Graph Contrastive Clustering

Huasong Zhong[1]*, Jianlong Wu[23]*, Chong Chen[1,4]†, Jianqiang Huang[1],
Minghua Deng[4], Liqiang Nie[2], Zhouchen Lin[5], Xian-Sheng Hua[1]
[1]DAMO Academy, Alibaba Group [2]Shandong University [3]Zhejiang Lab
[4]School of Mathematical Sciences, Peking University
[5]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
huasong.zhs@alibaba-inc.com, jlwu1992@sdu.edu.cn, {cheung.cc, jianqiang.hjq}@alibaba-inc.com,
dengmh@pku.edu.cn, nieliqiang@gmail.com, zlin@pku.edu.cn, huaxiansheng@gmail.com

Abstract

Recently, some contrastive learning methods have been proposed to simultaneously learn representations and clustering assignments, achieving significant improvements. However, these methods do not take the category information and clustering objective into consideration, thus the learned representations are not optimal for clustering and the performance might be limited. Towards this issue, we first propose a novel graph contrastive learning framework, and then apply it to the clustering task, resulting in the Graph Constrastive Clustering (GCC) method. Different from basic contrastive clustering that only assumes an image and its augmentation should share similar representation and clustering assignments, we lift the instance-level consistency to the cluster-level consistency with the assumption that samples in one cluster and their augmentations should all be similar. Specifically, on the one hand, we propose the graph Laplacian based contrastive loss to learn more discriminative and clustering-friendly features. On the other hand, we propose a novel graph-based contrastive learning strategy to learn more compact clustering assignments. Both of them incorporate the latent category information to reduce the intra-cluster variance as well as increase the inter-cluster variance. Experiments on six commonly used datasets demonstrate the superiority of our proposed approach over the state-of-the-art methods.[1]

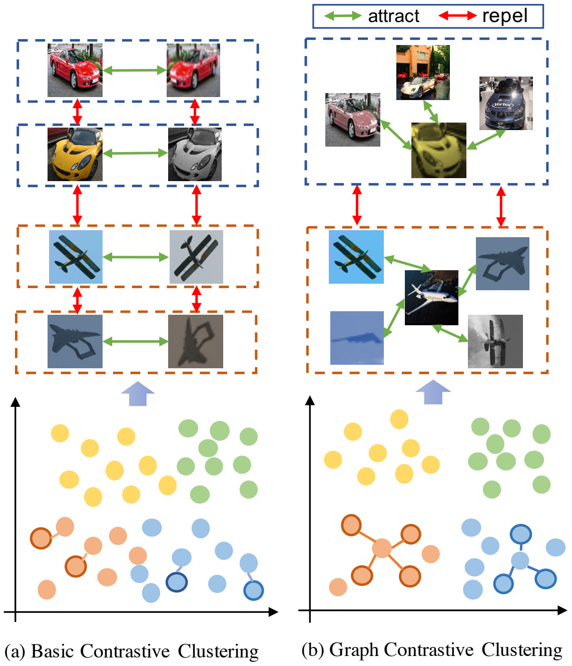

attract
repel
(a) Basic Contrastive Clustering
(b) Graph Contrastive Clustering

Figure 1. Motivation of the proposed GCC. (a) Existing contrastive learning based clustering methods mainly focus on instance-level consistency, which maximizes the correlation between self-augmented samples and treats all other samples as negative samples. (b) GCC incorporates the category information to perform the contrastive learning at both the instance and the cluster levels, which can better minimize the intra-cluster variance and maximize the inter-cluster variance.

1. Introduction

Based on a large number of annotated training samples, deep learning achieves significant success in the past decade [15]. However, it is very expensive and time-consuming to manually label a large training dataset. It is also impractical to collect a labeled dataset for each domain or task. In this case, clustering attracts more attention recently, which aims to divide the samples into separate clusters without knowing the label information.

*H. Zhong and J. Wu contributed equally and are joint first authors. This work was done when H. Zhong worked full time at DAMO Academy, Alibaba Group.

†Corresponding author.

[1] Code address: https://github.com/mynameischaos/GCC

Clustering [3, 36, 16, 17] is a very challenging task since samples in the same class have various appearances and supervision signals are lacked to train the model. Classic clustering methods [43, 10, 2, 35, 37], such as spectral clustering [26] and subspace clustering [24, 9], suffer from two obvious limitations, including indiscriminative feature representation and sub-optimal solution for clustering caused by the separation of feature extraction and clustering. Some recent deep learning based methods can well handle the above issues. For example, auto-encoder related methods [34, 19] minimize the reconstruction error and assign various regularization terms in the latent feature space, such as the KL-divergence [39]. Deep adaptive clustering (DAC) [3] maximizes the similarity between self-augmented samples to adaptively train the neural network. Deep comprehensive correlation mining (DCCM) [36] thoroughly investigates various kinds of correlation among samples and features. These approaches achieve good clustering performance, but their upper bound accuracy is limited since the learned features are not discriminative enough.

