

# Contrastive Clustering

Yunfan Li<sup>1</sup>, Peng Hu<sup>1</sup>, Zitao Liu<sup>2</sup>, Dezhong Peng<sup>1,4,5</sup>, Joey Tianyi Zhou<sup>3</sup>, Xi Peng<sup>1\*</sup>

<sup>1</sup>College of Computer Science, Sichuan University, China

<sup>2</sup>TAL Education Group, China

<sup>3</sup>Institute of High Performance Computing, A\*STAR, Singapore

<sup>4</sup>Shenzhen Peng Cheng Laboratory, China

<sup>5</sup>College of Computer & Information Science, Southwest University, China

## Abstract

In this paper, we propose an online clustering method called Contrastive Clustering (CC) which explicitly performs the instance- and cluster-level contrastive learning. To be specific, for a given dataset, the positive and negative instance pairs are constructed through data augmentations and then projected into a feature space. Therein, the instance- and cluster-level contrastive learning are respectively conducted in the row and column space by maximizing the similarities of positive pairs while minimizing those of negative ones. Our key observation is that the rows of the feature matrix could be regarded as soft labels of instances, and accordingly the columns could be further regarded as cluster representations. By simultaneously optimizing the instance- and cluster-level contrastive loss, the model jointly learns representations and cluster assignments in an end-to-end manner. Besides, the proposed method could timely compute the cluster assignment for each individual, even when the data is presented in streams. Extensive experimental results show that CC remarkably outperforms 17 competitive clustering methods on six challenging image benchmarks. In particular, CC achieves an NMI of 0.705 (0.431) on the CIFAR-10 (CIFAR-100) dataset, which is an up to 19% (39%) performance improvement compared with the best baseline. The code is available at <https://github.com/XLearning-SCU/2021-AAAI-CC>.

## Introduction

As one of the most fundamental tools in unsupervised learning, clustering could group data into different clusters without any label. Although some promising results have been achieved recently (Nie et al. 2011; Liu et al. 2016; Nie et al. 2016; Liu, Shen, and Tsang 2017; Wang et al. 2020), most of the algorithms would produce undesirable results due to the over-high complexity in real-world datasets. To solve the problem, deep clustering (Guo et al. 2017b; Ghasedi Dizaji et al. 2017; Peng et al. 2016, 2018) utilizes neural networks to extract representative information from images for facilitating the downstream clustering tasks. In very recent, the focus of the community has shifted to how to learn representation and perform clustering in an end-to-end fashion. For example, JULE (Yang, Parikh, and Batra 2016) progressively merges data points and takes the clustering re-

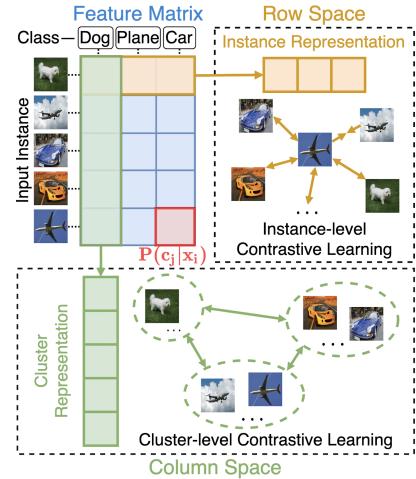


Figure 1: The key observation. By regarding the rows of the feature matrix as the soft labels of instances (*i.e.*,  $P(c_j|x_i)$  denotes the probability of sample  $i$  belonging to cluster  $j$ ), the columns could be interpreted as cluster representations distributed over the dataset. As a result, the instance- and cluster-level contrastive learning could be conducted in the row and column space of the feature matrix, respectively.

sults as supervisory signals to learn a more discriminative representation by a neural network. DeepClustering (Caron et al. 2018) iteratively groups the features with k-means and uses the subsequent assignments to update the deep network. This kind of alternation-learning method would suffer from the error accumulated during the alternation between the stages of representation learning and clustering, which results in suboptimal clustering performance. Moreover, the aforementioned methods can only deal with offline tasks, *i.e.*, the clustering is based on the whole dataset, which limits their application on large-scale online learning scenarios, *i.e.*, to cluster the data stream.

To conquer the aforementioned offline limitation, this paper proposes an online deep clustering method called Contrastive Clustering (CC). Our idea comes from the observations shown in Fig. 1. For a given dataset, we use a deep network to learn the feature matrix whose rows and columns correspond to the instance and cluster representations, respectively. In other words, we treat the label as a special

\*Corresponding author: Xi Peng, email: pengx.gm@gmail.com  
Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

representation by projecting input instances into a subspace with a dimensionality of the cluster number. In this sense, the rows of the feature matrix could be interpreted as the cluster assignment probabilities (*i.e.*, instance soft labels), and the columns could then be regarded as the cluster distributions over instances (*i.e.*, cluster representations). Owing to the observation of “label as representation”, it is feasible to perform online clustering since the clustering prediction is now recast as a special representation learning task that is “independent” of other instances.

With the above observations, we propose a novel dual contrastive learning framework to learn instance and cluster representations. Specifically, CC first learns the feature matrix of data pairs constructed through a variety of data augmentations such as random crop and blurring. After that, the instance- and cluster-level contrastive learning are conducted in the row and column space of the feature matrix by gathering the positive pairs and scattering the negatives. By considering the instance- and cluster-level similarity under our dual contrastive learning framework, CC is able to simultaneously learn discriminative features and perform online clustering in an online and end-to-end manner. To summarize, the major contributions of our work are as follows:

- For the first time, we reveal that the row and column of the feature matrix intrinsically correspond to the instance and cluster representation, respectively. Hence, deep clustering could be elegantly unified into the framework of representation learning;
- To the best of our knowledge, this could be the first work of clustering-specified contrastive learning. Different from existing studies in contrastive learning, the proposed method conducts contrastive learning at not only the instance-level but also the cluster-level. Such a dual contrastive learning framework could produce clustering-favorite representations as proved in our experiments;
- The proposed model works in an online and end-to-end fashion, which only needs batch-wise optimization and thus can be applied to large-scale datasets. Moreover, the proposed method could timely predict the cluster assignment for each new coming data point without accessing the whole dataset, which suits streaming data.

