Deep Comprehensive Correlation Mining for Image Clustering

Jianlong Wu[1*] Keyu Long[2*] Fei Wang[2] Chen Qian[2] Cheng Li[2] Zhouchen Lin[3,✉] Hongbin Zha[3]

[1]School of Computer Science and Technology, Shandong University
[2]SenseTime Research
[3]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University

jlwu1992@sdu.edu.cn, corylky114@gmail.com, {wangfei, qianchen, chengli}@sensetime.com, zlin@pku.edu.cn, zha@cis.pku.edu.cn

Abstract

Recent developed deep unsupervised methods allow us to jointly learn representation and cluster unlabelled data. These deep clustering methods mainly focus on the correlation among samples, e.g., selecting high precision pairs to gradually tune the feature representation, which neglects other useful correlations. In this paper, we propose a novel clustering framework, named deep comprehensive correlation mining (DCCM), for exploring and taking full advantage of various kinds of correlations behind the unlabeled data from three aspects: 1) Instead of only using pairwise information, pseudo-label supervision is proposed to investigate category information and learn discriminative features. 2) The features' robustness to image transformation of input space is fully explored, which benefits the network learning and significantly improves the performance. 3) The triplet mutual information among features is presented for clustering problem to lift the recently discovered instance-level deep mutual information to a triplet-level formation, which further helps to learn more discriminative features. Extensive experiments on several challenging datasets show that our method achieves good performance, e.g., attaining 62.3% clustering accuracy on CIFAR-10, which is 10.1% higher than the state-of-the-art results[1].

1. Introduction

Clustering is one of the fundamental tasks in computer vision and machine learning. Especially with the development of the Internet, we can easily collect thousands of images and videos every day, most of which are unlabeled. It is very expensive and time-consuming to manually label these data. In order to make use of these unlabeled data and investigate their correlations, unsupervised clustering draws much attention recently, which aims to categorize similar data into one cluster based on some similarity measures.

---

[*]Equal contribution and the work was done during interns at SenseTime Research
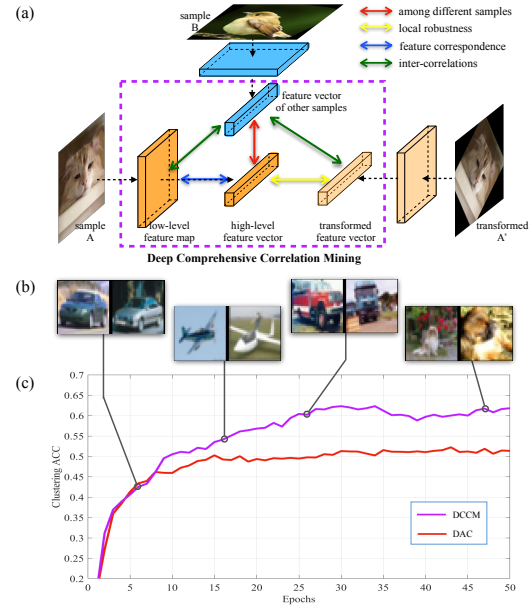[1]Project address: https://github.com/Cory-M/DCCM

Figure 1. Comprehensive correlations mining. (a) Various correlations; (b) Connect pair-wise items in higher semantic level progressively; (c) Better results of DCCM than the state-of-the-art DAC [8] on CIFAR-10 [27]. Best viewed in color!

Image clustering is a challenging task due to the image variance of shape and appearance in the wild. Traditional clustering methods [55, 19, 6], such as K-means, spectral clustering [35, 48], and subspace clustering [31, 17] may fail for two main issues: first, hand-crafted features have limited capacity and cannot dynamically adjust to capture the prior distribution, especially when dealing with large-scale real-world images; second, the separation of feature extraction and clustering will make the solution sub-optimal. Recently, with the booming of deep learning [28, 21, 46, 30, 51], many researchers shift their attention to deep unsupervised feature learning and clustering [42, 23, 8], which can well solve the aforementioned limitations. Typically, to learn a better representation, [3, 50, 52] adopt the auto-encoder and [22] maximizes the mutual information between features. DAC [8] constructs

1

positive and negative pairs to guide network training.

However, for these methods, several points are still missing. Firstly, feature representations that only consider reconstruction or mutual information lack discriminative power. Secondly, traditional cluster method like K-means effectively use category assumption on data. Contrast to that, DAC only focuses on pair-wise correlation and neglects the category information, which limits its performance. Thirdly, there are also other correlations that are helpful for deep image feature learning, for example, [29] shows that measuring feature equivariance can benefit image representation understanding.

To tackle above issues, as shown in Figure 1(a), we propose a novel method, namely deep comprehensive correlation mining (DCCM), which comprehensively explores correlations among different samples (red line), local robustness to geometry transformation (yellow line), between different layer features of the same sample (blue line), and their inter-correlations (green lines) to learn discriminative representations and train the network in a progressive manner. First of all, for the correlation among different samples, we adopt the deep convolutional neural network (CNN) to generate prediction feature for the input image. With proper constraints, the learned prediction feature will tend to be one-hot. Then we can compute the cosine similarity and construct the similarity graph. Based on the similarity graph and prediction feature, we assign a large threshold to get highly-confident pseudo-graph and pseudo-label to guide the feature learning. Secondly, for the local robustness to small perturbations, we add small perturbation or transformation on the original input image to generate a transformed image. Under the local robustness assumption, the prediction of the transformed image should be consistent with that of the original image. So we can use the prediction of the original image to guide the feature learning of the transformed image. Thirdly, feature representation of deep layer should preserve distinct information of the input. So we maximize the mutual information between the deep layer feature and shallow layer feature of the same sample. To make the representation more discriminative, we further extend it to a triplet form by incorporating the graph information above. Finally, we combine the loss function of these three different aspects and jointly investigate these correlations in an end-to-end way. Results in Figure 1(c) show the superiority of our method (purple curve) over the state-of-the-art method DAC [8] (red curve).