Recently, contrastive learning [4] has received much attention in unsupervised feature learning, which emphasizes the importance of data augmentation and maximizes the agreement between two augmented samples. Because of its success, a few approaches [16, 44, 23, 36] are proposed to jointly optimize the contrastive learning and clustering. For instance, partition confidence maximisation (PICA) [16] learns the most semantically plausible clustering solution by maximizing partition confidence, which corresponds to the cluster-wise contrastive learning. Instead of only using the cluster contrast in PICA, deep robust clustering (DRC) [44] adopts the conventional contrastive learning in feature and cluster space simultaneously. These methods significantly improve the clustering performance, but they still face another obvious issue: both of them still follow the basic framework of contrastive learning and only assume that a sample and its augmentations should be similar in the feature space, which does not incorporate the latent category information into clustering.

In view of the above limitations, we propose the graph contrastive framework and apply it to the clustering task, resulting in the Graph Contrastive Clustering (GCC) method. As shown in Figure 1, we assume that samples in one cluster and their augmentations should share similar feature representations and clustering assignments, which lifts the commonly-used instance-level consistency in PICA and DRC to the cluster-level consistency. By incorporating the latent category/cluster information, GCC can help to learn more discriminative features and better clustering assignments, which is more suitable for the clustering task. Specifically, we first construct a similarity graph based on the current features, then we apply it to both representation learning and clustering learning. For representation learning, the graph Laplacian based contrastive loss is proposed to learn more clustering-friendly features. For clustering learning, a novel graph-based contrastive learning strategy is proposed to learn more compact clustering assignments. Both of them can help to decrease the intra-class variance and increase inter-class variance. Experimental results on six challenging datasets validate the effectiveness of the proposed method. We also perform extensive ablation analysis to demonstrate the superiority of graph contrastive.

Our main contributions are summarized as follows:

1. By incorporating the latent category information, we propose a novel graph contrastive framework, which assumes that samples in one cluster and their augmentations should share similar representations and clustering assignments. This framework lifts the tradition instance-level consistency to cluster-level consistency, thus can better reduce the intra-class variance as well as increase the inter-class variance.

2. We apply the proposed graph contrastive framework to the clustering task, and come up with the graph contrastive clustering method (GCC), which consists of two graph contrastive modules. For representation graph contrastive module, a graph Laplacian based contrastive loss is proposed to learn more discriminative and clustering-friendly features. For assignment graph contrastive module, a novel graph-based contrastive learning strategy is proposed to learn more compact clustering assignments.

3. We conduct extensive experiments on image clustering and our proposed method achieves significant improvement on various datasets. We also conduct an extensive ablation study to validate the effectiveness of each proposed module.

2. Related work

2.1. Deep Clustering

According the difference in self-supervised signal, deep clustering methods can be mainly divided into two categories, including the reconstruction based methods [39, 28, 8, 11, 40] and the self-augmentation based methods [3, 36, 17, 12, 16, 33, 44].

The former adopts the auto-encoder [34] framework and imposes different regularization terms on the latent feature learning. For example, DEC [39] and IDEC [11] minimize the KL-divergence for features in the latent subspace. Peng et al. [28] incorporate the sparsity prior. Yang et al. [40] combine it with K-means. DEPICT [8] proposes the relative entropy minimization based on convolutional auto-encoder. The latter focuses on exploiting the consistent information between original images and their transformed images to train the network. DAC [3] adopts a binary pairwise classification framework for image clustering to make the feature learning in a "supervised" manner. DCCM [36] comprehensively utilizes various kinds of correlations among representations. IIC [17] maximizes the mutual information of positive pairs to make them keep a similar assignment probability. PICA [16] learns the most semantically plausible clustering solution by maximizing partition confidence. DRC [44] tries to learn invariant features and clusters by introducing contrastive learning to optimize the consistency between image and its augmentation. SCAN [33] utilizes a three-stage method to improve the clustering. These approaches achieve good results, but they ignore the connections between cluster assignment learning and representation learning. As a contrast, our method considers their connections, and simultaneously learns both feature representation and cluster assignment.

2.2. Contrastive Learning

Recently, constrastive learning achieves significant progress, and it can learn discriminative feature representation without any manual annotations. For example, Wu et al. [38] introduce a memory bank to store the embedding of instance representation. Zhuang et al. [45] extend the above memory bank by learning an embedding function to maximize a metric of local aggregation, causing similar data instances to move together in the embedding space. MoCo [14] views contrastive learning as dictionary loop-up and builds a dynamic dictionary with a queue and a moving-averaged encoder. MoCo v2 [6] makes simple modifications to MoCo by using an MLP projection head and more data augmentations. simCLR [4] simplifies recently proposed contrastive self-supervised learning algorithms without requiring specialized architectures or a memory bank. simCLR v2 [5] finds that bigger self-supervised models are more label efficient, performing significantly better when fine-tuned on only a few labeled examples, even though they have more capacity to potentially overfit. Tian et al. [31, 32] extend the constrastive learning to the multi-view case and representation distillation. Although these methods can learn good feature representations, how to apply them to the clustering task to improve the performance still remains challenging.