The proposed method shows superior performance on six challenging image datasets, including CIFAR-10/100, STL-10, ImageNet-10/Dogs, and Tiny-ImageNet. It significantly outperforms state-of-the-art methods on all six datasets. In particular, it achieves an up to 39% performance improvement in terms of NMI on the CIFAR-100 dataset compared with the most competitive baseline.

## Related Work

In this section, we briefly introduce some recent developments in two related topics, namely, contrastive learning and deep clustering.

### Contrastive Learning

As a promising paradigm of unsupervised learning, contrastive learning has lately achieved state-of-the-art performance in representation learning (Grill et al. 2020; Li et al.

2020). The basic idea of contrastive learning is to map the original data to a feature space wherein the similarities of positive pairs are maximized while those of negative pairs are minimized (Hadsell, Chopra, and LeCun 2006). In early works, the positive and negative pairs are known as prior. Recently, various works have shown that large quantities of data pairs are crucial to the performance of contrastive models (He et al. 2020) and they could be constructed using the following two strategies under the unsupervised setting. One is to use clustering results as pseudo labels to guide the pair construction (Sharma et al. 2020). The other, which is more direct and commonly used, is to treat each instance as a class represented by a feature vector and data pairs are constructed through data augmentations (Dosovitskiy et al. 2014). To be specific, the positive pair composes of two augmented views of the same instance, and the other pairs are defined to be negative. Given the data pairs, several loss functions have been proposed for contrastive learning. For example, triplet loss (Schroff, Kalenichenko, and Philbin 2015) minimizes the distance between an anchor and a positive, while maximizing the distance between the anchor and a negative, NCE (Gutmann and Hyvärinen 2010) performs nonlinear logistic regression to discriminate between the observed data and some artificially generated noise, and SimCLR (Chen et al. 2020) adopts the normalized temperature-scaled cross-entropy loss (NT-Xent) to identify positive pairs across the dataset.

The differences between our method and existing contrastive learning methods are addressed below. On the one hand, the existing works only perform contrastive learning at the instance level, whereas our method simultaneously conducts contrastive learning at both the instance- and cluster-level following the observation of “label as representation”. On the other hand, the existing works aim to learn a general representation, which is off-the-shelf for the downstream tasks. On the contrary, our method is specifically designed for clustering, which could be the first successful attempt of task-specified contrastive learning.

## Deep Clustering

Although promising results have been achieved, traditional clustering algorithms give discouraging results on large-scale complex datasets due to the inferior capability of representation learning. Benefit from the powerful representative ability of deep neural networks, deep clustering (Xie, Girshick, and Farhadi 2016; Guo et al. 2017a; Li et al. 2020) has shown promising performance on complex datasets. For example, JULE (Yang, Parikh, and Batra 2016) performs agglomerative clustering by iteratively learning the data representations and cluster assignments. Analogously, Deep-Clustering (Caron et al. 2018) groups the features using k-means and updates the deep network according to the cluster assignments in turn. Another recent work SL (Asano, Rupprecht, and Vedaldi 2019) makes cluster assignments by solving the optimal transport problem and alternatively perform representation learning and self-labelling. Though this kind of two-stage methods could jointly learn representations and perform clustering, their performance might be hurt by the errors accumulated during the alternation. Be-

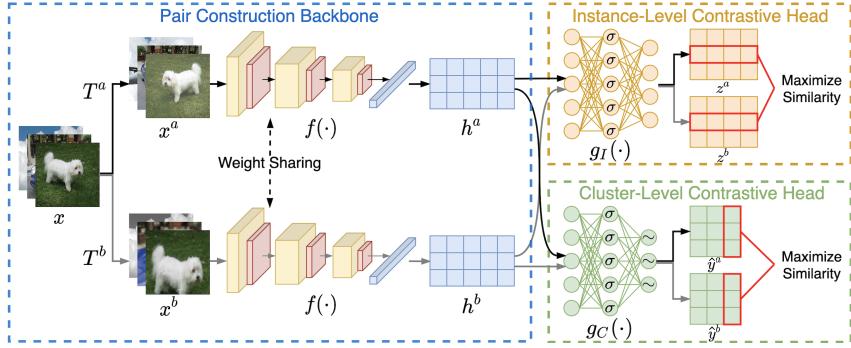


Figure 2: The framework of Contrastive Clustering. We construct data pairs using two data augmentations. Given data pairs, one shared deep neural network is used to extract features from different augmentations. Two separate MLPs ( $\sigma$  denotes the *ReLU* activation and  $\sim$  denotes the *Softmax* operation to produce soft labels) are used to project the features into the row and column space wherein the instance- and cluster-level contrastive learning are conducted respectively.

sides, the entire dataset is usually needed to perform clustering, which limits their application in large-scale and online scenarios. Recently, some online clustering methods have been proposed (Peng, Yi, and Tang 2015; Zhong et al. 2020). For example, IIC (Ji, Henriques, and Vedaldi 2019) discovers clusters by maximizing mutual information between the cluster assignments of data pairs and DHOG (Darlow and Storkey 2020) extends it to a hierarchical manner. PICA (Huang, Gong, and Zhu 2020) learns the most semantically plausible data separation by maximizing the partition confidence of the clustering solution. Though grounded in theory, these works rely heavily on the auxiliary over-clustering trick which is hard to explain.

Different from the above deep clustering methods, we treat the label as a special representation so that the instance- and cluster-level representation learning could be conducted in the row and column space, respectively. Besides, former works mainly utilize the representative capability of deep neural networks for clustering, whereas our method dually utilizes contrastive samples to facilitate clustering under a unified framework. Such a clustering-oriented contrastive learning paradigm helps the model to minimize the inter-cluster similarities to separate different clusters. To the best of our knowledge, this could be one of the first successful attempts to promote clustering through contrastive learning.

