GATCluster: Self-Supervised Gaussian-Attention Network for Image Clustering

Chuang Niu[1], Jun Zhang[2], Ge Wang[3], and Jimin Liang[1]

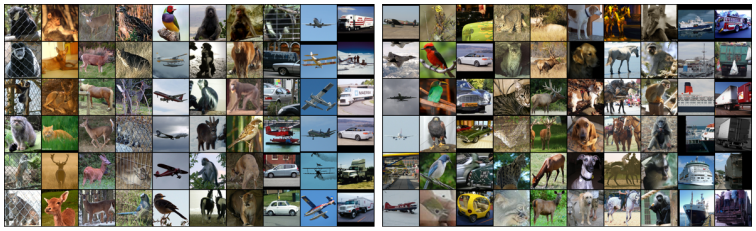
[1] School of Electronic Engineering, Xidian University, Xi'an, Shaanxi 710071, China
[2] Tencent AI Lab, Shenzhen, Guangdong 518057, China
[3] Rensselare Polytechnic Institute, Troy, NY 12180, US

Abstract. We propose a self-supervised Gaussian ATtention network for image Clustering (GATCluster). Rather than extracting intermediate features first and then performing traditional clustering algorithms, GATCluster directly outputs semantic cluster labels without further post-processing. We give a Label Feature Theorem to guarantee that the learned features are one-hot encoded vectors and the trivial solutions are avoided. Based on this theorem, we design four self-learning tasks with the constraints of transformation invariance, separability maximization, entropy analysis, and attention mapping. Specifically, the transformation invariance and separability maximization tasks learn the relations between samples. The entropy analysis task aims to avoid trivial solutions. To capture the object-oriented semantics, we design a self-supervised attention mechanism that includes a Gaussian attention module and a soft-attention loss. Moreover, we design a two-step learning algorithm that is memory-efficient for clustering large-size images. Extensive experiments demonstrate the superiority of our proposed method in comparison with the state-of-the-art image clustering benchmarks.

1 Introduction

Clustering is the process of separating data into groups according to sample similarity, which is a fundamental unsupervised learning task with numerous applications. Similarity or discrepancy measurement between samples plays a critical role in data clustering. Specifically, the similarity or discrepancy is determined by both data representation and distance function.

Before the extensive application of deep learning, handcrafted features, such as SIFT [30] and HoG [8], and domain-specific distance functions are often used to measure the similarity. Based on the similarity measurement, various rules were developed for clustering. These include space-partition based (e.g., k-means [31] and spectral clustering [34]) and hierarchical methods (e.g., BIRCH [53]). With the development of deep learning techniques, researchers have been dedicated to leverage deep neural networks for joint representation learning and clustering, which is commonly referred to as deep clustering. Although significant advances have been witnessed, deep clustering still suffers from an inferior performance for natural images (e.g., ImageNet [38]) in comparison with that for simple handwritten digits in MNIST.



(a) **(b)**

Fig. 1. Clustering results on STL10. Each column represents a cluster. (a) Sample images clustered by the proposed model without attention, where the clustering principles focus on trivial cues, such as texture (first column), color (second column), or background (fifth column); and (b) Sample images clustered by the proposed model with attention, where the object concepts are well captured.

Various challenges arise when applying deep clustering on natural images. *First*, many deep clustering methods use stacked auto-encoders (SAE) [2] to extract clustering-friendly intermediate features by imposing some constraints on the hidden layer and the output layer respectively. However, pixel-level reconstruction is not an effective constraint for extracting discriminative semantic features of natural images, since these images usually contain much more instance-specific details that are unrelated to semantics. Recent progress [5][13][45][21] has demonstrated that it is an effective way to directly map data to label features just as in the supervised classification task. However, training such a model in an unsupervised manner is difficult to extract clustering-related discriminative features. *Second*, clusters are expected to be defined by appropriate semantics while current methods tend to group the images by alternative principles (such as colors, textures, or background), as shown in Fig. 1. *Third*, the dynamic change between different clustering principles during the training process tends to make the model unstable and easily get trapped in trivial solutions that assign all samples to a single or very few clusters. *Fourth*, the existing methods were usually evaluated on small images (32×32 to 96×96). This is mainly due to the large batch of samples required for training the deep clustering model preventing us from processing large images on memory-limited devices.

To tackle these problems, we propose a self-supervised Gaussian attention network for clustering (GATCluster) that directly outputs discriminative semantic label features. Theoretically, we introduce a Label Feature Theorem, ensuring that the learned features are one-hot encoded vectors and the trivial solutions can be avoided. Accordingly, we design four self-learning tasks with the constraints of transformation invariance, separability maximization, entropy analysis, and attention mapping. GATCluster is trained in a completely unsupervised manner, as all the guiding signals for clustering are self-generated during training. Specifically, 1) the transformation invariance maximizes the similarity between a sample and its random transformations. 2) The separability maximization task explores both similarity and discrepancy of each paired samples

to guide the model learning. 3) The entropy analysis task helps avoid trivial solutions. 4) To capture object-orientated semantics, an attention mechanism is proposed based on the observation that the discriminative information of objects is usually presented on local regions.

For processing large-size images, we develop an efficient two-step learning algorithm. First, the pseudo-targets over a large batch of samples are computed statistically in a split-and-merge manner. Second, the model is iteratively trained on the same batch in a supervised learning manner using the pseudo-targets. It should be noted that GATCluster is trained by optimizing all loss functions simultaneously instead of alternately. Our learning algorithm is memory-efficient and thus easy to process large images.

To summarize, the contributions of this paper include

(1) We introduce a Label Feature Theorem ensuring that the learned features are one-hot encoded vectors and trivial solutions can be avoided.

(2) We propose an attention module with a Gaussian kernel and a soft-attention loss to capture object-oriented semantics. To our best knowledge, this is the first attempt in exploring the attention mechanism for unsupervised learning.

(3) Our two-step learning algorithm that is memory-efficient makes it possible to perform the clustering on large-size images.