3. Graph Contrastive Clustering

3.1. Problem Formulation

Given a set of N unlabelled images $\mathbf{I} = \{I_1, ..., I_N\}$ from K different categories, deep clustering aims to separate these images into K different clusters by convolutional neural network (CNN) models such that the images with the same semantic labels can be grouped into the same cluster. Here we aim to learn a deep CNN network based mapping function Φ with parameters θ, such that each image I_i can be mapped to (z_i, p_i), where z_i is the d-dimensional representation feature with regularization $\|z_i\|_2 = 1$ and p_i is the K-dimension assignment probability which satisfies $\sum_{j=1}^{K} p_{ij} = 1$. Then the cluster assignment for the i-th sample ($i = 1, ..., N$) can be predicted by the following maximum likelihood:

$$\ell_i = \arg\max_j (p_{ij}), 1 \leq j \leq K.$$

3.2. Graph Contrastive (GC)

Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, \cdots, v_N\}$. The edge set E can be represented by the adjacency matrix A such that:

$$A_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let d_i be the degree of v_i, if we define $D = \begin{bmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{bmatrix}$, then the normalized symmetric Graph Laplacian of G can be defined as:

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}. \tag{2}$$

It is easy to check that $L_{ij} = -\frac{A_{ij}}{\sqrt{d_i d_j}}, i \neq j$.

Given N representation features $\mathbf{x} = \{x_1, ..., x_N\}$ with unit ℓ_2 norm, the intuition of GC is that x_i should be close to x_j if $A_{ij} > 0$ while x_i should be far away from x_j if $A_{ij} = 0$. Assume that the graph can be partitioned into several communities, the intuition of GC tells us that the similarities of feature representations in the same community should be larger than that between communities. Approximately, we can define

$$\mathcal{S}_{intra} = \sum_{L_{ij} < 0} -L_{ij} S(x_i, x_j) \tag{3}$$

as the total intra-community similarity and

$$\mathcal{S}_{inter} = \sum_{L_{ij} = 0} S(x_i, x_j) \tag{4}$$

as the total inter-community similarity, where $S(x_i, x_j)$ is the similarity between x_i and x_j. Then we can mathematically define the loss of GC as:

$$\mathcal{L}_{GC} = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{\sum_{L_{ij}<0} -L_{ij} S(x_i, x_j)}{\sum_{L_{ij}=0} S(x_i, x_j)} \right). \tag{5}$$

Minimizing $\mathcal{L}_{GC}$ can simultaneously increase total intra-community similarity and decrease total inter-community similarity, which can improve the separableness and lead to the result that learned feature representations are consistent with the graph structure.

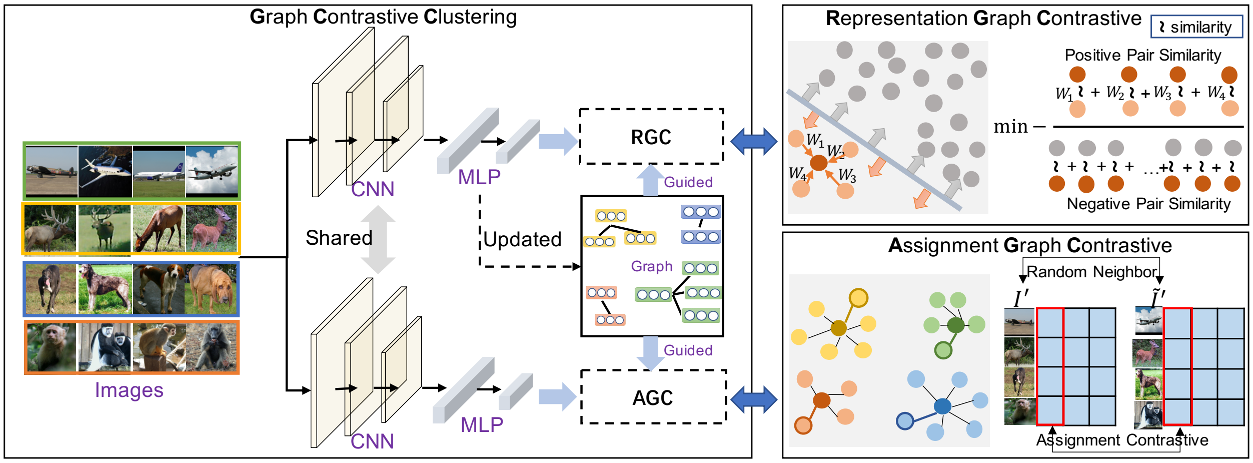


Figure 2. Framework of the proposed Graph Contrastive Clustering. GCC has two heads with shared CNN parameters. The first head is a representation graph contrastive (RGC) module, which helps to learn clustering-friendly features. The second head is an assignment graph contrastive (AGC) module, which leads to a more compact cluster assignment.

3.3. Framework of GCC

We introduce a novel end-to-end deep clustering framework by applying GC to both representation learning and assignment learning. As shown in Figure 2, there are two heads with shared CNN parameters in our GCC model. The upper head is a representation graph contrastive (RGC) module, which learns clustering-friendly features based on representation graph contrastive learning. The bottom head is an assignment graph contrastive (AGC) module, which achieves the final cluster assignment with cluster-level graph contrastive learning. With these two modules, GCC can simultaneously learn more discriminative features and clusters to improve clustering. We will present the details of GCC below.