## Method

As illustrated in Fig. 2, our method consists of three jointly learned components, namely, a pair construction backbone (PCB), an instance-level contrastive head (ICH), and a cluster-level contrastive head (CCH). In brief, PCB constructs data pairs through data augmentations and extracts features from augmented samples, after that ICH and CCH respectively apply contrastive learning in the row and column space of the feature matrix. After training, the cluster assignments can be easily obtained through the soft labels predicted by CCH. Notably, although our basic idea indicates that the dual contrastive learning could be directly conducted on the feature matrix, we experimentally find that the clustering performance could be improved by decoupling the instance- and cluster-level contrastive learning into

two independent subspaces. The possible reason is that such a decoupling strategy could improve the representability of ICH and CCH. In the following, we will elaborate on the three components in turn and introduce the proposed objective function at the end.

## Pair Construction Backbone

Inspired by the recent progress in contrastive learning (Chen et al. 2020), CC uses data augmentations to construct data pairs. Specifically, given a data instance  $x_i$ , two stochastic data transformations  $T^a, T^b$  sampled from the same family of augmentations  $\mathcal{T}$  are applied to it, resulting in two correlated samples denoted as  $x_i^a = T^a(x_i)$  and  $x_i^b = T^b(x_i)$ . The previous works have suggested that a proper choice of augmentation strategy is essential to achieve a good performance in downstream tasks. In this work, five types of data augmentation methods are used, including ResizedCrop, ColorJitter, Grayscale, HorizontalFlip, and GaussianBlur. For a given image, each augmentation is applied independently with a certain probability following the setting in SimCLR (Chen et al. 2020). Specifically, ResizedCrop crops an image to a random size and resize the crop to the original size; ColorJitter changes the brightness, contrast, and saturation of an image; Grayscale converts an image to grayscale; HorizontalFlip horizontally flip an image and GaussianBlur blurs an image by a Gaussian function.

One shared deep neural network  $f(\cdot)$  is used to extract features from the augmented samples via  $h_i^a = f(x_i^a)$  and  $h_i^b = f(x_i^b)$ . As for the architecture of the network, theoretically, our method does not depend on a specific network. Here, we simply adopt ResNet34 (He et al. 2016) as the backbone for fair comparison.

## Instance-level Contrastive Head

Contrastive learning aims to maximize the similarities of positive pairs while minimizing those of negative ones. The characteristics of pairs can be defined by different criteria. For example, one can define pairs of within-class samples to be positive and leave the others negative. In this work, since no prior label is available on the clustering task, the positive

---

**Algorithm 1:** Contrastive Clustering

---

**Input:** dataset  $\mathcal{X}$ ; training epochs  $E$ ; batch size  $N$ ; temperature parameter  $\tau_I$  and  $\tau_C$ ; cluster number  $M$ ; structure of  $\mathcal{T}$ ,  $f$ ,  $g_I$ , and  $g_C$ .

**Output:** cluster assignments.

```

// training
for epoch = 1 to E do
    sample a mini-batch  $\{x_i\}_{i=1}^N$  from  $\mathcal{X}$ 
    sample two augmentations  $T^a, T^b \sim \mathcal{T}$ 
    compute instance and cluster representations by
         $h_i^a = f(T^a(x_i)), h_i^b = f(T^b(x_i))$ 
         $z_i^a = g_I(h_i^a), z_i^b = g_I(h_i^b)$ 
         $y_i^a = g_C(h_i^a), y_i^b = g_C(h_i^b)$ 
    compute instance-level contrastive loss  $\mathcal{L}_{ins}$ 
        through Eq. 1–3
    compute cluster-level contrastive loss  $\mathcal{L}_{clu}$ 
        through Eq. 4–6
    compute overall loss  $\mathcal{L}$  by Eq. 7
    update  $f, g_I, g_C$  through gradient descent to
        minimize  $\mathcal{L}$ 
end
// test
for  $x$  in  $\mathcal{X}$  do
    extract features by  $h = f(x)$ 
    compute cluster assignment by
         $c = \arg \max g_C(h)$ 
end

```

---

and negative pairs are constructed at the instance-level according to pseudo-labels generated by data augmentations. More specifically, the positive pairs consist of samples augmented from the same instance, and the negative pairs otherwise.

Formally, given a mini-batch of size  $N$ , CC performs two types of data augmentations on each instance  $x_i$  and results in  $2N$  data samples  $\{x_1^a, \dots, x_N^a, x_1^b, \dots, x_N^b\}$ . For a specific sample  $x_i^a$ , there are  $2N - 1$  pairs in total, among which we choose its corresponding augmented sample  $x_i^b$  to form a positive pair  $\{x_i^a, x_i^b\}$  and leave other  $2N - 2$  pairs to be negative.

To alleviate the information loss induced by contrastive loss, we do not directly conduct contrastive learning on the feature matrix. Instead, we stack a two-layer nonlinear MLP  $g_I(\cdot)$  to map the feature matrix to a subspace via  $z_i^a = g_I(h_i^a)$  where the instance-level contrastive loss is applied. The pair-wise similarity is measured by cosine distance, *i.e.*,

$$s(z_i^{k_1}, z_j^{k_2}) = \frac{(z_i^{k_1})(z_j^{k_2})^\top}{\|z_i^{k_1}\| \|z_j^{k_2}\|}, \quad (1)$$

where  $k_1, k_2 \in \{a, b\}$  and  $i, j \in [1, N]$ . To optimize pair-wise similarities, without loss of generality, the loss for a given sample  $x_i^a$  is in the form of

$$\ell_i^a = -\log \frac{\exp(s(z_i^a, z_i^b)/\tau_I)}{\sum_{j=1}^N [\exp(s(z_i^a, z_j^a)/\tau_I) + \exp(s(z_i^a, z_j^b)/\tau_I)]}, \quad (2)$$

where  $\tau_I$  is the instance-level temperature parameter to control the softness. Since we hope to identify all positive pairs across the dataset, the instance-level contrastive loss is computed over every augmented samples, namely,

$$\mathcal{L}_{ins} = \frac{1}{2N} \sum_{i=1}^N (\ell_i^a + \ell_i^b). \quad (3)$$

### Cluster-level Contrastive Head

Following the idea of “label as representation”, when projecting a data sample into a space whose dimensionality equals to the number of clusters, the  $i$ -th element of its feature can be interpreted as its probability of belonging to the  $i$ -th cluster, and the feature vector denotes its soft label accordingly.