(4) Extensive experimental results demonstrate that the proposed GATCluster significantly outperforms or is comparable to the state-of-the-art methods on image clustering datasets. Our code has been made publicly available at https://github.com/niuchuangnn/GATCluster.

2 Related work

2.1 Deep Clustering

We divide the deep clustering methods into two categories: 1) intermediate-feature-based deep clustering and 2) semantic deep clustering. The first category extracts intermediate features and then conducts conventional clustering. The second one directly constructs a nonlinear mapping between original data and cluster labels. By doing so, the samples are clustered just as in the supervised classification task, without any need for additional processing.

Some intermediate-feature-based deep clustering methods usually employ the SAE [15][2] or its variants [43][33][32][23] to extract intermediate features, and then conduct k-means [18][6] or spectral clustering [20]. Instead of performing representation learning and clustering separately, some studies integrate these two stages into a unified framework [46][25][47][41][51][10][22][9][55]. However, as applied to complex natural images, the reconstruction loss of SAE tends to overestimate the importance of low-level features. In constrast to the SAE-based methods, some methods [48][17][16] directly use the convolutional neural network (CNN) or multi-layer perceptron (MLP) for representation learning by designing specific loss functions. Unfortunately, the high-dimensional nature of intermediate features are too abundant to effectively reveal the discriminative semantic information of natural images.

Semantic deep clustering methods have recently shown a great promise for clustering. To train such models in the unsupervised manner, various rules have been designed for supervision. DAC [5] recasts clustering into a binary pairwise-classification problem, and the supervised labels are adaptively generated by thresholding the similarity matrix. As an extension to DAC, DCCM [45] investigates both pair-wise sample relations and triplet mutual information between deep and shallow layers. However, these two methods are practically susceptible to trivial solutions. IIC [21] directly trains a classification network by maximizing the mutual information between original data and their transformations. However, the computation of mutual information requires a very large batch size in the training process, which is challenging to apply on large images.

2.2 Self-supervised learning

Self-supervised learning can learn general features by optimizing cleverly designed objective functions of some pretext tasks, in which all supervised pseudo labels are automatically generated from the input data without manual annotations. Various pretext tasks were proposed, including image completion [37], image colorization [52], jigsaw puzzle [35], counting [36], rotation [12], clustering [4][51], etc. For the pretext task of clustering, cluster assignments are often used as pseudo labels, which can be obtained by k-means or spectral clustering algorithms. In our study, both the self-generated relation of paired samples and object attention are used as the guiding signals for clustering.

2.3 Attention

In recent years, the attention mechanism has been successfully applied to various tasks in machine learning and computer vision, such as machine translation [42], image captioning and visual question answering [1], GAN [50], person re-identification [28], visual tracking [44], crowd counting [29], weakly- and semi-supervised semantic segmentation [26], and text detection and recognition [14]. Given the ground-truth labels, the attention weights are learned to scale-up more related local features for better predictions. However, it is still not explored for deep clustering models that are trained without human-annotated labels. In this work, we design a Gaussian-kernel-based attention module and a soft-attention loss to learn the attention weights in a self-supervised manner.

2.4 Learning algorithm of deep clustering

Various of learning algorithms are designed for training deep clustering models. Most existing deep clustering models are alternatively trained between updating cluster assignments and network parameters [46], or between different clustering heads [21]. Some of them need pre-training in an unsupervised [46][47][51] or supervised manner [17][16]. On the other hand, some studies [5][45] directly train the deep clustering models by optimizing all component objective functions

simultaneously. However, they do not consider the statistical constraint and are susceptible to trivial solutions. In this work, we propose a two-step self-supervised learning algorithm that is memory-efficient for processing the large batch training with large-size images.

3 Method

3.1 Label Feature Theorem and problem formulation

Given a set of samples $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ and the predefined number of clusters k, this work aims to automatically divide $\mathcal{X}$ into k groups by predicting the label features $\boldsymbol{l}_i \in R^k$ of each sample $\boldsymbol{x}_i$, where N is the total number of samples.

We first review the theorem introduced by DAC [5]. Clustering can be recast as a binary classification problem that measures the similarity and discrepancy between two samples and then determines whether they belong to the same cluster. For each sample $\boldsymbol{x}_i$, the label feature $\boldsymbol{l}_i = f(\boldsymbol{x}_i; \boldsymbol{w})$ is computed, where $f(\cdot, \boldsymbol{w})$ is a mapping function with parameters $\boldsymbol{w}$. The parameters $\boldsymbol{w}$ are obtained by minimizing the following objective function:

$$\begin{aligned} &\min_{\boldsymbol{w}} \mathbf{E}(\boldsymbol{w}) = \sum_{i,j}^{N} L(r_{ij}, \boldsymbol{l}_i \cdot \boldsymbol{l}_j), \\ &s.t.\ \forall i\ \|\boldsymbol{l}_i\|_2 = 1, l_{ih} \geq 0, h = 1, \cdots, k, \end{aligned} \tag{1}$$

where r_{ij} is the ground-truth relation between samples x_i and x_j, i.e., $r_{ij} = 1$ indicates that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same cluster and $r_{ij} = 0$ otherwise. In the unsupervised setting, r_{ij} can be estimated by thresholding [5][45] or the approach introduced in Section 3.3; the inner product $\boldsymbol{l}_i \cdot \boldsymbol{l}_j$ is the cosine distance between two samples as the label feature is constrained with $\|\boldsymbol{l}_i\|_2 = 1$; L is a loss function instantiated by the binary cross entropy; and k is the predefined number of clusters. The theorem proved in [5] claimed that if the optimal value of Eq. (1) is attained, the learned label features will be k diverse one-hot vectors. Thus, the cluster identification c_i of image $\boldsymbol{x}_i$ can be directly obtained by selecting the maximum of label features, i.e., $c_i = argmax_h\ l_{ih}$. However, it practically tends to obtain trivial solutions that assign all samples to a single or a few clusters. In the supplementary, we give a theoretical analysis of why it will get trapped in the trivial solutions when optimizing Eq. (1).