3.3.1 Graph Construction

Since the deep learning model usually fluctuates during training, the representation features of an epoch may have large biases. We take advantage of moving average to reduce this kind of bias before graph construction. To be specific, assume that $\Phi_\theta^{(t)}$ is the model and $Z^{(t)} = (z_1^{(t)}, \cdots, z_N^{(t)}) = (\Phi_\theta^{(t)}(I_1), \cdots, \Phi_\theta^{(t)}(I_N))$ are the representation features of t-th epoch, the moving average of representation features can be defined as:

$$\bar{z}_i^{(t)} = \frac{(1-\alpha)\bar{z}_i^{(t-1)} + \alpha z_i^{(t)}}{\|(1-\alpha)\bar{z}_i^{(t-1)} + \alpha z_i^{(t)}\|_2}, i = 1, \cdots, N,$$

where α is a parameter to trade-off current and past effects and $\bar{z}_i^{(0)} = z_i^{(0)}$. Then we can construct the KNN graph by

$$A_{ij}^{(t)} = \begin{cases} 1, & \text{if } \bar{z}_j^{(t)} \in \mathcal{N}^k(\bar{z}_i^{(t)}) \text{ or } \bar{z}_i^{(t)} \in \mathcal{N}^k(\bar{z}_j^{(t)}); \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

for $i, j = 1, \cdots, N$. After that, the Graph Laplacian $L^{(t)}$ can be obtained by Eq. (2).

3.3.2 Similarity Function

To compute the similarity between two samples, we adopt the Gaussian kernel function which is commonly used in spectral clustering. The similarity in GC loss Eq. (5) can be defined as:

$$S(x_i, x_j) = e^{-\|x_i - x_j\|_2^2/\tau},$$

where τ is a parameter that represents variance or temperature. Since $\|x_i - x_j\|_2^2 = \|x_i\|_2^2 + \|x_j\|_2^2 - 2x_i \cdot x_j = 2 - 2x_i \cdot x_j$, we use the following similarity function as a substitution:

$$S(x_i, x_j) = e^{x_i \cdot x_j/\tau}. \quad (7)$$

3.3.3 Representation Graph Contrastive

Assume $\mathbf{I}' = \{I_1', ..., I_N'\}$ is a random transformation of original images, and their corresponding features are $\mathbf{z}' = (z_1', \cdots, z_N')$. According to graph contrastive mentioned before, z_i' and z_j' should be similar if they are linked while be far away if they are disconnected. Let $\mathbf{x} = \mathbf{z}'$ in Eq. (5), we can get the loss of RGC learning as:

$$\mathcal{L}_{RGC}^{(t)} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{\sum_{L_{ij}^{(t)}<0} -L_{ij}^{(t)} e^{z_i' \cdot z_j'/\tau}}{\sum_{L_{ij}=0} e^{z_i' \cdot z_j'/\tau}}\right). \quad (8)$$

3.3.4 Assignment Graph Contrastive

For traditional contrastive learning based clustering, images and their augmentations should share similar cluster assignment distribution, *e.g.* the index of images and their augmentations assigned to cluster k should be consistent. It is reasonable but does not take advantage of clustering information. As the model gets better and better during training, images and their neighbors also should share similar cluster assignment distribution with high probability. Due to this motivation, we propose the assignment graph contrastive learning.

Assume that $\mathbf{I}^{'} = \{I_1^{'}, ..., I_N^{'}\}$ are the random augmentations of original images and $\tilde{\mathbf{I}}^{'} = \{\tilde{I}_1^{'}, ..., \tilde{I}_N^{'}\}$ satisfies that $\tilde{I}_j^{'}$ is a transformation of a random neighbor of I_i according to graph $A^{(t)}$, the assignment probability matrix for $\mathbf{I}^{'}$ and $\tilde{\mathbf{I}}^{'}$ can be defined as

$$\mathbf{p}^{'} = \begin{bmatrix} p_1^{'} \\ ... \\ p_N^{'} \end{bmatrix}_{N \times K} \quad \text{and } \tilde{\mathbf{p}}^{'} = \begin{bmatrix} p_{\mathrm{RN}(I_1)}^{'} \\ ... \\ p_{\mathrm{RN}(I_N)}^{'} \end{bmatrix}_{N \times K},$$

where $\mathrm{RN}(I_i)$ denotes a random neighbor of image I_i. We can reformulate them by the following column vector forms:

$$\mathbf{q}^{'} = [q_1^{'}, \quad ... \quad , q_K^{'}]_{N \times K},$$
$$\tilde{\mathbf{q}}^{'} = [\tilde{q}_1^{'}, \quad ... \quad , \tilde{q}_K^{'}]_{N \times K},$$

where $q_i^{'}$ and $\tilde{q}_i^{'}$ can tell us which pictures in $\mathbf{I}^{'}$ and $\tilde{\mathbf{I}}^{'}$ will be assigned to cluster i, respectively. Then we can define the AGC learning loss as:

$$\mathcal{L}_{AGC} = -\frac{1}{K} \sum_{i=1}^{K} \log \left(\frac{e^{q_i^{'} \cdot \tilde{q}_i^{'} / \tau}}{\sum_{j=1}^{K} e^{q_i^{'} \cdot \tilde{q}_j^{'} / \tau}} \right). \tag{9}$$

