $\boldsymbol{L}$. The matrix $\widetilde{\boldsymbol{L}} = \boldsymbol{L}\boldsymbol{L}^T$ is then calculated, where each element $\widetilde{l}_{ij}$ in the resulting matrix shows the ground-truth pairwise relationship between sample i and j. To compare the ground-truth pairwise relationships with the learned pairwise constraints from the self-learning approach,

TABLE III: Clustering performance of comparative methods on five benchmark datasets.

Dataset	CIFAR-10			CIFAR-100			STL-10			ImageNet-10			ImageNet-Dogs		
Methods	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
K-means	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
AC	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
SC	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
NMF	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
VAE	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	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
DAC	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
PICA	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
DCCM	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
DRC	0.621	0.727	0.547	0.356	0.367	0.208	0.644	0.747	0.569	0.830	0.884	0.798	0.384	0.389	0.230
NNCC	0.737	0.819	-	0.421	0.438	-	0.616	0.725	-	0.683	0.751	-	0.372	0.401	-
MICE	0.737	0.835	0.698	0.436	0.440	0.280	0.635	0.752	0.575	-	-	-	0.423	0.438	0.286
CC	0.705	0.790	0.637	0.431	0.429	0.266	**0.719**	**0.817**	**0.726**	**0.859**	0.893	0.822	0.445	0.429	0.274
GCC	0.764	0.856	0.728	0.472	0.472	0.305	0.684	0.788	0.631	0.842	0.901	0.822	0.490	0.526	0.362
ICC-SPC	0.732	0.832	0.697	0.433	0.457	0.299	0.705	0.806	0.645	0.843	**0.908**	0.825	0.498	0.490	0.363
IGCC-SPC	**0.784**	**0.870**	**0.752**	**0.493**	**0.474**	**0.322**	0.695	0.777	0.641	**0.892**	**0.952**	**0.898**	**0.608**	**0.641**	**0.504**

TABLE IV: Comparison with the boosting baseline model.

Dataset	Methods	NMI	ACC	ARI
CIFAR-10	TCL	0.819	0.887	0.780
	IGCC-SPC*	**0.850**	**0.915**	**0.833**
CIFAR-100	TCL	0.529	**0.531**	0.357
	IGCC-SPC*	**0.534**	0.507	**0.359**
STL-10	TCL	**0.799**	**0.868**	**0.757**
	IGCC-SPC*	0.725	0.796	0.675
ImageNet-10	TCL	**0.875**	0.895	0.837
	IGCC-SPC*	0.871	**0.911**	**0.846**
ImageNet-Dogs	TCL	0.623	0.623	**0.516**
	IGCC-SPC*	**0.624**	**0.683**	0.514

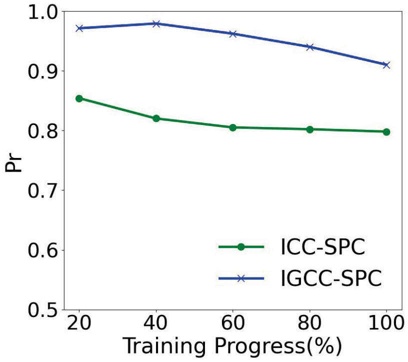


Fig. 4: The performance of learning pairwise constraints during training in Cifar-10 on ICC-SPC and IGCC-SPC methods.

a matrix $\widetilde{\boldsymbol{C}}$ is created by setting the non-zero elements of the learned pairwise constraints (denoted by $\overline{\boldsymbol{C}}$) to 1. The element $\widetilde{c}_{ij}$ in $\widetilde{\boldsymbol{C}}$ represents the pairwise constraints between the ith and jth samples. To evaluate the performance of the learned pairwise constraints, we calculate the proportion of correctly identified pairwise relationships between samples in the matrix $\widetilde{\boldsymbol{C}}$. This provides a measure of how well the learned pairwise constraints capture the ground-truth pairwise relationships. The proportion is calculated as:

$$Pr = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M}\widetilde{c}_{ij}\widetilde{l}_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{M}\widetilde{c}_{ij}}, \tag{15}$$

where M is the sample size. From Fig. 4, we can see that the performance of learning pairwise constraints remains consistently high for both ICC-SPC and IGCC-SPC models during the training process.