Formally, let  $Y^a \in \mathbb{R}^{N \times M}$  be the output of CCH for a mini-batch under the first augmentation (and  $Y^b$  for the second augmentation), and then  $Y_{n,m}^a$  can be interpreted as the probability of sample  $n$  being assigned to cluster  $m$ , where  $N$  is the batch size and  $M$  equals to the number of clusters. Since each sample belongs to only one cluster, ideally, the rows of  $Y^a$  tends to be one-hot. In this sense, the  $i$ -th column of  $Y^a$  can be seen as a representation of the  $i$ -th cluster and all columns should differ from each other.

Similar to  $g_I(\cdot)$  used in the instance-level contrastive head, we use another two-layer MLP  $g_C(\cdot)$  to project the feature matrix into an  $M$ -dimensional space via  $y_i^a = g_C(h_i^a)$ , where  $y_i^a$  denotes the soft label of sample  $x_i^a$  (the  $i$ -th row of  $Y^a$ ). For clarity, let  $\hat{y}_i^a$  be the  $i$ -th column of  $Y^a$ , namely, the representation of cluster  $i$  under the first data augmentation, and we combine it with  $\hat{y}_i^b$  to form a positive cluster pair  $\{\hat{y}_i^a, \hat{y}_i^b\}$ , while leaving other  $2M - 2$  pairs to be negative, where  $\hat{y}_i^b$  denotes the second augmented representation of cluster  $i$ . Again, we use cosine distance to measure the similarity between cluster pairs, that is

$$s(\hat{y}_i^{k_1}, \hat{y}_j^{k_2}) = \frac{(\hat{y}_i^{k_1})^\top (\hat{y}_j^{k_2})}{\|\hat{y}_i^{k_1}\| \|\hat{y}_j^{k_2}\|}, \quad (4)$$

where  $k_1, k_2 \in \{a, b\}$  and  $i, j \in [1, M]$ . Without loss of generality, the following loss function is adopted to distinguish cluster  $\hat{y}_i^a$  from all other clusters except  $\hat{y}_i^b$ , *i.e.*,

$$\hat{\ell}_i^a = -\log \frac{\exp(s(\hat{y}_i^a, \hat{y}_i^b)/\tau_C)}{\sum_{j=1}^M [\exp(s(\hat{y}_i^a, \hat{y}_j^a)/\tau_C) + \exp(s(\hat{y}_i^a, \hat{y}_j^b)/\tau_C)]}, \quad (5)$$

where  $\tau_C$  is the cluster-level temperature parameter to control the softness. By traversing all clusters, the cluster-level contrastive loss is finally computed by

$$\mathcal{L}_{clu} = \frac{1}{2M} \sum_{i=1}^M (\hat{\ell}_i^a + \hat{\ell}_i^b) - H(Y), \quad (6)$$

where  $H(Y) = -\sum_{i=1}^M [P(\hat{y}_i^a) \log P(\hat{y}_i^a) + P(\hat{y}_i^b) \log P(\hat{y}_i^b)]$  is the entropy of cluster assignment probabilities  $P(\hat{y}_i^k) = \frac{1}{N} \sum_{t=1}^N Y_{ti}^k$ ,  $k \in \{a, b\}$  within a mini-batch under each data augmentation. This term helps to avoid the trivial solution that most instances are assigned to the same cluster (Hu et al. 2017).

## Objective Function

The optimization of ICH and CCH is a one-stage and end-to-end process. Two heads are simultaneously optimized and the overall objective function consists of the instance-level and cluster-level contrastive loss, *i.e.*,

$$\mathcal{L} = \mathcal{L}_{ins} + \mathcal{L}_{clu}. \quad (7)$$

Generally, a dynamic weight parameter could be applied to balance the two losses across the training process (Grill et al. 2020), but in practice, we find a simple addition of the two losses already works well. The full training and test process of the model is summarized in Algorithm 1.

## Experiments

**Datasets** We evaluate the proposed method on six challenging image datasets. A brief description of these datasets is summarized in Table 1. Both the training and test set are used for CIFAR-10, CIFAR-100 (Krizhevsky and Hinton 2009), and STL-10 (Coates, Ng, and Lee 2011), while only the training set is used for ImageNet-10, ImageNet-Dogs (Chang et al. 2017a), and Tiny-ImageNet (Le and Yang 2015). For CIFAR-100, its 20 super-classes rather than 100 classes are taken as the ground-truth. For STL-10, its 100,000 unlabeled samples are additionally used to train the instance-level contrastive head.

| Dataset       | Split      | Samples | Classes |
|---------------|------------|---------|---------|
| CIFAR-10      | Train+Test | 60,000  | 10      |
| CIFAR-100     | Train+Test | 60,000  | 20      |
| STL-10        | Train+Test | 13,000  | 10      |
| ImageNet-10   | Train      | 13,000  | 10      |
| ImageNet-Dogs | Train      | 19,500  | 15      |
| Tiny-ImageNet | Train      | 100,000 | 200     |

Table 1: A summary of datasets used for evaluations.