Based on the above analysis, we formulate the clustering as the following optimization problem with a probability and a nonempty cluster constraint:

$$\begin{aligned} &\min_{w} \mathbf{E}(w) = \sum_{i,j}^{N} L(r_{ij}, \frac{\boldsymbol{l}_i}{\|\boldsymbol{l}_i\|_2} \cdot \frac{\boldsymbol{l}_j}{\|\boldsymbol{l}_j\|_2}) - \sum_{i=1}^{N} \boldsymbol{l}_i \cdot \boldsymbol{l}_i, \\ &s.t.\ \forall i\ \|\boldsymbol{l}_i\|_1 = 1, 0 \leq \boldsymbol{l}_{ih} \leq 1, h = 1, \cdots, k. (probability) \\ &\forall h\ p_h > 0, p_h = \frac{1}{N} \sum_{i=1}^{N} l_{ih}. (nonempty\ cluster) \end{aligned} \tag{2}$$

Although DCCM [45] also implements the probability constraint, it cannot guarantee the trivial solutions being avoided. However, our probability constraint is necessary for the nonempty cluster constraint and computing the entropy loss to avoid trivial solutions. In the nonempty cluster constraint, p_h denotes the frequency of assigning N samples into the h^{th} cluster. And we have a Label Feature Theorem (the proof of this theorem can be found in the supplementary) as follows:

Label Feature Theorem. *If the optimal value of Eq. (2) is attained, for* $\forall i, j, \boldsymbol{l}_i \in E^k, \boldsymbol{l}_i \neq \boldsymbol{l}_j \Leftrightarrow r_{ij} = 0$, $\boldsymbol{l}_i = \boldsymbol{l}_j \Leftrightarrow r_{ij} = 1$, *and* $|\{\boldsymbol{l}_i\}_{i=1}^N| = k$, *where* $|\cdot|$ *denotes the cardinality of a set.*

Label Feature Theorem ensures that the learned features are one-hot encoded vectors in which each bit represents a cluster, and all predefined k clusters are nonempty. However, the learned features may focus on various of cues for clustering as introduced in Section 1. To capture the object-oriented semantics in the unsupervised setting, we propose a Gaussian attention mechanism with a soft-attention loss. By incorporating the Label Feature Theorem with the attention mechanism, we formulate clustering as the following optimization problem,

$$\min_{w} \mathbf{E}(w) = \sum_{i,j=1}^{N} L_R(r_{ij}, \boldsymbol{l}_i, \boldsymbol{l}_j) + \sum_{i=1}^{N} (\alpha_1 L_T(\boldsymbol{l}_i) + \alpha_2 L_E(\boldsymbol{l}_i) + \alpha_3 L_A(\boldsymbol{l}_i, \boldsymbol{l}_i^a)), \quad (3)$$

where L_R and L_T correspond to the first and second items in the objective function of Eq. (2), L_E is to satisfy the nonempty cluster constraint, L_A represents the attention loss, which is described in Section 3.3, and $\alpha_1, \alpha_2, \alpha_3$ are the hyper-parameters to balance the importance of different losses. In practice, the probability constraint is always satisfied by setting the label features as the outputs of the softmax function. To optimize the problem of Eq. (3) for unsupervised clustering, we propose a GATCluster model with four self-learning tasks as introduced in the following sections.

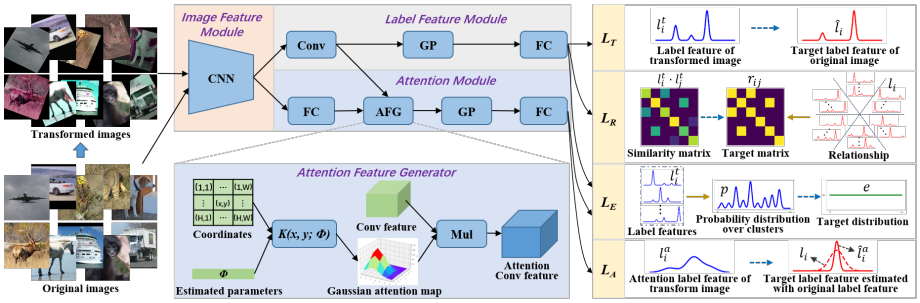


Fig. 2. GATCluster framework. *CNN* is a convolutional network, *GP* means global pooling, *Mul* represents channel-independent multiplication, *Conv* is a convolution layer, *FC* is a fully connected layer, and *AFG* represents an attention feature generator.

3.2 Framework

GATCluster consists of the following three components: 1) an image feature module, 2) a label feature module, and 3) an attention module, as shown in Figure 2. The image feature module extracts convolutional features of images with a fully convolutional network. The label feature module, which contains a convolutional layer, a global pooling layer and a fully-connected layer, aims to map the convolutional features to semantic label features. The attention module makes the model focus on discriminative local regions automatically, facilitating the capture of object-oriented semantics. The attention module consists of three submodules, including a fully connected layer for estimating the parameters of Gaussian kernel, an attention feature generator, and a global pooling layer followed by another fully connected layer for computing the attention label features. The attention feature generator has three inputs, i.e., the estimated Gaussian parameters Φ, the convolutional features from the label feature module, and the two-dimensional coordinates of the attention map that are self-generated according to the attention map size H and W.

In the training stage, we design four learning tasks driven by the transformation invariance, separability maximization, entropy analysis and attention mapping. Specifically, the transformation invariance and separability maximization losses are computed with respect to the predicted label features, the attention loss is evaluated with the attention module outputs, and the entropy loss is used to supervise both the label feature module and the attention module. For inference, only the image feature module and label feature module are combined as a classifier to suggest the cluster assignments. The clustering results in successive training stages are visualized in Fig. 3.