Our main contributions are summarized as follows:

1) We propose a novel end-to-end deep clustering framework to comprehensively mine various kinds of correlations, and select highly-confident information to train the network in a progressive way;

2) We first derive the rationality of pseudo-label and introduce the highly-confident pseudo-label loss to directly investigate the category information and guide the unsupervised training of deep network;

3) We make use of the local robustness assumption and utilize above pseudo-graph and pseudo-label to learn better representation;

4) We extend the instance-level mutual information to triplet-level, and come up with triplet mutual information loss to learn more discriminative features.

2. Related Work

2.1. Deep Clustering

Existing deep clustering methods [53, 50, 8] mainly aim to combine the deep feature learning [3, 45, 54] with traditional clustering methods [55, 19, 6]. Auto-encoder (AE) [3] is a very popular feature learning method for deep clustering, and many methods are proposed to minimize the loss of traditional clustering methods to regularize the learning of latent representation of auto-encoder. For example, [50, 20] proposes the deep embedding clustering to utilize the KL-divergence loss. [11] also uses the KL-divergence loss, but adds a noisy encoder to learn more robust representation. [52] adopts the K-means loss, and [23, 42, 41] incorporate the self-representation based subspace clustering loss.

Besides the auto-encoder, some methods directly design specific loss function based on the last layer output. [53] introduces a recurrent-agglomerative framework to merge clusters that are close to each other. [8] explores the correlation among different samples based on the label features, and uses such similarity as supervision. [44] extends the spectral clustering into deep formulation.

2.2. Deep Unsupervised Feature Learning

Instead of clustering, several approaches [3, 25, 34, 14, 39, 2, 47, 49] mainly focus on deep unsupervised learning of representations. Based on Generative Adversarial Networks (GAN), [13] proposes to add an encoder to extract visual features. [4] directly uses the fixed targets which are uniformly sampled from a unit sphere to constrain the deep features assignment. [7] utilizes the pseudo-label computed by the K-means on output features as supervision to train the deep neural networks. [22] proposes the deep infomax to maximize the mutual information between the input and output of a deep neural network encoder.

2.3. Self-supervised Learning

Self-supervised learning [24, 26] generally needs to design a pretext task, where a target objective can be computed without supervision. They assume that the learned representations of the pretext task contain high-level semantic information that is useful for solving downstream tasks of interest, such as image classification. For example, [12]

tries to predict the relative location of image patches, and [36, 37] predict the permutation of a jigsaw puzzle created from the full image. [15] regards each image as an individual class and generates multiple images of it by data augmentation to train the network. [18] rotates an image randomly by one of four different angles and lets the deep model predict the rotation.

3. Deep Comprehensive Correlation Mining

Without labels, correlation stands in the most important place in deep clustering. In this section, we first construct pseudo-graph to explore binary correlation between samples to start the network training. Then we propose the pseudo-label loss to make full use of category information behind the data. Next, we mine the local robustness of predictions before and after adding transform on input image. We also lift the instance level mutual information to triplet level to make it more discriminative. Finally, we combine them together to get our proposed method.

3.1. Preliminary: Pseudo-graph Supervision

We first compute the similarity among samples and select highly-confident pair-wise information to guide the network training by constructing pseudo-graph. Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ be the unlabeled dataset, where $\mathbf{x}_i$ is the i-th image and N is the total number of images. Denote K as the total number of classes. We aim to learn a deep CNN based mapping function f which is parameterized by θ. Then we can use $\mathbf{z}_i = f_\theta(\mathbf{x}_i) \in \mathbb{R}^K$ to represent the prediction feature of image $\mathbf{x}_i$ after the softmax layer of CNN. It has the following properties:

$$\sum_{t=1}^K z_{it} = 1, \forall i = 1, \cdots, N, \text{ and } z_{it} \geq 0, \forall t = 1, \cdots, K. \quad (1)$$

Based on the label feature $\mathbf{z}$, the cosine similarity between the i-th and the j-th samples can be computed by $S_{ij} = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\|\mathbf{z}_i\|_2 \|\mathbf{z}_j\|_2}$, where $\cdot$ is the dot production of two vectors. Similar to DAC [8], we can construct the pseudo-graph $\mathbf{W}$ by setting a large threshold $thres_1$:

$$W_{ij} = \begin{cases} 1, & \text{if } s_{ij} \geq thres_1, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

If the similarity between two samples is larger than the threshold, then we judge that these two samples belong to the same class ($W_{ij} = 1$), and the similarity of these samples should be maximized. Otherwise ($W_{ij} = 0$), the similarity of these samples should be minimized. The pseudo-graph supervision can be defined by:[2]

$$\min_\theta L_{PG}(\theta) = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}} \ell_g(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}_j); W_{ij}). \quad (3)$$

---

[2]For the loss function ℓ_g, there are many choices, such as the contrastive Siamese net loss [5, 32] regularizing the distance between two samples, and the binary cross-entropy loss [8] regularizing the similarity.