**Implementation Details** For a fair comparison with previous works (Ji, Henriques, and Vedaldi 2019; Huang, Gong, and Zhu 2020), we adopt ResNet34 as the backbone network. As ResNet is designed for images of size  $224 \times 224$ , some previous works modified the standard ResNet and used some tricks (*e.g.*, the Sobel layer used in PICA) to help the network to handle small-sized inputs (*e.g.*, CIFAR-10). However, these specialized modifications and tricks should vary with images of different sizes, which brings difficulty in model selection. In this work, we simply resize all input images to the size of  $224 \times 224$ , and no modification is applied to the standard ResNet which produces a feature vector of size 512 for each sample. Notably, as up-scaling already leads to blurred images, we leave the GaussianBlur augmentation out for the small image collections including CIFAR-10, CIFAR-100, STL-10, and Tiny-ImageNet.

For the instance-level contrastive head, the dimensionality of the row space is set to 128 to keep more information of images, and the instance-level temperature parameter  $\tau_I$  is fixed to 0.5 in all experiments. For the choice of the dimensionality of the row space, we conduct additional analysis in the supplementary material. As for the cluster-level

contrastive head, the dimensionality of the column space is naturally set to the number of clusters, and the cluster-level temperature parameter  $\tau_C = 1.0$  is used for all datasets.

The Adam optimizer with an initial learning rate of 0.0003 is adopted to simultaneously optimize the two contrastive heads and the backbone network. No weight decay or scheduler is used. The batch size is set to 256 due to the memory limitation, and we train the model from scratch for 1,000 epochs to compensate for the performance loss caused by small batch size as suggested by Chen et al. The experiments are carried out on Nvidia TITAN RTX 24G and it takes about 70 gpu-hours to train the model on CIFAR-10, 90 gpu-hours for CIFAR-100, 160 gpu-hours on STL-10, 20 gpu-hours on ImageNet-10, 30 gpu-hours on ImageNet-Dogs, and 130 gpu-hours on Tiny-ImageNet.

**Evaluation Metrics** Three widely-used clustering metrics including Normalized Mutual Information (NMI), Accuracy (ACC), and Adjusted Rand Index (ARI) are utilized to evaluate our method. Higher values of these metrics indicate better clustering performance.

**Comparisons with State of the Arts** We evaluate the proposed CC on six challenging image benchmarks and compare it with 17 representative state-of-the-art clustering approaches, including k-means (MacQueen et al. 1967), SC (Zelnik-Manor and Perona 2005), AC (Gowda and Krishna 1978), NMF (Cai et al. 2009), AE (Bengio et al. 2007), DAE (Vincent et al. 2010), DCGAN (Radford, Metz, and Chintala 2015), DeCNN (Zeiler et al. 2010), VAE (Kingma and Welling 2013), JULE (Yang, Parikh, and Batra 2016), DEC (Xie, Girshick, and Farhadi 2016), DAC (Chang et al. 2017b), ADC (Haeusser et al. 2018), DDC (Chang et al. 2019), DCCM (Wu et al. 2019), IIC (Ji, Henriques, and Vedaldi 2019) and PICA (Huang, Gong, and Zhu 2020). For SC, NMF, AE, DAE, DCGAN, DeCNN, and VAE, clustering results are obtained via k-means on the features extracted from images.

According to the results shown in Table 2, CC significantly outperforms these state-of-the-art baselines by a large margin on all six datasets. In particular, CC surpasses the closest competitor PICA by 0.114 on CIFAR-10, 0.121 on CIFAR-100, and 0.153 on STL-10 in terms of NMI. Moreover, CC achieves more than 50% performance improvements on the best baseline on CIFAR-100 and Tiny-ImageNet in terms of ARI. The remarkable results demonstrate the powerful clustering ability of CC, which benefits from the incorporation of the instance- and cluster-level contrastive learning.

## Qualitative Study

We carry out two experiments to analyze the pair-wise similarity across the training process and the evolution of the learned instance representation and cluster assignments on ImageNet-10.

**Analysis on Pair-wise Similarity** To provide an intuitive understanding of how contrastive clustering works, we visualize the changes of both the instance- and cluster-level pair-wise similarities w.r.t. the training epoch. As shown