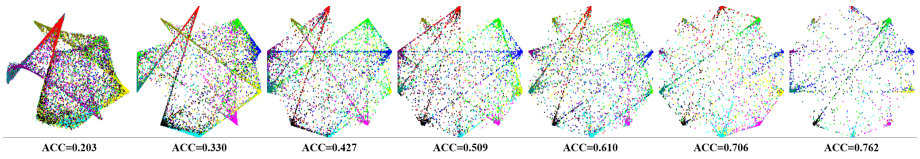


Fig. 3. Visualization of clustering results in successive training stages (from left to right) for 13K images in ImageNet-10. The results are visualized based on the predicted label features, and each point represents an image and the colors are rendered with the ground-truth label. The corresponding clustering accuracy is presented under each picture. Details can be found in the supplementary.

3.3 Self-learning tasks

Transformation invariance task An image after any practically reasonable transformations still reflect the same object. Hence, these transformed images should have similar feature representations. To learn such a similarity, the label

feature $\boldsymbol{l}_i$ of original sample $\boldsymbol{x}_i$ is constrained to be close to its transformed counterpart $\boldsymbol{l}_i^t$ of $T(\boldsymbol{x}_i)$, where T is a practically reasonable transformation function. In this work, the transformation function is predefined as the composition of random flipping, random affine transformation, and random color jittering, see Fig. 2. Specifically, the loss function is defined as

$$L_T(\boldsymbol{l}_i^t, \hat{\boldsymbol{l}}_i) = -\boldsymbol{l}_i^t \cdot \hat{\boldsymbol{l}}_i, \tag{4}$$

where $\hat{\boldsymbol{l}}_i$ is the target label feature of an original image $\boldsymbol{x}_i$ that is recomputed as:

$$\hat{l}_{ih} = \frac{l_{ih}/z_h}{\sum_{h'} l_{ih'}/z_{h'}}, \quad z_h = \sum_{j=1}^{M} l_{jh}, h = 1, 2, \cdots, k, \tag{5}$$

where M is the number of samples, i.e., the batch size used in the training process. Eq. (5) can balance the sample assignments by dividing the cluster assignment frequency z_h, preventing the empty clusters.

Separability maximization task If the relations between all pairs of samples are well captured, the label features will be one-hot encoded vectors as introduced in Section 3.1. However, the ground-truth relations cannot be obtained in the unsupervised learning environment. Therefore, we evaluate the relationships of a batch of samples as follows:

$$r_{ij} = \begin{cases} 1, & c_i = c_j \ or \ i = j, \\ 0, & otherwise, \end{cases} \tag{6}$$

where $c_i = c_j$ indicates that the samples x_i and x_j belong to the same cluster, $i = j$ indicates that the similarity of a sample to itself is 1. To get the cluster identification c_i, k-means algorithm is conducted on a set of samples based on the predicted label features. Instead of estimating r_{ij} with a high pre-defined threshold as in [5][45], our approach can determine it adaptively.

The separability maximization task is to improve the purity of clusters by encouraging samples that are similar to be closer to each other while dissimilar samples to be further away from each other. The loss function is defined as:

$$L_R(r_{ij}, \boldsymbol{l}_i, \boldsymbol{l}_j) = -\, r_{ij} \log(d(\boldsymbol{l}_i, \boldsymbol{l}_j)) - (1 - r_{ij}) \log(1 - d(\boldsymbol{l}_i, \boldsymbol{l}_j)), \tag{7}$$

where $d(\boldsymbol{l}_i, \boldsymbol{l}_j) = \frac{\boldsymbol{l}_i}{\|\boldsymbol{l}_i\|_2} \cdot \frac{\boldsymbol{l}_j}{\|\boldsymbol{l}_j\|_2}$ is the cosine distance.

Entropy analysis task The entropy analysis task is designed to avoid trivial solutions by satisfying the nonempty cluster constraint in Eq. (2). We maximize the entropy of the empirical probability distribution $\boldsymbol{p}$ over k cluster assignments. Thus, the loss function is defined as

$$
\begin{aligned}
L_E(\boldsymbol{l}_1, \cdots, \boldsymbol{l}_m) &= \sum_{h=1}^{k} p_h \log(p_h), \\
p_h &= \frac{1}{m} \sum_{i=1}^{m} l_{ih}, h = 1, \cdots, k,
\end{aligned}
\tag{8}
$$

where $\boldsymbol{p} = [p_1, \cdots, p_k]$ is estimated with the predicted label features of m samples, which can be a subset of the whole batch. Actually, maximizing the entropy will steer $\boldsymbol{p}$ towards a uniform distribution (denoted by $\boldsymbol{e}$ in Fig. 2), i.e., $\forall h, p_h \to \frac{1}{m} > 0$, and thus the nonempty constraint is satisfied so that the trivial solutions are avoided according to the **Label Feature Theorem**.

Attention mapping task The attention mapping task aims to make the model recognize the most discriminative local regions concerning the whole image semantic. The basic idea is that the response to the discriminative local regions should be more intense than that to the entire image. To this end, there are two problems to be solved: 1) how to design the attention module for localizing the discriminative local regions? and 2) how to train the attention module in a self-supervised manner?

With regard to the first problem, we design a two-dimensional Gaussian kernel $K(\boldsymbol{u}; \Phi)$ to generate an attention map A as:

$$
\begin{aligned}
A(x, y) = K(\boldsymbol{u}; \Phi) &= e^{-\frac{1}{\alpha}(\boldsymbol{u}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{u}-\boldsymbol{\mu})}, \\
x = 1, \cdots, H, \ &and \ y = 1, \cdots, W,
\end{aligned}
\tag{9}
$$

where $\boldsymbol{u} = [x, y]^T$ denotes the coordinate vector, $\Phi = [\boldsymbol{\mu}, \Sigma]$ denotes the parameters of the Gaussian kernel, $\boldsymbol{\mu} = [\mu_x, \mu_y]^T$ is the mean vector that defines the most discriminative location, $\Sigma \in \mathbf{R}^{2\times 2}$ is the covariance matrix that defines the shape and size of a local region, α is a predefined hyper parameter, and H and W are the height and width of the attention map. In our implementation, the coordinates are normalized over $[0, 1]$. Taking CNN features as the input, a fully connected layer is used to estimate the parameter Φ. Then, the model can focus on the discriminative local region by multiplying each channel of convolutional features with the attention map. The weighted features are mapped to the attention label features using a global pooling layer and a fully connected layer, as shown in Fig. 2. It should be noted that there are also alternative designs of the attention module to generate attention maps, such as a convolution layer followed by a sigmoid function. However, we obtained better results with the parameterized Gaussian attention module due to that it has a much less number of parameters to be estimated, and the Gaussian attention prior fits for capturing the local object in the unsupervised learning setting.