3.3.5 Cluster Regularization Loss

In deep clustering, it is easy to fall into a local optimal solution that assign most samples into a minority of clusters. To avoid trivial solution, we also add a clustering regularization loss similar to PICA [16] and SCAN [33]:

$$\mathcal{L}_{CR} = \log(K) - H(\mathcal{Z}), \tag{10}$$

where H is the entropy function, $\mathcal{Z}_i = \frac{\sum_{j=1}^{N} q_{ij}}{\sum_{i=1}^{K} \sum_{j=1}^{N} q_{ij}}$, and $\mathbf{q} = [q_1, \quad \cdots \quad , q_K]_{N \times K}$ is the assign probability of $\mathbf{I}$.

Then the overall objective function of GCC can be formulated as:

$$\mathcal{L} = \mathcal{L}_{RGC} + \lambda \mathcal{L}_{AGC} + \eta \mathcal{L}_{CR}, \tag{11}$$

where λ and η are weight parameters.

3.4. Model Training

The objective function in Eq. (11) is differentiable and end-to-end, enabling the conventional stochastic gradient descent algorithm for model training. The training procedure is summarized in Algorithm 1.

Algorithm 1: Training algorithm for GCC

```
Input: Training images $\mathcal{I} = \{I_1, \dots, I_N\}$, training
       epochs $N_{ep}$, and number of clusters $K$.
Output: A deep clustering model with parameters
        $\theta$.
Initializing graph $A$ and parameters $\theta$;
for each epoch do
    Step 1: Sampling a random mini-batch of
      images and their neighbors according to $A$;
    Step 2: Generating augmentations for the
      sampled images and their neighbors;
    Step 3: Computing RGC loss by Eq. (8);
    Step 4: Computing AGC loss by Eq. (9);
    Step 5: Computing cluster regularization loss
      according to Eq. (10);
    Step 6: Update $\theta$ with SGD by minimizing the
      overall loss according to Eq. (11);
    Step 7: Update $A$ according to Eq. (6).
end
```

4. Experiments

4.1. Experimental Settings

4.1.1 Datasets

We conducted extensive experiments on six widely-adopted benchmark datasets. For a fair comparison, we adopted the same experimental setting as [3, 16]. The characteristics of these datasets are introduced in the following.

CIFAR-10/100: [20] The image size is 32 × 32 × 3. 10 classes and 20 super-classes are considered for the CIFAR-10/CIFAR-100 dataset in experiments. All 60,000 images are jointly utilized to clustering.

STL-10: [7] The STL-10 is an image recognition dataset containing 500/800 training/test images for each of 10 classes with image size 96 × 96 × 3 and additional 100,000 samples from several unknown classes for training stage.

ImageNet-10 and ImageNet-Dogs: [3] Two subsets of ImageNet [21]: the former contains 10 randomly selected subjects and the latter contains 15 dog breeds. Their size is set to 96 × 96 × 3.

Tiny-ImageNet: [22] It is a very challenging tiny ImageNet dataset for clustering with 200 classes. There are 100,000/10,000 training/test images with dimension 64 × 64 × 3 in each category.

4.1.2 Evaluation Metrics

Similar to [16], we adopted three standard metrics for evaluating the performance of clustering, including Accuracy (ACC), Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI).

Table 1. Clustering performance of different methods on six challenging datasets.

Datasets	CIFAR-10			CIFAR-100			STL-10			ImageNet-10			Imagenet-dog-15			Tiny-ImageNet		
Methods	NMI	ACC	ARI	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	0.065	0.025	0.005
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	0.063	0.022	0.004
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	0.069	0.027	0.005
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	0.072	0.029	0.005
AE	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	0.131	0.041	0.007
DAE	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	0.127	0.039	0.007
GAN	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	0.135	0.041	0.007
DeCNN	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	0.111	0.035	0.006
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	0.113	0.036	0.006
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	0.102	0.033	0.006
DEC	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	0.115	0.037	0.007
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	0.190	0.066	0.017
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	0.224	0.108	0.038
IIC	-	0.617	-	-	0.257	-	-	0.610	-	-	-	-	-	-	-	-	-	-
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	0.277	0.098	0.040
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	0.233	0.321	**0.139**	0.056
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**	**0.347**	0.138	**0.075**

4.1.3 Compared Methods

We compared the proposed method with both traditional and deep learning based methods, including K-means, spectral clustering (SC) [30], agglomerative clustering (AC) [10], the nonnegative matrix factorization (NMF) based clustering [2], auto-encoder (AE) [1], denoising auto-encoder (DAE) [34], GAN [29], deconvolutional networks (DECNN) [42], variational auto-encoding (VAE) [19], deep embedding clustering (DEC) [39], jointly unsupervised learning (JULE) [41], deep adaptive image clustering (DAC) [3], invariant information clustering [17], deep comprehensive correlation Mining (DCCM) [36], partition confidence maximisation (PICA) [16], and deep robust clustering (DRC) [44].