2) Analysis of False Positive and Negative Sample Rates: In this subsection, we further quantify our ability to accurately select positive and negative samples in contrastive learning. Same as above, we also use the ground-truth labels in the dataset for assistance in measurement. We calculate whether the positive and negative images selected for each target sample have the same label as the target. False positive samples are those selected positive samples whose labels are different from our target sample; false negative samples are those selected negative samples whose labels are the same as our target sample. The error rate for each target sample in the contrastive learning process is computed by determining the proportion of combined false positive and false negative samples. This error rate summed across all target samples is denoted as P_e:

$$P_e = \sum_{i=1}^{M}\frac{FP_i + FN_i}{PN_i}, \tag{16}$$

where FP_i and FN_i are the counts of false positive and false negative samples for the ith image, respectively. PN_i represents all candidate positive and negative samples of the ith image. A smaller P_e value indicates better ability of sample selection in the contrastive learning process. Fig. 5 shows the changes of P_e during the training process for both GCC and IGCC-SPC methods. Notably, the GCC method displays a persistently elevated P_e with minor fluctuations throughout its training. Conversely, the IGCC-SPC method starts with a smaller P_e value, which further decreases as training progresses. This observation demonstrates the superiority of our proposed method in selecting positive and negative samples.

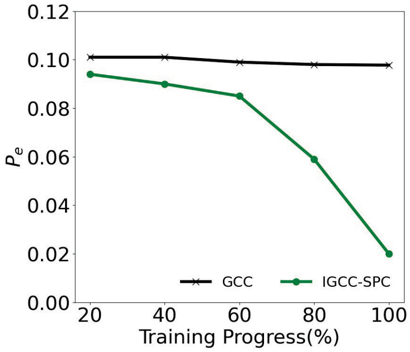


Fig. 5: Evolution of P_e throughout training for both GCC and IGCC-SPC methods.

F. Ablation Studies

In this subsection, we perform a series of ablation studies to demonstrate the effectiveness of the pairwise constraints learning module. As Table III indicates, the introduction of pairwise constraints leads to IGCC-SPC exhibiting the best performance. We conduct these ablation studies on IGCC-SPC using three datasets for illustrative purposes.

1) Impact of Pairwise Constraints on Clustering: Pairwise constraints play a crucial role as supervisory information for contrastive clustering. As demonstrated in Table V, the introduction of pairwise constraints in the contrastive clustering method leads to significantly improved cluster performance compared to the method without pairwise constraints. This improvement further highlights the effectiveness of these constraints in enhancing the performance of contrastive clustering.

TABLE V: Effectiveness of introducing self-learning pairwise constraints on clustering performance.

Dataset	Pairwise Constaints	NMI	ACC	ARI
CIFAR-10		0.764	0.856	0.728
	✓	**0.784**	**0.870**	**0.752**
CIFAR-100		0.472	0.472	0.305
	✓	**0.493**	**0.474**	**0.322**
ImageNet-dogs		0.490	0.526	0.362
	✓	**0.608**	**0.641**	**0.504**

2) Performance Across Different Loss Functions: We employ the loss function $\mathcal{L}_{g_{con}}$ to learn the network $g_{con}(\cdot)$, which is specifically designed to learn pairwise constraints in a self-supervised manner. The compulsory loss in $\mathcal{L}_{g_{con}}$ is $\mathcal{L}_1$, which ensures that pairwise constraints capture the information contained in the pseudo labels. In addition, $\mathcal{L}_{g_{con}}$ includes other loss functions to enforce consistency between pairwise constraints derived from two augmented views and ensure that pairwise constraints reflect the similarities observed in the latent space. To evaluate the significance of incorporating these additional losses, we conduct ablation experiments as presented in Table VI. The results show that the model trained with the complete loss $\mathcal{L}_{g_{con}}$ outperforms the model trained with only $\mathcal{L}_1$. This finding highlights the importance of incorporating all losses in learning pairwise constraints through self-supervision.

TABLE VI: Results of using different loss.