With regard to the second problem, we define a soft-attention loss as

$$
L_A(\boldsymbol{l}_i^a, \hat{\boldsymbol{l}}_i^a) = \frac{1}{k} \sum_{h=1}^{k} -\hat{l}_{ih}^a \log(l_{ih}^a) - (1 - \hat{l}_{ih}^a) \log(1 - l_{ih}^a), \tag{10}
$$

$$\hat{l}^a_{ih} = \frac{l^2_{ih}/z_h}{\sum_{h'} l^2_{ih'}/z_{h'}}, h = 1, \cdots, k, \tag{11}$$

where $\boldsymbol{l}^a_i$ is the output of the attention module, $\hat{\boldsymbol{l}}^a_i$ is the target label feature for regression, and z_h is the same as in Eq. (5) to balance the cluster assignments. As defined in Eq. (11), the target label feature $\hat{\boldsymbol{l}}^a_i$ encourages the current high scores and suppresses low scores of the whole image label feature $\boldsymbol{l}_i$, thus making $\hat{\boldsymbol{l}}^a_i$ a more confident version of the whole image label feature $\boldsymbol{l}_i$, see Fig. 2 for demonstration. By doing so, the local image region, which is localized by the attention module, is discriminative in terms of the whole image semantics. In practice, the local region usually presents the expected object or the discriminative part as shown in Fig. 4.

Algorithm 1: GATCluster learning algorithm.

Input: Dataset $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, k, M, m_1, m_2
Output: Cluster label c_i of $\boldsymbol{x}_i \in \mathcal{X}$
Randomly initialize network parameters $\boldsymbol{w}$;
Initialize e = 0;
while $e <$ *total epoch number* **do**
  for $b \in \{1, 2, \ldots, \lfloor \frac{N}{M} \rfloor\}$ **do**
    Select M samples as $\mathcal{X}_b$ from $\mathcal{X}$;
    ***Step-1*:**
    for $u \in \{1, 2, \ldots, \lfloor \frac{M}{m_1} \rfloor\}$ **do**
      Select m_1 samples as $\mathcal{X}_u$ from $\mathcal{X}_b$;
      Calculate the label features of $\mathcal{X}_u$;
    end
    Concatenate all label features of M samples ;
    Calculate pseudo targets $T_b = \{(\hat{\boldsymbol{l}}_i, r_{ij}, \hat{\boldsymbol{l}}^a_i)\}$ of $\mathcal{X}_b$ with Eqs. (5), (6), and (11);
    ***Step-2*:**
    Randomly transform samples in $\mathcal{X}_b$ as $\mathcal{X}^t_b$;
    for $v \in \{1, 2, \ldots, \lfloor \frac{M}{m_2} \rfloor\}$ **do**
      Randomly select m_2 samples as $[\mathcal{X}_v; T_v]$ from $[\mathcal{X}^t_b; T_b]$;
      Optimize $\boldsymbol{w}$ on $[\mathcal{X}_v; T_v]$ by minimizing Eq. (3) using Adam ;
    end
  end
  $e := e + 1$
end
foreach $\boldsymbol{x}_i \in \mathcal{X}$ **do**
  $\boldsymbol{l}_i := f(\boldsymbol{l}_i; \boldsymbol{w})$;
  $c_i := \arg\max_h(l_{ih})$;
end

3.4 Learning algorithm

We develop a two-step learning algorithm that combines all the self-learning tasks to train GATCluster in an unsupervised learning manner. The total loss function is defined by Eq. (3), in which the entropy loss is computed with the label features $\boldsymbol{l}_i$ and $\boldsymbol{l}^a_i$ predicted by the label feature module and attention module respectively, i.e., $L_E = L_E(\boldsymbol{l}_1, \cdots, \boldsymbol{l}_M) + L_E(\boldsymbol{l}^a_1, \cdots, \boldsymbol{l}^a_M)$.

The proposed two-step learning algorithm is presented in Algorithm 1. Since deep clustering methods usually require a large batch of samples for training, it is difficult to process large images with a memory-limited device. To tackle this problem, we divide the large-batch-based training process into two steps for

each iteration. The first step is the forward process that statistically calculates the pseudo-targets for a large batch of M samples using the model trained in the last iteration. To achieve this with a memory-limited device, we further split the large batch into sub-batches and calculate the label features for each sub-batch of m_1 samples independently. Then, all label features of M samples are concatenated for computing their pseudo labels. Given these samples with pseudo labels, the second step is the supervised training process that trains the model with a sub-batch of m_2 samples iteratively.

4 Experiments and Results

4.1 Data

We evaluated the proposed and the compared deep clustering methods on five datasets, including STL10 [7] that contains 13K 96×96 images of 10 clusters, ImageNet-10 [5] that contains 13K images of 10 clusters, ImageNet-Dog [5] that contains 19.5K images of 15 dog subcategories, Cifar10 and Cifar100-20 [24]. The image size of ImageNet-10 and ImageNet-Dog is around 500×300. Cifar10 and Cifar100-20 both contain 60K 32×32 images, and have 10 and 20 clusters.

4.2 Implementation details

At the training stage, especially at the beginning, samples tend to be clustered by color cues. Therefore, we took grayscale images as inputs except for ImageNet-Dog, as the color plays an important role in differentiating the sub-categories of dogs. It is noted that the images are converted to grayscale after applying the random color jittering during training. For simplicity, we assume $\Sigma = \begin{bmatrix} \delta & 0 \\ 0 & \delta \end{bmatrix}$, and there are only three parameters for Gaussian kernel to be estimated, i.e., $[\mu_x, \mu_y; \delta]$. We used Adam to optimize the network parameters and the base learning rate was set to 0.001. We set the batch size M to 1000 for STL10 and ImageNet-10, 1500 for ImageNet-Dog, 4000 for Cifar10, and 6000 for Cifar100-20. The sub-batch size m_1 in calculating pseudo targets can be adjusted according to the device memory and will not affect the results. The sub-batch size m_2 was 32 for all experiments. Hyper parameters α, α_1, α_2, and α_3 were empirically set to 0.05, 5, 5, and 3 respectively.