Please note that there are two differences between our pseudo-graph and that in DAC [8]: 1) Unlike the strong ℓ_2-norm constrain in DAC, we relax this assumption which only needs to take the output after softmax layer. This relaxation increases the capacity of labeling feature and finally induces a better result in our experiment. 2) Instead of dynamically decreasing threshold in DAC, we only need a fixed threshold of $thres_1$. This prevents the training from the disadvantage caused by noisy false positive pairs.

3.2. Pseudo-label Supervision

The correlation explored in pseudo-graph is not transitive and limited to pair-wise samples. Towards this issue, in this subsection, we propose the novel pseudo-label loss and prove its rationality. We first prove the existence of K-partition of the pseudo-graph, which could be naturally regarded as pseudo-label. And then we state that this partition would make the optimal solution θ^* in Eq. (3) lead to one-hot prediction, which formulates the pseudo-label. Finally, the pseudo-label loss will be introduced to optimize convolutional neural networks.

Existence of K-partition. The binary relation W_{ij} between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ defined in Eq. (3) is not transitive: W_{ij} is not deterministic given W_{ik} and W_{jk}, and this may lead to unstability in training. Therefore, we introduce Lemma (1) to extend it to a stronger relation.

Lemma 1. *For any weighted complete graph $G = (V, E)$ with weight $\omega(e)$ for edge e, if $\omega(e_i) \neq \omega(e_j)$ for $\forall i \neq j$, then there exists a threshold t that $G_t = (V, E_t)$ has exactly K partitions, where*

$$E_t = \{e_i | \omega(e) > t, e_i \in E\}. \quad (4)$$

If we take the assumption that S_{ij} is distinctive to each other in similarity graph $\mathbf{S}$, it can be seen as a weighted complete graph under the assumption of Lemma (1). Then there exists a threshold t dividing $\mathcal{X}$ into exactly K partitions $\{P^1, P^2, \cdots, P^K\}$.

Formulation of the Pseudo-label. Let $\mathbf{x}^k$ denote the sample belongs to partition P^k, and we can define a transitive relation δ as:

$$\delta(\mathbf{x}_i^l, \mathbf{x}_j^k) = \begin{cases} 1, & \text{if } l = k, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

which indicates that pairs with high cosine similarity are guaranteed to be in the same partition. This is to say, as the quality of similarity matrix $\mathbf{S}$ increases during training, this partition gets closer to the ground truth partition, therefore can be regarded as a target to guide and speed up training. Hence, we set the partition k of each $\mathbf{x}$ as its pseudo label.

The following claim reveals the relationship between the assigned pseudo-label and the prediction after softmax:
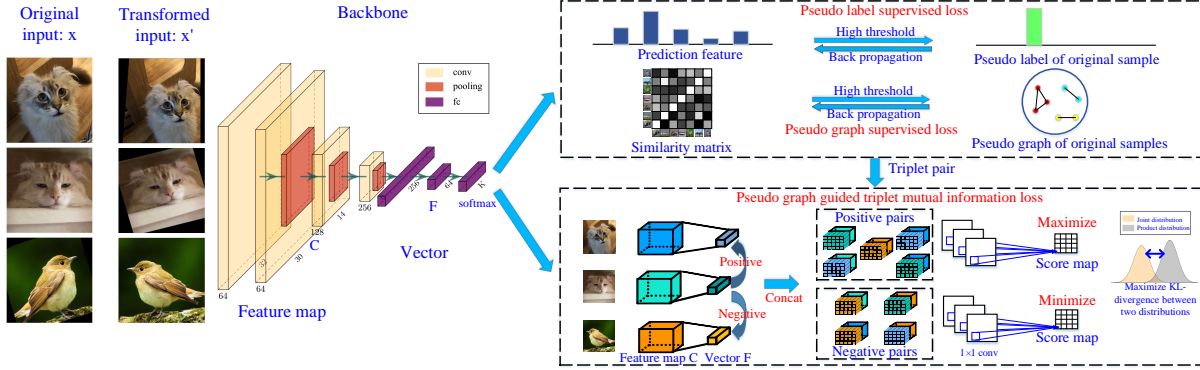
Figure 2. The pipeline of the proposed DCCM method. Based on the ideally one-hot prediction feature, we compute the highly-confident pseudo-graph and pseudo-label to guide the feature learning of both original and transformed samples, investigating both correlations among different samples and local robustness after small perturbation. Meanwhile, to investigate discriminative feature correspondence, the pseudo-graph is utilized to select highly-confident positive and negative pairs for triplet mutual information optimization.

Claim 1. [3] *Let θ^* denote the optimal solution to Eq. (3). If* $\mathbf{W}$ *has K partitions, then the prediction would be one-hot:*

$$f_{\theta^*}(\mathbf{x}) = (0, \cdots, 0, 1, 0, \cdots, 0), \quad for \ \forall \mathbf{x}. \qquad (6)$$

Hence we can formulate our pseudo-label as:

$$y_i = \arg\max_k \ [f_\theta(\mathbf{x}_i)]_k, \qquad (7)$$

where $[\cdot]_k$ denotes the k-th component of the prediction vector. Its corresponding probability of the predicted pseudo-label can be computed by $p_i = \max \ [f_\theta(\mathbf{x}_i)]_k$. In practice, $f_\theta(\mathbf{x}_i)$ does not strictly follow the one-hot property, since it is difficult to attain the optimal solution for the problem in Eq. (3) due to the non-convex property. So we also set a large threshold $thres_2$ for probability p_i to select highly-confident pseudo-label for supervision:

$$V_i = \begin{cases} 1, & \text{if } p_i \geq thres_2, \\ 0, & \text{otherwise.} \end{cases} \qquad (8)$$

$V_i = 1$ indicates the predicted pseudo-label is highly-confident, and only under this situation, will the pseudo-label y_i of the i-th samples join the network training.