4.1.4 Implementation Details

We utilized PyTorch [27] to implement all experiments. In our framework, we used ResNet-18 [15] as the main network architecture and train networks on one Tesla P100 GPU. We first train the model by simCLR [4] loss with 50 epochs. The SGD optimizer is adopt with $lr = 0.4$, a weight decay $1e-4$ and momentum coefficient 0.9. The learning rate decays by cosine scheduler with decay rate 0.1. The batch size is set to 256 and the same data augmentation is adopted as [4]{color jitter, random grayscale, randomly resized crop}. The temperatures in RGC and AGC are set to $\tau = 0.1$ and $\tau = 1.0$, respectively. For hyper-parameters, we set $\alpha = 0.5$, $\lambda = 0.5$ and $\eta = 1.0$ for all datasets. For the construction of KNN graph, we set $K = 5$ and utilized the efficient similarity search library 'Faiss' [2]. Even for 1 million samples with 256 dimensional features on a CPU with 64 cores and 2.5GHz, it takes about 50 seconds to construct a KNN graph. Therefore, its time cost is neglectable and the KNN graph construction does not limit its application to large scale datasets. For the ablation study, we adopted the same setting as SCAN [33] to perform self-labeling processing.

4.2. Experimental Results and Analysis

In Table 1, we presented the clustering results of GCC and other related methods on these six challenging datasets. The results of other methods are directly copied from DRC [44]. Based on the results, we can first see that deep learning based methods achieve much better results than traditional clustering methods due to the large parameter capacity. For instance, the accuracy of most deep learning based clustering methods on CIFAR-10 is much higher than 0.3, while the accuracy of these classic methods, including SC, AC, and NMF, is lower than 0.25. Secondly, these contrastive learning based methods, such as PICA, DRC and GCC, are more suitable for the clustering task since they can learn more discriminative feature representation. Most importantly, it is obvious that our GCC significantly surpasses other methods by a large margin on most benchmarks under three different evaluation metrics. Even compared with the recent state-of-the-art methods PICA and DRC, the improvement of GCC is also remarkable. Take the clustering accuracy for example, our results are 12.9%, 10.5%, 4.1% higher than that of the second best method DRC on CIFAR-10, CIFAR-100 and STL-10, respectively. The above results can well demonstrate the effectiveness and robustness of our proposed method.

4.3. Ablation Study

According to the objective function in Eq. (11), there are three different losses in total. In this section, we will demonstrate that RGC loss in Eq. (8), AGC loss in Eq. (9),

[2] https://github.com/facebookresearch/faiss

Table 2. Effect of two graph contrastive losses, where ✓ means using graph information. Metric: ACC.

RGC	AGC	CIFAR-10	CIFAR-100	ImageNet-10
		0.752	0.438	0.878
✓		0.809	0.463	0.884
	✓	0.825	0.462	0.893
✓	✓	**0.856**	**0.472**	**0.901**

Table 3. Effect of cluster regularization loss. Metric: ACC.

Method	CIFAR-10	CIFAR-100	ImageNet-10
GCC *w/o* CR	0.680	0.348	0.828
GCC	**0.856**	**0.472**	**0.901**

and cluster regularization loss in Eq. (10) are all very important to improve the performance. We will also evaluate the influence of a post-processing strategy used in SCAN [33] and the superiority of graph contrastive for clustering-oriented representation learning over the basic contrastive learning method.

4.3.1 Effect of Graph Contrastive Loss

We first investigated how RGC and AGC losses affect the clustering performance on CIFAR-10, CIFAR-100 and ImageNet-10. Results are shown in Table 2. Method in the first line only adopts the basic contrastive loss. Compared with it, both RGC and AGC improve the clustering results on all three datasets, especially on CIFAR-10. All best results are achieved by GCC, which implies that both RGC and ARC terms are indispensable.

4.3.2 Effect of Cluster Regularization Loss

Deep clustering methods can easily fall into a local optimal solution when most samples are assigned to the same cluster. We examined how the cluster regularization loss addresses this problem. As shown in Table 3, we can see that it significantly helps to improve the clustering performance. It is interesting to see the cluster regularization loss has little impact on ImageNet-10 since it is a relatively easy dataset where images from different classes are well separated.

4.3.3 Effect of Self-labeling Fine-tuning

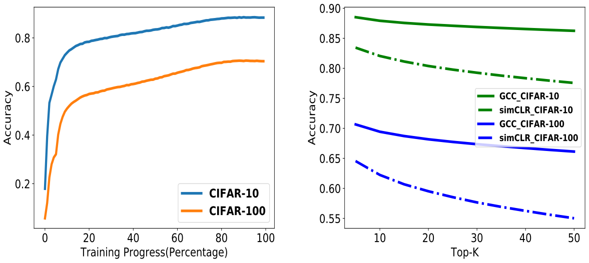
SCAN [33] proposes a three-stage method for image clustering and achieved high performance. The clustering results benefit a lot by fine-tuning through self-labeling. For a fair comparison, we also performed self-labeling after GCC and the results are shown in Table 4. We can see that GCC outperforms SCAN [33] both before and after self-labeling on all three datasets reported in the paper of SCAN, which indicates that GCC learns more clustering-friendly representations and better clustering assignments.