In all experiments, we used the VGG-style convolutional network with batch normalization to implement the image feature extraction module. The architecture details of different experiments can be found in the supplementary.

4.3 Evaluation metrics

We used three popular metrics to evaluate the performance of the involved clustering methods, including Adjusted Rand Index (ARI) [19], Normalized Mutual Information (NMI) [40] and clustering Accuracy (ACC) [27].

Table 1. Comparison with the existing methods. GATCluster-128 resizes input images to 128×128 for ImageNet-10 and ImageNet-Dog while other models take 96×96 images as inputs. On Cifar10 and Cifar100, the input size is 32×32. The best three results are highlighted in **bold** .

Method	STL10			ImageNet-10			ImageNet-dog			Cifar10			Cifar100-20		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
k-means [31]	0.192	0.125	0.061	0.241	0.119	0.057	0.105	0.055	0.020	0.229	0.087	0.049	0.130	0.084	0.028
SC [34]	0.159	0.098	0.048	0.274	0.151	0.076	0.111	0.038	0.013	0.247	0.103	0.085	0.136	0.090	0.022
AC [11]	0.332	0.239	0.140	0.242	0.138	0.067	0.139	0.037	0.021	0.228	0.105	0.065	0.138	0.098	0.034
NMF [3]	0.180	0.096	0.046	0.230	0.132	0.065	0.118	0.044	0.016	0.190	0.081	0.034	0.118	0.079	0.026
AE [2]	0.303	0.250	0.161	0.317	0.210	0.152	0.185	0.104	0.073	0.314	0.239	0.169	0.165	0.100	0.048
SAE [2]	0.320	0.252	0.161	0.335	0.212	0.174	0.183	0.113	0.073	0.297	0.247	0.156	0.157	0.109	0.044
SDAE [43]	0.302	0.224	0.152	0.304	0.206	0.138	0.190	0.104	0.078	0.297	0.251	0.163	0.151	0.111	0.046
DeCNN [49]	0.299	0.227	0.162	0.313	0.186	0.142	0.175	0.098	0.073	0.282	0.240	0.174	0.133	0.092	0.038
SWWAE [54]	0.270	0.196	0.136	0.324	0.176	0.160	0.159	0.094	0.076	0.284	0.233	0.164	0.147	0.103	0.039
CatGAN [39]	0.298	0.210	0.139	0.346	0.225	0.157	N/A	N/A	N/A	0.315	0.265	0.176	N/A	N/A	N/A
GMVAE [9]	0.282	0.200	0.146	0.334	0.193	0.168	N/A	N/A	N/A	0.291	0.245	0.167	N/A	N/A	N/A
JULE-SF [48]	0.274	0.175	0.162	0.293	0.160	0.121	N/A	N/A	N/A	0.264	0.192	0.136	N/A	N/A	N/A
JULE-RC [48]	0.277	0.182	0.164	0.300	0.175	0.138	0.138	0.054	0.028	0.272	0.192	0.138	0.137	0.103	0.033
DEC [46]	0.359	0.276	0.186	0.381	0.282	0.203	0.195	0.122	0.079	0.301	0.257	0.161	0.185	0.136	0.050
DAC* [5]	0.434	0.347	0.235	0.503	0.369	0.284	0.246	0.182	0.095	0.498	0.379	0.280	0.219	0.162	0.078
DAC [5]	0.470	**0.366**	**0.257**	0.527	0.394	0.302	0.275	0.219	0.111	0.522	**0.396**	**0.306**	0.238	**0.185**	**0.088**
IIC [21]	**0.499**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	**0.617**	N/A	N/A	**0.257**	N/A	N/A
DCCM [45]	**0.482**	**0.376**	**0.262**	**0.710**	**0.608**	**0.555**	**0.383**	**0.321**	**0.182**	**0.623**	**0.496**	**0.408**	**0.327**	**0.285**	**0.173**
GATCluster	**0.583**	**0.446**	**0.363**	**0.739**	**0.594**	**0.552**	**0.322**	**0.281**	**0.163**	**0.610**	**0.475**	**0.402**	**0.281**	**0.215**	**0.116**
GATCluster-128	N/A	N/A	N/A	**0.762**	**0.609**	**0.572**	**0.333**	**0.322**	**0.200**	N/A	N/A	N/A	N/A	N/A	N/A

4.4 Comparison with existing methods

Table 1 presents a comparison with the existing methods. Under the same conditions, the proposed method significantly improves the clustering performance by 8%, 7%, and 10% approximately compared with the best of the others in terms of ACC, NMI and ARI on STL10. On ImageNet-10, ACC is improved by 5% compared with the strong baseline that is set by the most recently proposed DCCM [45]. On the sub-category dataset ImageNet-Dog, our method achieves results comparable to that of DCCM. Moreover, our method is capable of processing large images, and in that case the clustering results are further improved. On the small image datasets, i.e., Cifar10 and Cifar100-20, the proposed method also achieves comparable performance relative to the state-of-the-art. Importantly, our GATCluster has the interpretability to the learned cluster semantics by presenting the corresponding local regions. The above results strongly demonstrate the superiority of our proposed method.