| Dataset         | CIFAR-10     |              |              | CIFAR-100    |              |              | STL-10       |              |              | ImageNet-10  |              |              | ImageNet-Dogs |              |              | Tiny-ImageNet |              |              |
|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|--------------|--------------|---------------|--------------|--------------|
| Metrics         | NMI          | ACC          | ARI          | NMI           | ACC          | ARI          | NMI           | ACC          | ARI          |
| K-means         | 0.087        | 0.229        | 0.049        | 0.084        | 0.130        | 0.028        | 0.125        | 0.192        | 0.061        | 0.119        | 0.241        | 0.057        | 0.055         | 0.105        | 0.020        | 0.065         | 0.025        | 0.005        |
| SC              | 0.103        | 0.247        | 0.085        | 0.090        | 0.136        | 0.022        | 0.098        | 0.159        | 0.048        | 0.151        | 0.274        | 0.076        | 0.038         | 0.111        | 0.013        | 0.063         | 0.022        | 0.004        |
| AC              | 0.105        | 0.228        | 0.065        | 0.098        | 0.138        | 0.034        | 0.239        | 0.332        | 0.140        | 0.138        | 0.242        | 0.067        | 0.037         | 0.139        | 0.021        | 0.069         | 0.027        | 0.005        |
| NMF             | 0.081        | 0.190        | 0.034        | 0.079        | 0.118        | 0.026        | 0.096        | 0.180        | 0.046        | 0.132        | 0.230        | 0.065        | 0.044         | 0.118        | 0.016        | 0.072         | 0.029        | 0.005        |
| AE              | 0.239        | 0.314        | 0.169        | 0.100        | 0.165        | 0.048        | 0.250        | 0.303        | 0.161        | 0.210        | 0.317        | 0.152        | 0.104         | 0.185        | 0.073        | 0.131         | 0.041        | 0.007        |
| DAE             | 0.251        | 0.297        | 0.163        | 0.111        | 0.151        | 0.046        | 0.224        | 0.302        | 0.152        | 0.206        | 0.304        | 0.138        | 0.104         | 0.190        | 0.078        | 0.127         | 0.039        | 0.007        |
| DCGAN           | 0.265        | 0.315        | 0.176        | 0.120        | 0.151        | 0.045        | 0.210        | 0.298        | 0.139        | 0.225        | 0.346        | 0.157        | 0.121         | 0.174        | 0.078        | 0.135         | 0.041        | 0.007        |
| DeCNN           | 0.240        | 0.282        | 0.174        | 0.092        | 0.133        | 0.038        | 0.227        | 0.299        | 0.162        | 0.186        | 0.313        | 0.142        | 0.098         | 0.175        | 0.073        | 0.111         | 0.035        | 0.006        |
| VAE             | 0.245        | 0.291        | 0.167        | 0.108        | 0.152        | 0.040        | 0.200        | 0.282        | 0.146        | 0.193        | 0.334        | 0.168        | 0.107         | 0.179        | 0.079        | 0.113         | 0.036        | 0.006        |
| JULE            | 0.192        | 0.272        | 0.138        | 0.103        | 0.137        | 0.033        | 0.182        | 0.277        | 0.164        | 0.175        | 0.300        | 0.138        | 0.054         | 0.138        | 0.028        | 0.102         | 0.033        | 0.006        |
| DEC             | 0.257        | 0.301        | 0.161        | 0.136        | 0.185        | 0.050        | 0.276        | 0.359        | 0.186        | 0.282        | 0.381        | 0.203        | 0.122         | 0.195        | 0.079        | 0.115         | 0.037        | 0.007        |
| DAC             | 0.396        | 0.522        | 0.306        | 0.185        | 0.238        | 0.088        | 0.366        | 0.470        | 0.257        | 0.394        | 0.527        | 0.302        | 0.219         | 0.275        | 0.111        | 0.190         | 0.066        | 0.017        |
| ADC             | —            | 0.325        | —            | —            | 0.160        | —            | —            | 0.530        | —            | —            | —            | —            | —             | —            | —            | —             | —            | —            |
| DDC             | 0.424        | 0.524        | 0.329        | —            | —            | —            | 0.371        | 0.489        | 0.267        | 0.433        | 0.577        | 0.345        | —             | —            | —            | —             | —            | —            |
| DCCM            | 0.496        | 0.623        | 0.408        | 0.285        | 0.327        | 0.173        | 0.376        | 0.482        | 0.262        | 0.608        | 0.710        | 0.555        | 0.321         | 0.383        | 0.182        | 0.224         | 0.108        | 0.038        |
| IIC             | —            | 0.617        | —            | —            | 0.257        | —            | —            | 0.610        | —            | —            | —            | —            | —             | —            | —            | —             | —            | —            |
| PICA            | 0.591        | 0.696        | 0.512        | 0.310        | 0.337        | 0.171        | 0.611        | 0.713        | 0.531        | 0.802        | 0.870        | 0.761        | 0.352         | 0.352        | 0.201        | 0.277         | 0.098        | 0.040        |
| <b>CC(Ours)</b> | <b>0.705</b> | <b>0.790</b> | <b>0.637</b> | <b>0.431</b> | <b>0.429</b> | <b>0.266</b> | <b>0.764</b> | <b>0.850</b> | <b>0.726</b> | <b>0.859</b> | <b>0.893</b> | <b>0.822</b> | <b>0.445</b>  | <b>0.429</b> | <b>0.274</b> | <b>0.340</b>  | <b>0.140</b> | <b>0.071</b> |

Table 2: The clustering performance on six challenging object image benchmarks. The best results are shown in boldface.

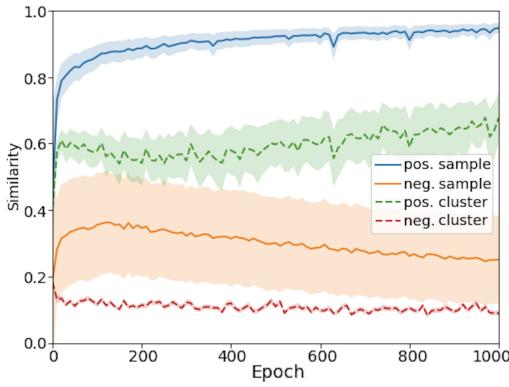


Figure 3: Instance-level and cluster-level pair-wise similarities across the training process on ImageNet-10. The colored areas denote the variances.

In Fig. 3, the similarities of positive instance/cluster pairs grows as the training process goes while the similarity of negative instance/cluster pairs stay at a low level. In addition, the similarity interval between the positive and negative is comparatively large at both the instance- and cluster-level, which explains the success of our model. Note that the variances of positive instance and negative cluster pairs are much lower than those of negative instance pairs and positive cluster pairs due to the following two reasons. On the one hand, the large variance of negative instance pairs could be attributed to the fact that some pairs consist of samples of different instances but the same class, which should be treated as positive theoretically. On the other hand, the variance of positive cluster pairs comes from the inconsistent cluster assignments of samples under different augmentations.

**Evolution of Instance Feature and Cluster Assignments**  
By simultaneously optimizing the instance- and cluster-level contrastive head, the model ought to learn discriminative representations and desirable cluster assignments at the same time. To see how our model converges to the goal, we perform t-SNE in the row space at four different times-

tamps throughout the training process. The results are shown in Fig. 4, where different colors indicated different labels predicted by the cluster-level contrastive head. The result shows that, at the beginning, features are all mixed and most instances are assigned to a few clusters. As the training process goes, cluster assignments become more reasonable, and features scatter and gather more distinctly.

### Ablation Study

Three ablation studies are carried out to further understand the importance of data augmentation, the effect of two contrastive heads, and the reliance on the backbone network.

