Clustering by Maximizing Mutual Information Across Views

Kien Do, Truyen Tran, Svetha Venkatesh
Applied Artificial Intelligence Institute (A2I2), Deakin University, Geelong, Australia
{k.do, truyen.tran, svetha.venkatesh}@deakin.edu.au

Abstract

We propose a novel framework for image clustering that incorporates joint representation learning and clustering. Our method consists of two heads that share the same backbone network - a "representation learning" head and a "clustering" head. The "representation learning" head captures fine-grained patterns of objects at the instance level which serve as clues for the "clustering" head to extract coarse-grain information that separates objects into clusters. The whole model is trained in an end-to-end manner by minimizing the weighted sum of two sample-oriented contrastive losses applied to the outputs of the two heads. To ensure that the contrastive loss corresponding to the "clustering" head is optimal, we introduce a novel critic function called "log-of-dot-product". Extensive experimental results demonstrate that our method significantly outperforms state-of-the-art single-stage clustering methods across a variety of image datasets, improving over the best baseline by about 5-7% in accuracy on CIFAR10/20, STL10, and ImageNet-Dogs. Further, the "two-stage" variant of our method also achieves better results than baselines on three challenging ImageNet subsets.

1. Introduction

The explosion of unlabeled data, especially visual content in recent years has led to the growing demand for effective organization of these data into semantically distinct groups in an unsupervised manner. Such data clustering facilitates downstream machine learning and reasoning tasks. Since labels are unavailable, clustering algorithms are mainly based on the similarity between samples to predict the cluster assignment. However, common similarity metrics such as cosine similarity or (negative) Euclidean distance are ineffective when applied to high-dimensional data like images. Modern image clustering methods [7, 17, 18, 37, 40, 41], therefore, leverage deep neural networks (e.g., CNNs, RNNs) to transform high-dimensional data into low-dimensional representation vectors in the latent space and perform clustering in that space. Ideally, a good clustering model assigns data to clusters to keep inter-group similarity low while maintaining high intra-group similarity. Most existing deep clustering methods do not satisfy both of these properties. For example, autoencoder-based clustering methods [19, 40, 42] often learn representations that capture too much information including distracting information like background or texture. This prevents them from computing proper similarity scores between samples at the cluster-level. Autoencoder-based methods have only been tested on simple image datasets like MNIST. Another class of methods [7, 17, 18] directly use cluster-assignment probabilities rather than representation vectors to compute the similarity between samples. These methods can only differentiate objects belonging to different clusters but not in the same cluster, hence, may incorrectly group distinct objects into the same cluster. This leads to low intra-group similarity.

To address the limitations of existing methods, we propose a novel framework for image clustering called Contrastive Representation Learning and Clustering (CRLC). CRLC consists of two heads sharing the same backbone network: a "representation learning" head (RL-head) that outputs a continuous feature vector, and a "clustering" head (C-head) that outputs a cluster-assignment probability vector. The RL-head computes the similarity between objects at the instance level while the C-head separates objects into different clusters. The backbone network serves as a medium for information transfer between the two heads, allowing the C-head to leverage disciminative fine-grained patterns captured by the RL-head to extract correct coarse-grained cluster-level patterns. Via the two heads, CRLC can effectively modulate the inter-cluster and intra-cluster similarities between samples. CRLC is trained in an end-to-end manner by minimizing a weighted sum of two sample-oriented contrastive losses w.r.t. the two heads. To ensure that the contrastive loss corresponding to the C-head leads to the tightest InfoNCE lower bound [30], we propose a novel critic called "log-of-dot-product" to be used in place of the conventional "dot-product" critic.

In our experiments, we show that CRLC significantly outperforms a wide range of state-of-the-art single-

stage clustering methods on five standard image clustering datasets including CIFAR10/20, STL10, ImageNet10/Dogs. The "two-stage" variant of CRLC also achieves better results than SCAN - a powerful two-stage clustering method on three challenging ImageNet subsets with 50, 100, and 200 classes. When some labeled data are provided, CRLC, with only a small change in its objective, can surpass many state-of-the-art semi-supervised learning algorithms by a large margin.

In summary, our main contributions are:

1. A novel framework for joint representation learning and clustering trained via two sample-oriented contrastive losses on feature and probability vectors;
2. An optimal critic for the contrastive loss on probability vectors; and,
3. Extensive experiments and ablation studies to validate our proposed method against baselines.