4.5 Ablation study

To validate the effectiveness of each component, we conducted the ablation studies as shown in Table 2. Similar to [13], each variant was evaluated ten times and the best accuracy, average accuracy and the standard deviation are reported. Table 2 demonstrates that the best accuracy is achieved when all learning tasks are used with grayscale images. Particularly, the attention mapping (AP) improves the accuracy by up to 4.4 percent for the best accuracy and 4.3 percent for average accuracy. This is attributed to that the attention module has the ability to localize the discriminative regions with respect to the whole image semantic, and thus it can well capture the expected object-oriented semantics,

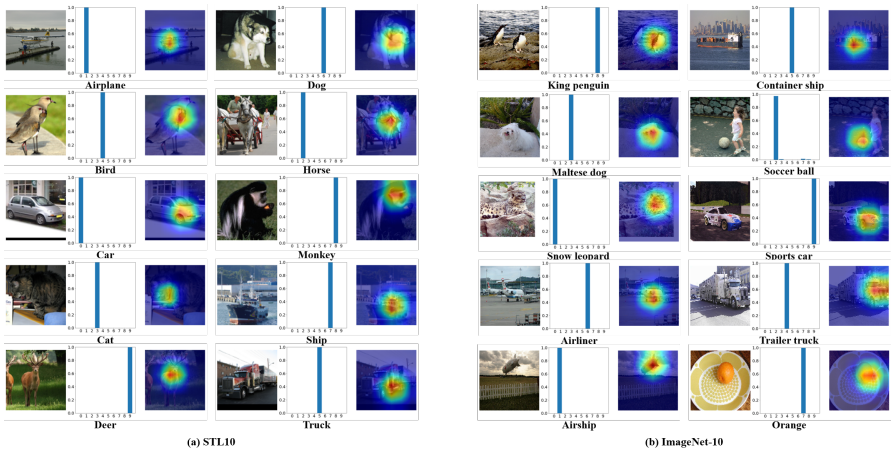


Fig. 4. Visualization of GATCluster on STL10 and ImageNet10. For each class, an example image, the predicted label feature, and the attention map overlaid on the image are shown from left to right.

as shown in Figure 4. In addition, the color information is a strong distraction for object clustering, and better clustering results can be obtained after the color images are changed to grayscale. We do not show the results of ablated entropy loss, as it is easy to get trapped at trivial solutions in our experiments.

Table 2. Ablation studies of GATCluster on STL10.

Method	ACC			NMI			ARI		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
Color	0.556	0.517	0.034	0.427	0.402	0.022	0.341	0.298	0.031
No TI	0.576	0.546	0.016	0.435	0.417	0.012	0.347	0.325	0.014
No SM	0.579	0.529	0.029	0.438	0.412	0.019	0.356	0.310	0.024
No AM	0.539	0.494	0.020	0.416	0.383	0.015	0.316	0.282	0.013
Full setting	**0.583**	0.537	0.033	**0.446**	0.415	0.022	**0.363**	0.315	0.032

4.6 Effectiveness of image size

The biggest image size used by most of the existing unsupervised clustering methods is not larger than 96×96 (e.g., in STL10). However, images in the modern datasets usually have much larger sizes, which are not effectively explored by unsupervised deep clustering methods. With the proposed two-step learning algorithm, we are able to process large images. An interesting question then arises: will large images help produce a better clustering accuracy? To answer this question, we explored the effect of image size on clustering results. Specifically, we evaluated four input image sizes, i.e., 96×96, 128×128, 160×160, and 192×192 by simply resizing the original images on ImageNet-10. We conducted five experimental trails for each image size and report the best

and average accuracies as well as the standard deviation in Table 3. The results show that the clustering performance is significantly improved when the image size is increased from 96×96 to 128×128. It is demonstrated that taking the larger images as inputs can benefit the clustering.

Practically, our proposed methods can be performed on much larger size of images. The clustering results are not further improved when the image size is larger than 128×128. It may be due to that networks become deepened with an increased image size, and thus there is a trade-off between the number of network parameters and the size of the training dataset. However, it is valuable to explore larger size of images for clustering in the future.

Table 3. Clustering results of different image sizes on ImageNet-10.

Size	ACC			NMI			ARI		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
96	0.739	0.708	0.031	0.594	0.581	0.012	0.552	0.529	0.019
128	**0.762**	0.735	0.020	0.609	0.592	0.013	0.572	0.544	0.023
160	0.712	0.669	0.033	0.567	0.511	0.043	0.500	0.453	0.039
192	0.738	0.608	0.067	0.612	0.474	0.071	0.559	0.405	0.079

Table 4. Clustering results of different attention map sizes on ImageNet-10.

Size	ACC			NMI			ARI		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
2	0.746	0.666	0.050	0.625	0.538	0.050	0.569	0.477	0.045
4	0.706	0.678	0.017	0.539	0.528	0.012	0.486	0.473	0.014
6	**0.762**	0.735	0.020	0.609	0.592	0.013	0.571	0.544	0.023
8	0.742	0.719	0.018	0.618	0.594	0.019	0.561	0.536	0.018
10	0.671	0.645	0.020	0.549	0.520	0.021	0.478	0.450	0.020

4.7 Effectiveness of attention map size

A high-resolution attention map will provide precise location but weaken the global semantics. We evaluated the effect of the attention map size on the clustering results for ImageNet-10. We set the size of input image in this experiment to 128×128, and evaluate five sizes of attention map (in pixels): 2×2, 4×4, 6×6, 8×8, and 10×10 as shown in Table 4. It shows that the 6×6 attention map achieves the best results.

5 Conclusion

For deep unsupervised clustering, we introduce a Label Feature Theorem that guarantees the learned features are one-hot encoded and all pre-defined clusters are nonempty. Based on this theorem, we formulate the clustering problem with four self learning tasks. Particulary, the attention mechanism can facilitate the formation of object semantics during the training process. We design a memory-efficient learning algorithm for processing large images. GATCluster model has a great potential for clustering the images with complex contents and discovering discriminative local regions in the unsupervised setting.

6 Acknowledgments

The research was supported by the National Natural Science Foundation of China (61976167, U19B2030, 61571353) and the Science and Technology Projects of Xian, China (201809170CX11JC12).