Table 4. Effect of self-labeling. * means that adopting self-label post-processing. Metric: ACC.

Method	CIFAR-10	CIFAR-100	STL-10
SCAN	0.818	0.422	0.755
GCC	**0.856**	**0.472**	**0.788**
SCAN*	0.883	0.507	0.809
GCC*	**0.901**	**0.523**	**0.833**

Table 5. Comparison of features learned by GCC and simCLR. Metric: ACC.

Method	CIFAR-10	CIFAR-100
simCLR + SC	0.660	0.292
GCC + SC	**0.746**	**0.367**
simCLR + K-means	0.628	0.380
GCC + K-means	**0.754**	**0.420**



(a) Training Accuracy of Top-5 NN (b) Top-K NN Accuracy

Figure 3. Top-K nearest neighbor accuracy of GCC and simCLR: (a) The evolution of top-5 NN accuracy for CIFAR-10 and CIFAR-100 during the training process of GCC. (b) The comparison of top-K NN accuracy of CIFAR-10 and CIFAR-100 when varying K from 1 to 50.

4.3.4 Superiority of Graph Contrastive

To demonstrate the superiority of Graph Contrastive on learned features, we performed two more quantitative analysis. First, we directly adopted K-means and Spectral Clustering (SC) [30] to cluster the learned features of basic contrastive learning (simCLR [4]) and GCC on testing datasets (10,000 samples). For a fair comparison, here we only used RGC loss for GCC, and the implementation details are same to simCLR [4]. As we can see from Table 5, the clustering performance of GCC is much better than simCLR, which verifies that the features learned by GCC are more conducive to clustering.

Furthermore, we calculated the accuracy of top-K nearest neighbor (NN) obtained by GCC and simCLR, and the results are shown in Figure 3. We can see that the top-5 NN accuracy of GCC becomes better and better during training from Figure 3(a), which verifies the motivation of our graph contrastive learning. The comparison of GCC and simCLR are shown in Figure 3(b), where the results of GCC are consistently better than simCLR when varying K from 1 to 50.

Several recent methods [13, 18] propose to extend basic

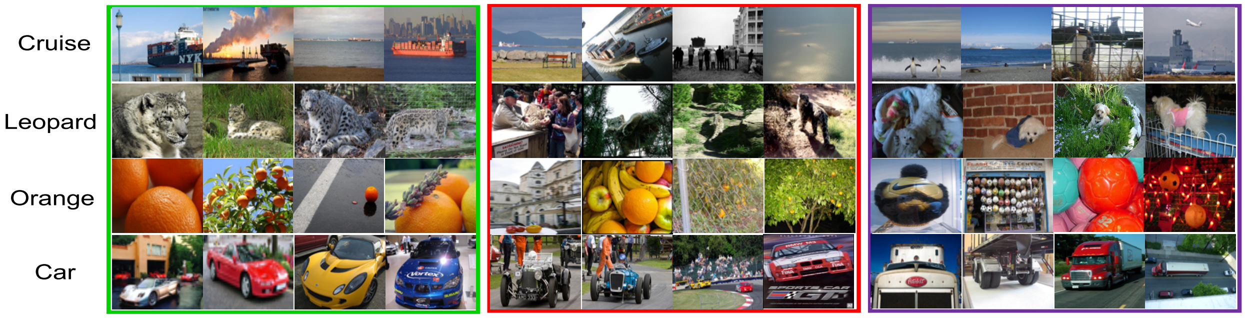


Figure 4. Case study on ImageNet-10. Successful cases (left), false negative cases (middle), and false positive failure cases (right).

Table 6. Comparison of graph contrastive and ordinary contrastive learning with multiple positives. Metric: ACC.

Method	CIFAR-10	CIFAR-100	ImageNet-10
Multi-positive	0.807	0.426	0.872
GCC	**0.856**	**0.472**	**0.901**

contrastive learning by simply adding more positive samples. We replaced RGC with this contrastive loss to perform clustering analysis and the result is shown in Table 6. It is clear that GCC performs much better, which again demonstrates the advantages of our GC framework.

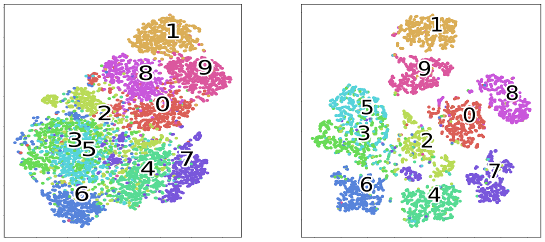
4.4. Qualitative Study

4.4.1 Visualization of Representations

To further illustrate that the features obtained by GCC are more suitable for clustering than simCLR, we visualized them on CIFAR-10 by t-SNE [25]. To be specific, we plotted the predictions of 6,000 randomly selected samples with the ground-truth classes color encoded by using t-SNE. As shown in Figure 5, samples in the same class are more compact and samples of different classes are significantly better separated for GCC. For example, the samples of class 2 (in green-yellow) are divided into two parts in simCLR but gathered together in GCC.