Dataset	Loss	NMI	ACC	ARI
CIFAR-10	$\mathcal{L}_1$	0.780	0.867	0.745
	$\mathcal{L}_{g_{con}}$	**0.784**	**0.870**	**0.752**
CIFAR-100	$\mathcal{L}_1$	0.478	0.465	0.309
	$\mathcal{L}_{g_{con}}$	**0.493**	**0.474**	**0.322**
ImageNet-Dogs	$\mathcal{L}_1$	0.590	0.623	0.483
	$\mathcal{L}_{g_{con}}$	**0.608**	**0.641**	**0.504**

3) Contribution of the HCPL Model to Learning: The results presented in Table VII highlight the effectiveness of incorporating HCPL into the learning of pairwise constraints. Our approach effectively addresses potential biases in the pseudo labels by determining their contributions based on their level of confidence. As a result, the clustering performance is improved, which suggests that confidence sample selection can be beneficial in utilizing pairwise constraints to enhance the overall clustering performance.

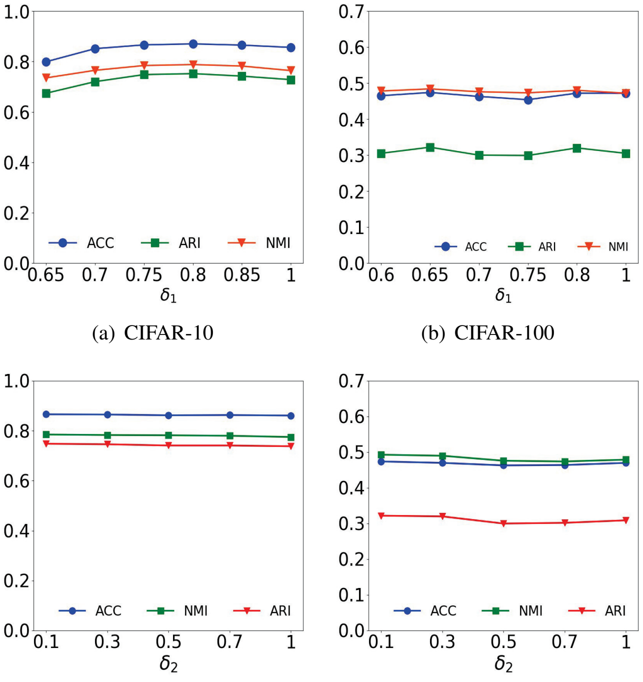
TABLE VII: Effectiveness of HCPL model for pairwise constraints learning.

Dataset	HCPL	NMI	ACC	ARI
CIFAR-10		0.780	0.868	0.748
	✓	**0.784**	**0.870**	**0.752**
CIFAR-100		0.481	0.469	0.317
	✓	**0.493**	**0.474**	**0.322**
ImageNet-Dogs		0.605	0.635	0.497
	✓	**0.608**	**0.641**	**0.504**

G. Sensitivity Analysis

Similar to the ablation study section, we perform sensitivity analyses on IGCC-SPC to investigate its sensitivity to hyperparameters.

1) Sensitivity analysis of the threshold δ_1: In our approach, the threshold value δ_1 is used in the process of obtaining $\overline{\boldsymbol{C}}$ as defined in Eq. 9. By setting a threshold value, we determine whether the elements in the matrix $\boldsymbol{C}$ should be set to 0. To examine the impact of the hyperparameter δ_1, we carry out experiments on CIFAR-10 and CIFAR-100 datasets. We assessed the effectiveness of our approach using different values of δ_1 and the results are presented in Fig. 6a) and b). To overcome the problem of difficulty in obtaining a value due to the sensitivity of the δ_1, we have implemented an automatic threshold selection mechanism in our ICC-SPC (or IGCC-SPC) method. Specifically, when we begin using pairwise supervised contrastive learning, if, at that point, the number of samples identified as belonging to the same class using a threshold exceeds three times (or one time) the total number of samples in the dataset, we designate that threshold as our selected threshold range. Following the automatic threshold selection method, the obtained threshold is 0.8. We analyze values around 0.8, including $\{0.65, 0.7, 0.75, 0.8, 0.85, 1\}$. The optimal thresholds for achieving the best performance vary between the two datasets. We observe that a threshold value of 0.8 yields near-optimal performance on the CIFAR-10 dataset, as the ACC, ARI, and NMI values are maximized. On the other hand, for the CIFAR-100 dataset, a threshold value of 0.65 results in performance close to the best.