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR (2018)
2. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., Montreal, U.: Greedy layer-wise training of deep networks. NeurIPS **19**, 153–160 (2007)
3. Cai, D., He, X., Wang, X., Bao, H., Han, J.: Locality preserving nonnegative matrix factorization. In: IJCAI. pp. 1010–1015 (2009)
4. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV. vol. 11218, pp. 139–156 (2018)
5. Chang, J., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep adaptive image clustering. In: ICCV. pp. 5880–5888 (2017)
6. Chen, D., Lv, J., Zhang, Y.: Unsupervised multi-manifold clustering by learning deep representation. In: AAAI Workshops. AAAI Workshops, vol. WS-17. AAAI Press (2017)
7. Coates, A., Ng, A.Y., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: AISTATS. vol. 15, pp. 215–223 (2011)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. vol. 1, pp. 886–893 vol. 1 (2005)
9. Dilokthanakul, N., Mediano, P.A.M., Garnelo, M., Lee, M.C.H., Salimbeni, H., Arulkumaran, K., Shanahan, M.: Deep unsupervised clustering with gaussian mixture variational autoencoders. ArXiv **abs/1611.02648** (2017)
10. Dizaji, K.G., Herandi, A., Deng, C., Cai, W., Huang, H.: Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: ICCV. pp. 5747–5756 (2017)
11. Franti, P., Virmajoki, O., Hautamaki, V.: Fast agglomerative clustering using a k-nearest neighbor graph. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(11), 1875–1881 (2006)
12. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR. OpenReview.net (2018)
13. Haeusser, P., Plapp, J., Golkov, V., Aljalbout, E., Cremers, D.: Associative deep clustering: Training a classification network with no labels. In: Brox, T., Bruhn, A., Fritz, M. (eds.) Pattern Recognition. pp. 18–32 (2019)
14. He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., Sun, C.: An end-to-end textspotter with explicit alignment and attention. In: CVPR (2018)
15. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
16. Hsu, C., Lin, C.: Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. IEEE Transactions on Multimedia **20**(2), 421–429 (2018)
17. Hu, W., Miyato, T., Tokui, S., Matsumoto, E., Sugiyama, M.: Learning discrete representations via information maximizing self-augmented training. In: ICML. vol. 70, pp. 1558–1567 (2017)
18. Huang, P., Huang, Y., Wang, W., Wang, L.: Deep embedding network for clustering. In: ICPR. pp. 1532–1537 (2014)
19. Hubert, L., Arabie, P.: Comparing partitions. Journal of classification **2**(1), 193–218 (1985)
20. Ji, P., Zhang, T., Li, H., Salzmann, M., Reid, I.: Deep subspace clustering networks. In: NeurIPS. pp. 23–32 (2017)

21. Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: ICCV (2019)
22. Jiang, Z., Zheng, Y., Tan, H., Tang, B., Zhou, H.: Variational deep embedding: An unsupervised and generative approach to clustering. In: IJCAI. pp. 1965–1972 (2017)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2013)
24. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
25. Li, F., Qiao, H., Zhang, B.: Discriminatively boosted image clustering with fully convolutional auto-encoders. Pattern Recognition **83**, 161 – 173 (2018)
26. Li, K., Wu, Z., Peng, K.C., Ernst, J., Fu, Y.: Tell me where to look: Guided attention inference network. In: CVPR (2018)
27. Li, T., Ding, C.H.Q.: The relationships among various nonnegative matrix factorization methods for clustering. In: ICDM. pp. 362–371. IEEE Computer Society (2006)
28. Li, W., Zhu, X., Gong, S.: Harmonious attention network for person reidentification. In: CVPR (2018)
29. Liu, J., Gao, C., Meng, D., Hauptmann, A.G.: Decidenet: Counting varying density crowds through attention guided detection and density estimation. In: CVPR (2018)
30. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV (1999)
31. Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: In 5-th Berkeley Symposium on Mathematical Statistics and Probability. pp. 281–297 (1967)
32. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I.J.: Adversarial autoencoders. CoRR **abs/1511.05644** (2015)
33. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) Artificial Neural Networks and Machine Learning. pp. 52–59 (2011)
34. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) NeurIPS, pp. 849–856 (2002)
35. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV. vol. 9910, pp. 69–84 (2016)
36. Noroozi, M., Pirsiavash, H., Favaro, P.: Representation learning by learning to count. In: ICCV. pp. 5899–5907. IEEE Computer Society (2017)
37. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2536–2544 (2016)
38. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
39. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2016)
40. Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining multiple partitions. Journal on Machine Learning Research (JMLR) **3**, 583–617 (2002)
41. Tian, K., Zhou, S., Guan, J.: Deepcluster: A general clustering framework based on deep learning. In: Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Džeroski,

S. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 809–825. Springer International Publishing, Cham (2017)
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) NeurIPS, pp. 5998–6008 (2017)
43. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research **11**(12), 3371–3408 (2010)
44. Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., Maybank, S.: Learning attentions: Residual attentional siamese network for high performance online visual tracking. In: CVPR (2018)
45. Wu, J., Long, K., Wang, F., Qian, C., Li, C., Lin, Z., Zha, H.: Deep comprehensive correlation mining for image clustering. In: ICCV (2019)
46. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: ICML. pp. 478–487 (2016)
47. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In: Precup, D., Teh, Y.W. (eds.) ICML. vol. 70, pp. 3861–3870 (2017)
48. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: CVPR (2016)
49. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Computer Vision and Pattern Recognition (2010)
50. Zhang, H., Goodfellow, I.J., Metaxas, D.N., Odena, A.: Self-attention generative adversarial networks. ArXiv **abs/1805.08318** (2018)
51. Zhang, J., Li, C.G., You, C., Qi, X., Zhang, H., Guo, J., Lin, Z.: Self-supervised convolutional subspace clustering network. In: CVPR (2019)
52. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV. vol. 9907, pp. 649–666. Springer (2016)
53. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: An efficient data clustering method for very large databases. In: SIGMOD Conference (1996)
54. Zhao, J.J., Mathieu, M., Goroshin, R., LeCun, Y.: Stacked what-where auto-encoders. CoRR **abs/1506.02351** (2015)
55. Zhou, P., Hou, Y., Feng, J.: Deep adversarial subspace clustering. In: CVPR (2018)