**Importance of Data Augmentation** Some existing works have shown that the performance of contrastive learning heavily relies on the proper strategy of data augmentation (Chen et al. 2020). To verify the significance of data augmentation, we test our model on CIFAR-10 and ImageNet-10 by removing one and both of the two augmentations. When data augmentations are removed, the raw image is directly used as the input. Table 3 shows that data augmentations could enhance the performance of CC, especially on more complicated datasets, *i.e.*, CIFAR-10. When no data augmentation is applied, every positive pair consists of two same samples/clusters and thus only negative pairs take part in model optimization, which leads to pretty poor results.

| Dataset     | Augmentation      | NMI          | ACC          | ARI          |
|-------------|-------------------|--------------|--------------|--------------|
| CIFAR-10    | $T^a(x) + T^b(x)$ | <b>0.705</b> | <b>0.790</b> | <b>0.637</b> |
|             | $T^a(x) + x$      | 0.630        | 0.690        | 0.533        |
|             | $x + x$           | 0.045        | 0.169        | 0.022        |
| ImageNet-10 | $T^a(x) + T^b(x)$ | <b>0.859</b> | <b>0.893</b> | <b>0.822</b> |
|             | $T^a(x) + x$      | 0.852        | 0.892        | 0.817        |
|             | $x + x$           | 0.063        | 0.177        | 0.030        |

Table 3: Importance of data augmentation.

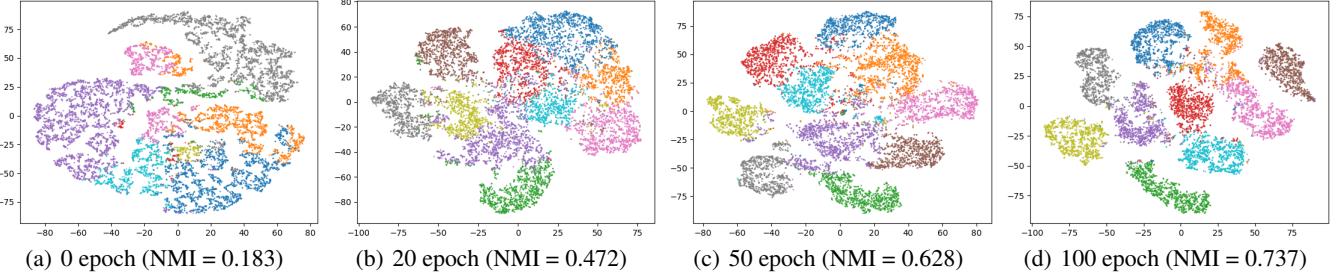


Figure 4: The evolution of instance features and cluster assignments across the training process on ImageNet-10. The colors indicate the cluster assignment obtained from CCH and the features for t-SNE are computed from ICH.

**Effect of Contrastive Head** To prove the effectiveness of the instance- and cluster-level contrastive head, we conduct ablation studies on CIFAR-10 and ImageNet-10 by removing one of the two heads. Since the cluster assignments can no longer be directly obtained when the cluster-level contrastive head is removed, we perform k-means in the instance space instead. The results are shown in Table 4. Interestingly, ICH shows comparable performance on CIFAR-10 while CCH performs better on ImageNet-10, which suggests the joint effects of the two heads to some extent. Despite the performance improvement brought by CCH, we would like to emphasize that CCH is essential in achieving online clustering as it directly makes cluster predictions.

| Dataset     | Contrastive Head | NMI          | ACC          | ARI          |
|-------------|------------------|--------------|--------------|--------------|
| CIFAR-10    | ICH + CCH        | <b>0.705</b> | <b>0.790</b> | <b>0.637</b> |
|             | ICH Only         | 0.699        | 0.782        | 0.616        |
|             | CCH Only         | 0.592        | 0.657        | 0.499        |
| ImageNet-10 | ICH + CCH        | <b>0.859</b> | <b>0.893</b> | <b>0.822</b> |
|             | ICH Only         | 0.838        | 0.888        | 0.780        |
|             | CCH Only         | 0.850        | 0.892        | 0.816        |

Table 4: Effect of two contrastive heads.

**Reliance on Backbone Network** In our framework, any feature extractor could be adopted as the backbone network. To examine how much our model relies on the structure of the backbone network, we test three ResNets of different depths and report the results in Table 5. The results suggest that the representability of the backbone network contributes to the clustering performance. On relatively simple datasets like ImageNet-10, ResNet18 is sufficiently powerful to extract discriminative features. In addition, the performance of ResNet50 is worse than ResNet34 on CIFAR-10, which suggests a deeper network does not promise better performance.

## Conclusion

Based on the observation that the rows and columns of the feature matrix could be respectively realized as the representation of instances and clusters, we proposed the Contrastive Clustering (CC) method which dually conducts contrastive learning at the instance- and cluster-level under a unified framework. The proposed CC shows its promising performance in clustering. In the future, we plan to extend it to

| Dataset     | Backbone | NMI          | ACC          | ARI          |
|-------------|----------|--------------|--------------|--------------|
| CIFAR-10    | ResNet18 | 0.650        | 0.736        | 0.569        |
|             | ResNet34 | <b>0.705</b> | <b>0.790</b> | <b>0.637</b> |
|             | ResNet50 | 0.663        | 0.747        | 0.585        |
| ImageNet-10 | ResNet18 | 0.851        | 0.889        | 0.816        |
|             | ResNet34 | 0.859        | 0.893        | 0.822        |
|             | ResNet50 | <b>0.859</b> | <b>0.895</b> | <b>0.823</b> |

Table 5: Reliance on backbone network.

other tasks and applications such as semi-supervised learning and transfer learning.

## Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1406702 and 2020AAA0104500; in part by the Fundamental Research Funds for the Central Universities under Grant YJ201949; in part by NFSC under Grant 61971296, U19A2078, 61625204, 61836006, and 61836011; in part by the Fund of Sichuan University Tomorrow Advancing Life; and in part by Beijing Nova Program (Z201100006820068) from Beijing Municipal Science & Technology Commission.

## Broader Impact

The proposed method considers the clustering task that aims to group a set of unlabeled data into several classes. As a fundamental problem in machine learning, clustering has a wide range of applications, such as pattern recognition, data analysis, and image processing due to its powerful ability in data annotation and preprocessing. The method is evaluated on six wide-spread image datasets that are not at risk, but just like any learning method, the performance of our method depends on the data bias and cannot be guaranteed in more complex real-world applications. In this sense, it might bring some disturbances in decision making, and thus it should be carefully used especially in the area of health care, anomaly detection, autonomous vehicles, etc.