Pseudo-label Loss. The pseudo-label supervision loss is formulated as:

$$L_{PL}(\theta) = \sum_{\mathbf{x}_i \in \boldsymbol{\mathcal{X}}} V_i \cdot \ell_l \left(f_\theta(\mathbf{x}_i), y_i \right). \qquad (9)$$

The loss function ℓ_l is often defined by the cross-entropy loss. By combining the supervision of highly-confident pseudo-graph and pseudo-label, we explore the correlation among different samples by minimizing:

$$L_{CDS} = L_{PG}(\theta) + \alpha L_{PL}(\theta), \qquad (10)$$

where α is a balance parameter. Those selected highly-confident information can supervise the training of deep network in a progressive manner.

---

[3]The proof is presented in supplementary materials.

3.3. The Local Robustness

An ideal image representation should be invariant to the geometry transformation, which can be regarded as the local robustness assumption. Mathematically, given an image sample $\mathbf{x}$ and a geometry transformation $\mathbf{G}$, we denote $\mathbf{x}' = \mathbf{G} \cdot \mathbf{x}$ as the transformed sample, then a good feature extractor f_θ should satisfy that these two samples have the same label and $f_\theta(\mathbf{x}) \approx f_\theta(\mathbf{x}')$. Thus we can incorporate the distance between $f_\theta(\mathbf{x})$ and $f_\theta(\mathbf{x}')$ as a feature invariant loss as:

$$\min_\theta \sum_{i=1}^{N} \ell_r \left(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}'_i) \right), \qquad (11)$$

where ℓ_r is the ℓ_2-norm to measure the distance between predictions of original and transformed samples. $\mathbf{x}$ and $\mathbf{G} \cdot \mathbf{x}$ generated by the transformation can be regarded as the 'easy' positive pair, which can well stabilize the training and boost the performance.

Moreover, please recall that for the original samples, we compute the pseudo-graph and pseudo-label as supervision. Instead of simply minimizing the distance of predictions, we hope the graph and label information computed based on transformed samples should be consistent with those of original samples. On the one hand, given an image $\mathbf{x}_i$ with highly-confident pseudo-label y_i, we also force $\mathbf{x}'_i$ has same pseudo-label. On the other hand, we also investigate the correlation among the transformed samples $\mathbf{x}'$ with the highly-confident pseudo-graph $\mathbf{W}$ computed on the original samples $\mathbf{x}_i$, which is beneficial to increase the network robustness. The loss function to achieve above targets can be formulated as:

$$L_{LR} = \sum_{\mathbf{x}'_i, \mathbf{x}'_j \in \boldsymbol{\mathcal{X}}'} \ell_g(f_\theta(\mathbf{x}'_i), f_\theta(\mathbf{x}'_j); W_{ij}) + \alpha \sum_{\mathbf{x}'_i \in \boldsymbol{\mathcal{X}}'} V_i \cdot \ell_l \left(f_\theta(\mathbf{x}'_i), y_i \right)$$
$$= L'_{PG}(\theta) + \alpha L'_{PL}(\theta), \qquad (12)$$

where $\boldsymbol{\mathcal{X}}' = \{\mathbf{x}'_i\}_{i=1}^{N}$ is the transformed data set, $\mathbf{W}$ and $\mathbf{V}$

are same to those of original set in Eqs. (2) and (8).

The deep unsupervised learning can benefit a lot from the above strategy. As we set high confidence for the construction of pseudo-graph and pseudo-label, it can be regarded as the easy sample, which will contribute little to the parameter learning [16]. By adding small perturbation, the prediction of transformed sample will not be easy as that of original sample, which will contribute a lot in return.

3.4. Triplet Mutual Information

In this section, we explore the correlation between deep and shallow layer representations of each instance and propose a novel loss, named triplet mutual information loss, to make full use of the feature correspondence information. Firstly, we introduce the mutual information loss which is proposed in [38, 22] and analyze its limitation. Next, the concept of triplet correlations is described. Finally, we propose the triplet mutual information loss that enables convolutional neural networks to learn discriminative features.

The mutual information (MI) between deep and shallow layer features of the same sample should be maximized, which guarantees the consistency of representation. Similar to [38], we also convert the MI of two random variables (D and S) to the Jensen-Shannon divergence (JSD) between samples coming from the joint distribution $\mathbb{J}$ and their product of marginals $\mathbb{M}$. Correspondingly, features of different layers should follow the joint distribution only when they are features of the same sample, otherwise, they follow the marginal product distribution. So JSD version MI is defined as:

$$\mathcal{MI}^{(JSD)}(D, S) = \mathbb{E}_{\mathbb{J}}[-\mathrm{sp}(-T(d, s))] - \mathbb{E}_{\mathbb{M}}[\mathrm{sp}(T(d, s))], \quad (13)$$

where d corresponds to the deep layer features, s corresponds to the shallow layer features, T is a discriminator trained to distinguish whether d and s are sampled from the joint distribution or not, and $\mathrm{sp}(z) = \log(1 + e^z)$ is the softplus function. For discriminator implementation, [22] shows that incorporating knowledge about locality in the input can improve the representations' quality.