(a) CIFAR-10

(b) CIFAR-100

(c) CIFAR-10

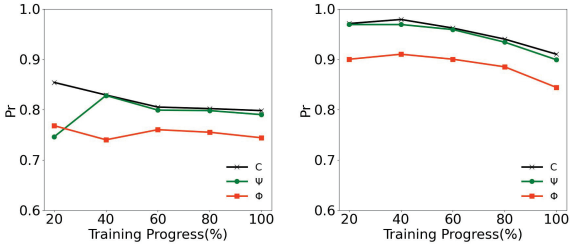
(d) CIFAR-100

Fig. 6: The impact of δ_1 and δ_2 on the performance of our proposed method.

2) Sensitivity analysis of the threshold δ_2: In the HCPL model, we utilize the output score of the discriminator to determine whether a sample is a high-confidence sample or not, using a threshold value δ_2. To investigate the effect of this hyperparameter on the model's performance, we conduct a sensitivity analysis by varying the value of δ_2 and evaluating its effect on CIFAR-10 and CIFAR-100 datasets, as illustrated in Fig. 6c) and d). Based on the results, we select 0.1 to conduct experiments in the above subsections. The sensitivity analysis reveals that varying the value of δ_2 does not have a significant impact on the performance of the IGCC-SPC model. This implies that the model is not sensitive to the choice of the threshold δ_2, and the performance is relatively stable across different values of δ_2.

H. Comparison with Alternative Constraints

Finally, to demonstrate the superiority of the pairwise constraint $\boldsymbol{C}$ proposed in this paper, we compare it with two other contraints, $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$. As depicted in Fig. 7, the resulting values of Pr for the Cifar-10 dataset using both ICC-SPC and IGCC-SPC methods with the three different constraints are shown. Our proposed pairwise constraints $\boldsymbol{C}$ consistently yields higher Pr than $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$, thus substantiating the superiority of $\boldsymbol{C}$.



(a) ICC-SPC

(b) IGCC-SPC

Fig. 7: Comparison of different pairwise constraints.

V. Conclusion

In this paper, we introduced a novel unsupervised contrastive clustering method that integrates a pairwise constraints learning module into the contrastive learning-based clustering framework. Our proposed approach enhances the quality of the learned latent representations by reducing the impact of false negatives and false positives while maintaining high cluster discrimination. Moreover, this fully unsupervised approach eliminates the need for labeled data, providing a practical solution to the challenge of insufficient supervised information in unsupervised clustering.To demonstrate the general applicability of our proposed pairwise constraints learning module, we integrated it into two widely-used contrastive clustering methods, CC and GCC, and evaluated its performance. The extensive experiments on various benchmark datasets demonstrated that our approach outperforms existing unsupervised clustering algorithms.

Our pairwise constraints approach has shown promising results with image data. Moving forward, we aim to adapt this method for diverse data types, notably text. In adapting to textual data, we will leverage its inherent properties and incorporate semantic relationships in our pairwise constraint calculations.

Acknowledgment

This work is supported by National Key Research and Development Program of China (No. 2021ZD0113303), the National Natural Science Foundation of China (Nos. 62022052, 62276159).

References

[1] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433–439, 1999.

[2] 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.

[3] 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.

[4] J. Wu, X. Xie, L. Nie, Z. Lin, and H. Zha, "Unified graph and low-rank tensor learning for multi-view clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 6388–6395.

[5] N. Altman and M. Krzywinski, "The curse (s) of dimensionality," *Nat Methods*, vol. 15, no. 6, pp. 399–400, 2018.

[6] I. Assent, "Clustering high dimensional data," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 340–350, 2012.

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

[8] 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.

[9] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 171–184, 2012.

[10] 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.

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

[12] 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.

[13] 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 (CVPR)*, June 2016.

[14] 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.

[15] 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 (CVPR)*, June 2020.

[16] H. Zhong, C. Chen, Z. Jin, and X.-S. Hua, "Deep robust clustering by contrastive learning," *arXiv preprint arXiv:2008.03030*, 2020.

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

[18] 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.

[19] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Scan: Learning to classify images without labels," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*. Springer, 2020, pp. 268–285.

[20] L. Bai, J. Wang, J. Liang, and H. Du, "New label propagation algorithm with pairwise constraints," *Pattern Recognition*, vol. 106, p. 107411, 2020.

[21] H. Wang, T. Li, T. Li, and Y. Yang, "Constraint neighborhood projections for semi-supervised clustering," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 636–643, 2014.