4.4.2 Case Study

At last, we investigated both success and failure cases to get extra insights into our method. Specifically, we studied three cases of four classes from ImageNet-10, including success cases, false negative failure cases, and false positive cases. As shown in Figure 4, GCC can successfully group together images of the same class with different backgrounds and angles. Two different failure cases tell us that GCC mainly learns the shape of objects. Samples of different classes with a similar pattern may be grouped together and samples of the same class with different patterns may be separated into different classes. It is hard to look into the details at the absence of the ground-truth labels, which is still an unsolved problem for unsupervised learning.



(a) Basic Contrastive Learning (b) Graph Contrastive Clustering

Figure 5. t-SNE visualization for basic contrastive learning and our graph contrastive learning on the CIFAR-10 dataset.

5. Conclusion

To address the shortage of existing contrastive learning based clustering methods, we propose a novel graph contrastive learning framework, which is then applied to the clustering task and we come up with the GCC method. Different from basic contrastive clustering that only maximizes the correlation between an image and its augmentation, we lift the instance-level feature consistency to the cluster-level consistency with the assumption that samples in one cluster and their augmentations should have similar representations. We perform extensive experiments on six widely-adopted benchmarks to demonstrate that GCC learns more clustering-friendly representations than basic contrastive learning and outperforms a wide range of state-of-the-art methods.

Acknowledgment

This work was supported by the Key-Area Research and Development (R&D) Program of Guangdong Province (no. 2019B121204008), the National Natural Science Foundation (NSF) of China (no.s 62006140, 61625301 and 61731018), the Key R&D Program of Shandong Province (Major scientific and technological innovation projects with no. 2020CXGC010111), the NSF of Shandong Province (no. ZR2020QF106), PKU-Baidu Fund (no. 2020BD006), Major Scientific Research Project of Zhejiang Lab (no.s 2019KB0AC01 and 2019KB0AB02), and Beijing Academy of Artificial Intelligence.

References

[1] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *NeurIPS*, pages 153–160, 2007.

[2] Deng Cai, Xiaofei He, Xuanhui Wang, Hujun Bao, and Jiawei Han. Locality preserving nonnegative matrix factorization. In *IJCAI*, volume 9, pages 1010–1015, 2009.

[3] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *IEEE ICCV*, pages 5879–5887, 2017.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223, 2011.

[8] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *IEEE ICCV*, pages 5747–5756, 2017.

[9] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *IEEE CVPR*, pages 2790–2797, 2009.

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

[11] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[12] Philip Haeusser, Johannes Plapp, Vladimir Golkov, Elie Aljalbout, and Daniel Cremers. Associative deep clustering: Training a classification network with no labels. In *German Conference on Pattern Recognition*, pages 18–32, 2018.

[13] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *arXiv preprint arXiv:2010.09709*, 2020.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE CVPR*, pages 9729–9738, 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016.

[16] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *IEEE CVPR*, pages 8849–8858, 2020.

[17] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *IEEE ICCV*, pages 9865–9874, 2019.

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

[22] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7, 2015.

[23] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *AAAI*, pages 8547–8555, 2021.

[24] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[25] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[26] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NeurIPS*, pages 849–856, 2002.

[27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[28] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Yi Zhang. Deep subspace clustering with sparsity prior. In *IJCAI*, pages 1925–1931, 2016.

[29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[30] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[31] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, pages 776–794, 2020.

[32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

[33] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *ECCV*, 2020.

[34] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(12), 2010.

[35] Jianlong Wu, Zhouchen Lin, and Hongbin Zha. Essential tensor learning for multi-view spectral clustering. *IEEE Transactions on Image Processing*, 28(12):5910–5922, 2019.

[36] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *IEEE ICCV*, pages 8150–8159, 2019.

[37] Jianlong Wu, Xingyu Xie, Liqiang Nie, Zhouchen Lin, and Hongbin Zha. Unified graph and low-rank tensor learning for multi-view clustering. In *AAAI*, volume 34, pages 6388–6395, 2020.

[38] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE CVPR*, pages 3733–3742, 2018.

[39] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016.

[40] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, pages 3861–3870, 2017.

[41] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *IEEE CVPR*, pages 5147–5156, 2016.

[42] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *IEEE CVPR*, pages 2528–2535, 2010.

[43] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *NeurIPS*, pages 1601–1608, 2005.

[44] Huasong Zhong, Chong Chen, Zhongming Jin, and Xian-Sheng Hua. Deep robust clustering by contrastive learning. *arXiv preprint arXiv:2008.03030*, 2020.

[45] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *IEEE ICCV*, pages 6002–6012, 2019.