Please note that currently, we do not incorporate any class information. For two different samples $\mathbf{x}_1$ and $\mathbf{x}_2$, the mutual information between $\mathbf{x}_1$'s shallow-layer representation and $\mathbf{x}_2$'s deep-layer representation will be minimized even if they belong to the same class, which is not reasonable. So we consider fixing this issue by introducing the mutual information loss of positive pairs. As shown in the bottom right of Figure 2, with the generated pseudo-graph $\mathbf{W}$ described in Section 3.1, we select positive pairs and negative pairs with the same anchor to construct triplet correlations. Analogous to supervised learning, this approach lifts the instance-level mutual information supervision to triplet-level supervision.

Algorithm 1 Deep Comprehensive Correlation Mining

Input: Unlabeled dataset $\boldsymbol{\mathcal{X}} = \{\mathbf{x}_i\}_{i=1}^N$, $thres_1$, $thres_2$.
1: Initialize the network parameter θ randomly;
2: **for** t in $[1, num\_epoches]$ **do**
3:      **for** each minibatch $\boldsymbol{\mathcal{X}_B}$ **do**
4:          Compute the prediction feature $f(\mathbf{x}_i)$ for each sample $\mathbf{x}_i$ in the minibatch set $\boldsymbol{\mathcal{X}_B}$;
5:          Compute the similarity s_{ij}, pseudo-graph $\mathbf{W}$ and pseudo-label based on Eqs. (2), (7) and (8);
6:          Select positive and negative pairs based on $\mathbf{W}$;
7:          Compute the DCCM loss by Eq. (15);
8:          Update θ using optimizers;
9:      **end for**
10: **end for**
Output: Compute the cluster label by Eq. (7).

Then we show how this approach is theoretically formulated by extending Eq. (13). We set the samples of random variable D and S to be sets, instead of instances. Denote the deep layer feature of sample j belongs to class i as d_j^i and its shallow layer feature as s_j^i, then $D^i = \{d_1^i, d_2^i, \cdots, d_n^i\}$ and $S^i = \{s_1^i, s_2^i, \cdots, s_n^i\}$ are feature sets of class i. Variables $\mathbb{D}$ and $\mathbb{S}$ are defined by $\mathbb{D} = \{D^1, D^2, \cdots, D^K\}$ and $\mathbb{S} = \{S^1, S^2, \cdots, S^K\}$, respectively. Then we can get the following extension of Eq. (13):

$$L_{MI} = -\mathcal{MI}_{set}^{(JSD)}(D, S) = -\left(\mathbb{E}_{(\mathbb{D},\mathbb{S})=\mathbb{J}}[-\mathrm{sp}(-T(d, s))] - \mathbb{E}_{\mathbb{D} \times \mathbb{S}=\mathbb{M}}[\mathrm{sp}(T(d, s))]\right), \quad (14)$$

where we investigate the mutual information based on class-related feature sets. In this case, besides considering the features of same sample, we also maximize the mutual information between different layers' features for samples belongs to the same class. The overview of triplet mutual information loss is shown in the bottom right of Figure 2. Specifically, we compute the loss function in Eq. (14) by pair-wise sampling. For each sample, we construct the positive pairs and negative pairs based on the pseudo-graph $\mathbf{W}$ to compute the triplet mutual information loss, which is very helpful to learn more discriminative representations.

3.5. The Unified Model and Optimization

By combining the investigations of these three aspects in above subsections and jointly train the network, we come up with our deep comprehensive correlation mining for unsupervised learning and clustering. The final objective function of DCCM can be formulated as:

$$\min_{\theta} L_{DCCM} = \widehat{L_{PG}} + \alpha \widehat{L_{PL}} + \beta L_{MI}, \quad (15)$$

where α and β are constants to balance the contributions of different terms, $\widehat{L_{PG}} = L_{PG} + L'_{PG}$ is the overall

Table 1. Statistics of different datasets.

Dataset	Train Images	Test Images	Clusters	Image size
CIFAR-10	50,000	10,000	10	$32 \times 32 \times 3$
CIFAR-100	50,000	10,000	20/100	$32 \times 32 \times 3$
STL-10	13,000	–	10	$96 \times 96 \times 3$
ImageNet-10	13,000	–	10	$96 \times 96 \times 3$
ImageNet-dog-15	19,500	–	15	$96 \times 96 \times 3$
Tiny-ImageNet	100,000	–	200	$64 \times 64 \times 3$

pseudo-graph loss, and $\widehat{L_{PL}} = L_{PL} + L'_{PL}$ is the overall pseudo-label loss. The framework of DCCM is presented in Figure 2. Based on the ideally one-hot prediction feature, we compute the highly-confident pseudo-graph and pseudo-label to guide the feature learning of both original and transformed samples, investigating both correlations among different samples and local robustness for small perturbation. In the meantime, to investigate feature correspondence for discriminative feature learning, the pseudo-graph is also utilized to select highly-confident positive and negative pairs for triplet mutual information optimization.

Our proposed method can be trained in a minibatch based end-to-end way, which can be optimized efficiently. After the training, the predicted feature is ideally one-hot. The predicted cluster label for sample $\mathbf{x}_i$ is exactly same to the pseudo-label y_i, which is easily computed by Eq. (7). We summarize the overall training process in Algorithm 1.