## References

- Asano, Y. M.; Rupprecht, C.; and Vedaldi, A. 2019. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371* .

- Bengio, Y.; Lamblin, P.; Popovici, D.; and Larochelle, H. 2007. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, 153–160.
- Cai, D.; He, X.; Wang, X.; Bao, H.; and Han, J. 2009. Locality preserving nonnegative matrix factorization. In *IJCAI*, volume 9, 1010–1015.
- Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 132–149.
- Chang, J.; Guo, Y.; Wang, L.; Meng, G.; Xiang, S.; and Pan, C. 2019. Deep Discriminative Clustering Analysis. *arXiv preprint arXiv:1905.01681*.
- Chang, J.; Wang, L.; Meng, G.; Xiang, S.; and Pan, C. 2017a. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, 5879–5887.
- Chang, J.; Wang, L.; Meng, G.; Xiang, S.; and Pan, C. 2017b. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, 5879–5887.
- Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.
- Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223.
- Darlow, L. N.; and Storkey, A. 2020. DHOG: Deep Hierarchical Object Grouping. *arXiv preprint arXiv:2003.08821*.
- Dosovitskiy, A.; Springenberg, J. T.; Riedmiller, M.; and Brox, T. 2014. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, 766–774.
- Ghasedi Dizaji, K.; Herandi, A.; Deng, C.; Cai, W.; and Huang, H. 2017. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, 5736–5745.
- Gowda, K. C.; and Krishna, G. 1978. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition* 10(2): 105–112.
- Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.
- Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017a. Improved deep embedded clustering with local structure preservation. In *IJCAI*, 1753–1759.
- Guo, X.; Liu, X.; Zhu, E.; and Yin, J. 2017b. Deep clustering with convolutional autoencoders. In *International conference on neural information processing*, 373–382. Springer.
- Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 297–304.
- Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 1735–1742. IEEE.
- Haeusser, P.; Plapp, J.; Golkov, V.; Aljalbout, E.; and Cremers, D. 2018. Associative deep clustering: Training a classification network with no labels. In *German Conference on Pattern Recognition*, 18–32. Springer.
- He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Hu, W.; Miyato, T.; Tokui, S.; Matsumoto, E.; and Sugiyama, M. 2017. Learning discrete representations via information maximizing self-augmented training. *arXiv preprint arXiv:1702.08720*.
- Huang, J.; Gong, S.; and Zhu, X. 2020. Deep Semantic Clustering by Partition Confidence Maximisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Ji, X.; Henriques, J. F.; and Vedaldi, A. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 9865–9874.
- Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.
- Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7.
- Li, J.; Zhou, P.; Xiong, C.; Socher, R.; and Hoi, S. C. 2020. Prototypical Contrastive Learning of Unsupervised Representations. *arXiv preprint arXiv:2005.04966*.
- Li, X.; Zhang, R.; Wang, Q.; and Zhang, H. 2020. Autoencoder Constrained Clustering With Adaptive Neighbors. *IEEE Transactions on Neural Networks and Learning Systems* 1–7.
- Liu, W.; Shen, X.; and Tsang, I. 2017. Sparse Embedded  $k$ -Means Clustering. In *Advances in Neural Information Processing Systems*, 3319–3327.
- Liu, X.; Dou, Y.; Yin, J.; Wang, L.; and Zhu, E. 2016. Multiple kernel k-means clustering with matrix-induced regularization. In *Proceedings of the thirtieth AAAI conference on artificial intelligence*, 1888–1894.

- MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.
- Nie, F.; Wang, X.; Jordan, M. I.; and Huang, H. 2016. The constrained laplacian rank algorithm for graph-based clustering. In *AAAI*, 1969–1976. Citeseer.
- Nie, F.; Zeng, Z.; Tsang, I. W.; Xu, D.; and Zhang, C. 2011. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks* 22(11): 1796–1808.
- Peng, X.; Feng, J.; Xiao, S.; Yau, W. Y.; Zhou, J. T.; and Yang, S. 2018. Structured AutoEncoders for Subspace Clustering. *IEEE Trans Image Process* 27(10): 5076–5086. ISSN 1057-7149. doi:10.1109/TIP.2018.2848470.
- Peng, X.; Xiao, S.; Feng, J.; Yau, W.; and Yi, Z. 2016. Deep Subspace Clustering with Sparsity Prior. In *Proc. of 25th Int. Joint Conf. Artif. Intell.*, 1925–1931. New York, NY, USA.
- Peng, X.; Yi, Z.; and Tang, H. 2015. Robust subspace clustering via thresholding ridge regression. In *AAAI*, volume 25, 3827–3833.
- Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.
- Sharma, V.; Tapaswi, M.; Sarfraz, M. S.; and Stiefelhagen, R. 2020. Clustering based Contrastive Learning for Improving Face Representations. *arXiv preprint arXiv:2004.02195*
- Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.-A.; and Bottou, L. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* 11(12).
- Wang, Z.; Li, Z.; Wang, R.; Nie, F.; and Li, X. 2020. Large graph clustering with simultaneous spectral embedding and discretization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- Wu, J.; Long, K.; Wang, F.; Qian, C.; Li, C.; Lin, Z.; and Zha, H. 2019. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, 8150–8159.
- Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, 478–487.
- Yang, J.; Parikh, D.; and Batra, D. 2016. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5147–5156.
- Zeiler, M. D.; Krishnan, D.; Taylor, G. W.; and Fergus, R. 2010. Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, 2528–2535. IEEE.
- Zelnik-Manor, L.; and Perona, P. 2005. Self-tuning spectral clustering. In *Advances in neural information processing systems*, 1601–1608.
- Zhong, H.; Chen, C.; Jin, Z.; and Hua, X.-S. 2020. Deep robust clustering by contrastive learning. *arXiv preprint arXiv:2008.03030* .