[22] L. Bai, J. Liang, and Y. Zhao, "Self-constrained spectral clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[23] 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.

[24] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[25] T. Huynh, S. Kornblith, M. R. Walter, M. Maire, and M. Khademi, "Boosting contrastive self-supervised learning with false negative cancellation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 2785–2795.

[26] T.-S. Chen, W.-C. Hung, H.-Y. Tseng, S.-Y. Chien, and M.-H. Yang, "Incremental false negative detection for contrastive learning," *arXiv preprint arXiv:2106.03719*, 2021.

[27] B. Pang, Y. Zhang, Y. Li, J. Cai, and C. Lu, "Unsupervised visual representation learning by synchronous momentum grouping," in *European Conference on Computer Vision*. Springer, 2022, pp. 265–282.

[28] S. Lin, C. Liu, P. Zhou, Z.-Y. Hu, S. Wang, R. Zhao, Y. Zheng, L. Lin, E. Xing, and X. Liang, "Prototypical graph contrastive learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[29] H. Zhao, X. Yang, C. Deng, and D. Tao, "Unsupervised structure-adaptive graph contrastive learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2023.

[30] Z. Zhao, R. Wang, Z. Wang, F. Nie, and X. Li, "Graph joint representation clustering via penalized graph contrastive learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2023.

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

[32] T. W. Tsai, C. Li, and J. Zhu, "Mice: Mixture of contrastive experts for unsupervised image clustering," in *International conference on learning representations*, 2021.

[33] J. Huang and S. Gong, "Deep clustering by semantic contrastive learning," *arXiv preprint arXiv:2103.02662*, 2021.

[34] 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.

[35] 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.

[36] Y. Jia, W. Wu, R. Wang, J. Hou, and S. Kwong, "Joint optimization for pairwise constraint propagation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 3168–3180, 2020.

[37] L. Bai, J. Liang, and F. Cao, "Semi-supervised clustering with constraints of different types from multiple information sources," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 43, no. 9, pp. 3247–3258, 2020.

[38] S. Kan, Y. Cen, Y. Li, V. Mladenovic, and Z. He, "Relative order analysis and optimization for unsupervised deep metric learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 999–14 008.

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

[40] 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.

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[42] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*. University of California Los Angeles LA USA, 1967, pp. 281–297.

[43] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

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

[45] 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.

[46] 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.

[47] C. Xu, R. Lin, J. Cai, and S. Wang, "Deep image clustering by fusing contrastive learning and neighbor relation mining," *Knowledge-Based Systems*, vol. 238, p. 107967, 2022.

[48] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

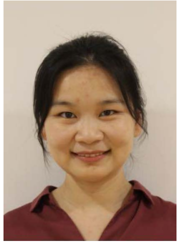
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[50] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, no. Dec, pp. 583–617, 2002.

[51] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, pp. 193–218, 1985.

Yecheng Guo Yecheng Guo is a postgraduate research student in the School of Computer and Information Technology at Shanxi University. His research interest is in the areas of deep clustering.

Liang Bai Liang Bai received his Ph.D degree in Computer Science from Shanxi University in 2012. He is currently a Professor with the School of Computer and Information Technology, Shanxi University. His research interest is in the areas of cluster analysis. He has published several papers in his research ¨fields, including IEEE TPAMI, IEEE TKDE, IEEE TFS, ICML, KDD, AAAI.



Xian Yang Dr Xian Yang is currently a lecturer (Assistant Professor) at Alliance Manchester Business School, the University of Manchester. Before joining AMBS, she worked as an Assistant Professor at the Department of Computer Science from HKBU, a researcher at Microsoft Research Asia and a research fellow at the Data Science Institute of Imperial College London. Dr Xian Yang received her PhD degree from the Department of Computing at Imperial College London in 2016. Her research interests include artificial intelligence in healthcare, natural language processing, data mining and computational epidemiology.

Jiye Liang Jiye Liang received the Ph.D degree from Xi'an Jiaotong University. He is a professor in Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, the School of Computer and Information Technology, Shanxi University. His research interests include artificial intelligence, granular computing, data mining, and machine learning. He has published more than 160 international papers in his research ¨fields, including AIJ, JMLR, IEEE TPAMI, IEEE TKDE, IEEE TFS, DMKD, ICML, KDD, AAAI.