4. Experiments

We distribute our experiments into a few sections. We first examine the effectiveness of DCCM by comparing it against other state-of-the-art algorithms. After that, we conduct more ablation studies by controlling several influence factors. Finally, we do a series of analysis experiments to verify the effectiveness of the unified model training framework. Next, we introduce the experimental setting.

Datasets. We select six challenging image datasets for deep unsupervised learning and clustering, including the CIFAR-10 [27], CIFAR-100 [27], STL-10 [9], Imagenet-10, and ImageNet-dog-15, and Tiny-ImageNet [10] datasets. We summarize the statistics of these datasets in Table 1.

For the clustering task, we adopt the same setting as that in [8], where the training and validation images of each dataset are jointly utilized, and the 20 superclasses are considered for the CIFAR-100 dataset in experiments. ImageNet-10 and ImageNet-dog-15 used in our experiments are same as [8], where they randomly choose 10 subjects and 15 kinds of dog images from the ImageNet dataset, and resize these images to $96 \times 96 \times 3$. As for the Tiny-ImageNet dataset, a reduced version of the ImageNet dataset [10], it totally contains 200 classes of 110,000 images, which is a very challenging dataset for clustering.

For the transfer learning classification task, we adopt the similar setting as that in [22], where we mainly consider the

CIFAR-10, CIFAR-100 of 100 classes. Training and testing samples are separated.

Evaluation Metrics. To evaluate the performance of clustering, we adopt three commonly used metrics including normalized mutual information (NMI), accuracy (ACC), adjusted rand index (ARI). These three metrics favour different properties in clustering task. For details, please refer to the appendix. For all three metrics, the higher value indicates the better performance.

To evaluate the quality of feature representation, we adopt the non-linear classification task which is the same as that in [22]. Specifically, after the training of DCCM, we fix the parameter of deep neural network and train a multilayer perception network with a single hidden layer (200 units) on top of the last convolutional layer and fully-connected layer features separately in a supervised way.

Implementation Details. The network architecture used in our framework is a shallow version of the AlexNet (details for different datasets are described in the supplementary materials). Similar to [8], we adopt the RMSprop optimizer with $lr = 1e^{-4}$. For hyper-parameters, we set $\alpha = 5$ and $\beta = 0.1$ for all datasets, which are relatively stable within a certain range. The thresholds to construct highly-confident pseudo-graph and select highly-confident pseudo-label are set to 0.95 and 0.9, respectively. The small perturbations used in the experiments include rotation, shift, rescale, etc. For discriminator of mutual information estimation, we adopt the network with three 1×1 convolutional layers, which is same to [22]. We use pytorch [40] to implement our approach.

4.1. Main Results

We first compare the DCCM with other state-of-the-art clustering methods on the clustering task. The results are shown in the Table 2. Most results of other methods are directly copied from DAC [8]. DCCM significantly surpasses other methods by a large margin on these benchmarks under all three evaluation metrics. Concretely, the improvement of DCCM is very significant even compared with the state-of-the-art method DAC [8]. Take the clustering ACC for example, our result 0.623 is 10.1% higher than the performance 0.522 of DAC [8] on the CIFAR-10 dataset. On the CIFAR-100 dataset, the gain of DCCM is 8.9% over DAC [8].

Figure 3 visualizes feature embeddings of the DCCM and DAC on CIFAR-10 using t-SNE [33]. We can see that compared with DAC, DCCM exhibits more discriminative feature representation. Above results can sufficiently verify the effectiveness and superiority of our proposed DCCM.

To further evaluate the quality of feature representations, we adopt the classification task and compare DCCM with other deep unsupervised feature learning methods. We compare DCCM against several unsupervised feature learning

Table 2. Clustering performance of different methods on six challenging datasets. The best results are highlighted in **bold**.

Datasets	CIFAR-10			CIFAR-100			STL-10			ImageNet-10			Imagenet-dog-15			Tiny-ImageNet		
Methods\Metrics	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 [55]	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 [19]	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 [6]	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 [3]	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 [45]	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 [43]	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 [54]	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 [25]	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 [53]	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 [50]	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 [8]	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 (ours)	**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**

(a) Initial stage of DCCM   (b) Middle stage of DCCM   (c) Final stage of DCCM   (d) Final stage of DAC
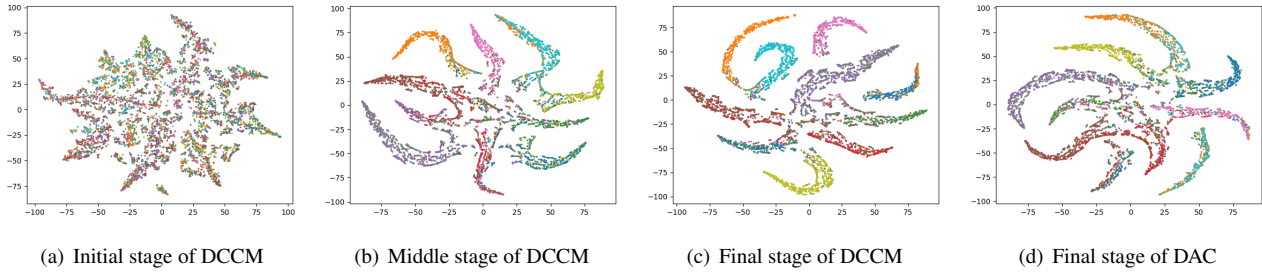
Figure 3. Visualizations of embeddings for different stages of DCCM and DAC on the CIFAR-10 dataset. Different colors denote various clusters. From (a) to (c), with the increasing of epochs, DCCM tends to progressively learn more discriminative features. Based on (c) and (d), features of DCCM are more discriminative than that of DAC.
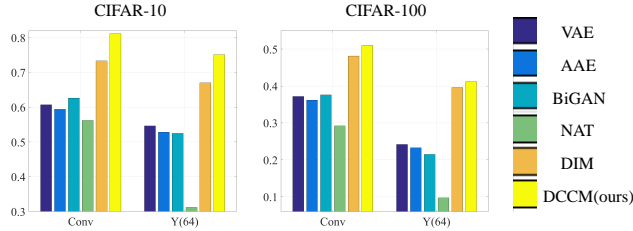

Figure 4. Non-linear classification accuracy (top 1) results of different deep unsupervised feature learning methods on two datasets. 'Conv' denotes the features after the last convolutional layer, and 'Y(64)' denotes the 64-dimensional feature of fully-connected layer.

methods, including variational AE (VAE) [25], adversarial AE (AAE) [34], BiGAN [13], noise as targets (NAT) [4], and deep infomax (DIM) [22]. The top 1 non-linear classification accuracy comparison is presented in Figure 4. We can also observe that DCCM achieves much better results than other methods on CIFAR-10 and CIFAR-100 datasets. Especially on the CIFAR-10 dataset, our results on both convolutional and fully-connected layer features are more than 8% higher than these of the second best method DIM. Since we incorporate the graph-based class information and transform the instance-level mutual information into the triplet-level, our method can learn much more discriminative features, which accounts for the obvious improvement.

Table 3. Ablation study of DCCM on the CIFAR-10 dataset. LR, PL, and MI corresponds to local robustness, pseudo-label, and mutual information, respectively.

	Methods	Correlations			Metrics		
		LR	PL	MI	NMI	ACC	ARI
M1	L_{PG}				0.304	0.405	0.232
M2	$\widehat{L_{PG}}$	✓			0.412	0.512	0.323
M3	$\widehat{L_{PG}} + \widehat{L_{PL}}$	✓	✓		0.448	0.583	0.358
M4	$\widehat{L_{PG}} + \widehat{L_{PL}} + L_{MI}$	✓	✓	✓	**0.496**	**0.623**	**0.408**

We also compare with several state-of-the-art methods under the same architecture and analyze the influence of various sampling strategy in the supplementary materials.

4.2. Correlation Analysis

We analyze the effectiveness of various correlations from three aspects: Local Robustness, Pseudo-label and Triplet Mutual Information in this section. The results are shown in Table 3.

Local Robustness Influence. The only difference between methods M2 and M1 lies in whether to use the local robustness mechanism or not. We can see that M2 significantly surpasses the M1, which demonstrates the robustness and effectiveness of local robustness. Because we set high threshold to select positive pairs, without transforma-
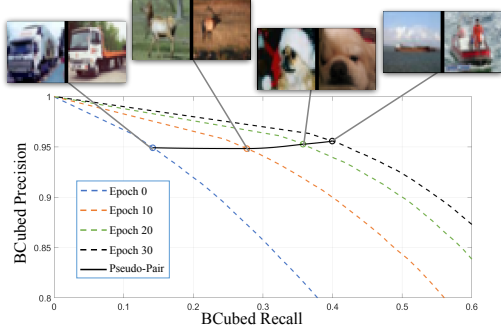
Figure 5. BCubed precision and recall curves [1] for the pseudo-graphs of various epochs on CIFAR-10. These circle points on the lines correspond to the fixed pseudo-graph threshold 0.95 in experiments.

tion, these easy pairs have limited contribution to parameter learning. With the local robustness loss, we construct many hard sample pairs to benefit the network training. So it significantly boosts the performance.

Effectiveness of Pseudo-label. With the help of pseudo-label, M3 (with both pseudo-graph and pseudo-label) achieves much better results than M2 (with only pseudo-graph) under all metrics. Specifically, there is a 7.1% improvement on clustering ACC. The reason is that pseudo-label can make full use of the category information behind the feature distribution, which can benefit the clustering.

Triplet Mutual Information Analysis. Comparing the results of M4 and M3, we can see that the triplet mutual information can further improve the clustering ACC by 4.0%. As we analyzed in Section 3.4, with the help of pseudo-graph, triplet mutual information can not only make use of the features correspondence of the same sample, but also introduce discriminative property by constructing positive and negative pairs. So it can further improve the result.

4.3. Overall Study of DCCM

In this section, we conducted experiments on CIFAR-10 [27] to investigate the behavior of deep comprehensive correlations mining. The model is trained with the unified model optimization which is introduced in Section 3.5.

BCubed Precision and Recall of Pseudo-graph. BCubed [1] is a metric to evaluate the quality of partitions in clustering. We validate that our method can learn better representation in a progressive manner by using the BCubed [1] precision and recall curves, which are computed based on the pseudo-graphs of different epochs in Figure 5. It is obvious that with the increasing of epochs, the precision of the pseudo-graph becomes much better, which will improve the clustering performance in return.

Statistics of Prediction Features. According to Claim 1, the ideal prediction features have the one-hot property, so that we can use the highly-confident pseudo-label to guide the training. To verify it, we compare the distribution of the
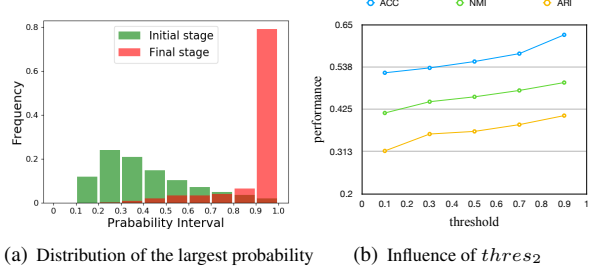
(a) Distribution of the largest probability   (b) Influence of $thres_2$

Figure 6. The distribution of the largest probability in all prediction features and the influence of threshold for highly-confident pseudo-label on the CIFAR-10 dataset.

largest prediction probability between the initial stage and the final stage. The results on the CIFAR-10 dataset is presented in Figure 6(a). For the CIFAR-10 dataset, the largest probability p is in the range of $[0.1, 1]$. We count the probability in nine disjoint intervals, such as $[0.1, 0.2]$, $[0.2, 0.3]$, $\cdots$, and $[0.9, 1]$. We can see that in the initial stage, less than 10% of all samples have the probability that is larger than 0.7, while after training, nearly 80% of all samples have the probability that is larger than 0.9. The above results imply that the largest probability tends to be 1, and others tend to be 0, which is consistent with our Claim 1.

Influence of Thresholds. In Figure 6, we test the influence of threshold to select highly-confident pseudo-label for training. We can see that with the increase of threshold, the performance also increases. The reason is that with low threshold, some incorrect pseudo-label will be adopted for network training, which will affect the performance. So it is important to set relatively high threshold to select highly-confident pseudo-label for supervision.

5. Conclusions

For deep unsupervised learning and clustering, we propose the DCCM to learn discriminative feature representation by mining comprehensive correlations. Besides the correlation among different samples, we also make full use of the mutual information between corresponding features, local robustness to small perturbations, and their intercorrelations. We conduct extensive experiments on several challenging datasets and two different tasks to thoroughly evaluate the performance. DCCM achieves significant improvement over the state-of-the-art methods.

References

[1] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

[2] M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *NIPS*, pages 3846–3854, 2016.

[3] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, pages 153–160, 2007.

[4] P. Bojanowski and A. Joulin. Unsupervised learning by predicting noise. In *ICML*, pages 517–526, 2017.

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a" siamese" time delay neural network. In *NIPS*, pages 737–744, 1994.

[6] D. Cai, X. He, X. Wang, H. Bao, and J. Han. Locality preserving nonnegative matrix factorization. In *IJCAI*, 2009.

[7] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[8] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. In *IEEE ICCV*, pages 5879–5887, 2017.

[9] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223, 2011.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, 2009.

[11] 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.

[12] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *IEEE ICCV*, pages 1422–1430, 2015.

[13] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. In *ICLR*, 2017.

[14] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE TPAMI*, 38(9):1734–1747, 2015.

[15] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, pages 766–774, 2014.

[16] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou. Deep adversarial metric learning. In *IEEE CVPR*, pages 2780–2789, 2018.

[17] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE CVPR*, pages 2790–2797, 2009.

[18] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

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

[20] X. Guo, L. Gao, X. Liu, and J. Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016.

[22] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[23] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid. Deep subspace clustering networks. In *NIPS*, pages 24–33, 2017.

[24] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv preprint arXiv:1902.06162*, 2019.

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

[26] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.

[27] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[29] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *IEEE CVPR*, pages 991–999, 2015.

[30] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, pages 254–269, 2018.

[31] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI*, 35(1):171–184, 2013.

[32] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *IEEE CVPR*, pages 8896–8905, 2018.

[33] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[34] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[35] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2002.

[36] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016.

[37] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *IEEE CVPR*, pages 9359–9367, 2018.

[38] S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, pages 271–279, 2016.

[39] E. Oyallon and S. Mallat. Deep roto-translation scattering for object classification. In *IEEE CVPR*, pages 2865–2873, 2015.

[40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. De-Vito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS Workshop*, 2017.

[41] X. Peng, J. Feng, J. Lu, W.-Y. Yau, and Z. Yi. Cascade sub-space clustering. In *AAAI*, 2017.

[42] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Y. Zhang. Deep subspace clustering with sparsity prior. In *IJCAI*, pages 1925–1931, 2016.

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

[44] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger. Spectralnet: Spectral clustering using deep neural networks. In *ICLR*, 2018.

[45] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[46] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *IEEE CVPR*, pages 3156–3164, 2017.

[47] X. Wang, L. Lu, H.-C. Shin, L. Kim, M. Bagheri, I. Nogues, J. Yao, and R. M. Summers. Unsupervised joint mining of deep features and image labels for large-scale radiology image categorization and scene recognition. In *IEEE WACV*, pages 998–1007, 2017.

[48] J. Wu, Z. Lin, and H. Zha. Essential tensor learning for multi-view spectral clustering. *IEEE Transactions on Image Processing*, 2019.

[49] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE CVPR*, pages 3733–3742, 2018.

[50] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016.

[51] X. Xie, J. Wu, G. Liu, Z. Zhong, and Z. Lin. Differentiable linearized admm. In *ICML*, pages 6902–6911, 2019.

[52] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. Towards K-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, pages 3861–3870, 2017.

[53] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *IEEE CVPR*, pages 5147–5156, 2016.

[54] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *IEEE CVPR*, pages 2528–2535, 2010.

[55] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, pages 1601–1608, 2005.