2. Preliminaries

2.1. Representation learning by maximizing mutual information across different views[1]

Maximizing mutual information across different views (or ViewInfoMax for short) allows us to learn view-invariant representations that capture the semantic information of data important for downstream tasks (e.g., classification). This learning strategy is also the key factor behind recent successes in representation learning [16, 28, 33, 36].

Since direct computation of mutual information is difficult [24, 32], people usually maximize the variational lower bounds of mutual information instead. The most common lower bound is InfoNCE [30] whose formula is given by:

$$I(X,\tilde{X}) \geq I_{\text{InfoNCE}}(X,\tilde{X}) \tag{1}$$

$$\triangleq \mathbb{E}_{p(x_{1:M})p(\tilde{x}|x_1)}\left[\log \frac{e^{f(\tilde{x},x_1)}}{\sum_{i=1}^{M} e^{f(\tilde{x},x_i)}}\right] + \log M \tag{2}$$

$$= -\mathcal{L}_{\text{contrast}} + \log M \tag{3}$$

where X, $\tilde{X}$ denote random variables from 2 different views. $x_{1:M}$ are M samples from p_X, $\tilde{x}$ is a sample from $p_{\tilde{X}}$ associated with x_1. $(\tilde{x}, x_1)$ is called a "positive" pair and $(\tilde{x}, x_i)$ $(i = 2, ..., M)$ are called "negative" pairs. $f(x, y)$ is a real value function called "critic" that characterizes the similarity between x and y. $\mathcal{L}_{\text{contrast}}$ is often known as the "contrastive loss" in other works [8, 33].

Since $\log \frac{e^{f(\tilde{x},x_1)}}{\sum_{i=1}^{M} e^{f(\tilde{x},x_i)}} \leq 0$, $I_{\text{InfoNCE}}(X,\tilde{X})$ is upper-bounded by $\log M$. It means that: i) the InfoNCE bound is very loose if $I(X,\tilde{X}) \gg \log M$, and ii) by increasing M, we can achieve a better bound. Despite being biased, $I_{\text{InfoNCE}}(X,\tilde{X})$ has much lower variance than other unbiased lower bounds of $I(X,\tilde{X})$ [30], which allows stable training of models.

Implementing the critic In practice, $f(\tilde{x}, x_i)$ is implemented as the scaled cosine similarity between the representations of $\tilde{x}$ and x_i as follows:

$$f(\tilde{x}, x_i) = f(\tilde{z}, z_i) = \tilde{z}^\top z_i/\tau \tag{4}$$

where $\tilde{z}$ and z_i are *unit-normed* representation vectors of $\tilde{x}$ and x_i, respectively; $\|\tilde{z}\|_2 = \|z_i\|_2 = 1$. $\tau > 0$ is the "temperature" hyperparameter. Interestingly, f in Eq. 4 matches the theoretically optimal critic that leads to the tightest InfoNCE bound for unit-normed representation vectors (detailed explanation in Appdx. A.4)

In Eq. 4, we use $f(\tilde{z}, z_i)$ instead of $f(\tilde{x}, x_i)$ to emphasize that the critic f in this context is a function of representations. In regard to this, we rewrite the contrastive loss in Eq. 3 as follows:

$$\mathcal{L}_{\text{FC}} = \mathbb{E}_{p(x_{1:M})p(\tilde{x}|x_1)}\left[-\log \frac{e^{f(\tilde{z},z_1)}}{\sum_{i=1}^{M} e^{f(\tilde{z},z_i)}}\right] \tag{5}$$

$$= \mathbb{E}_{p(x_{1:M})p(\tilde{x}|x_1)}\left[\tilde{z}^\top z_1/\tau - \log \sum_{i=1}^{M} \exp(\tilde{z}^\top z_i/\tau)\right] \tag{6}$$

where FC stands for "feature contrastive".

3. Method

3.1. Clustering by maximizing mutual information across different views

In the clustering problem, we want to learn a parametric classifier s_θ that maps each unlabeled sample x_i to a cluster-assignment probability vector $q_i = (q_{i,1}, ..., q_{i,C})$ (C is the number of clusters) whose component $q_{i,c}$ characterizes how likely x_i belongs to the cluster c ($c \in \{1, ..., C\}$). Intuitively, we can consider q_i as a representation of x_i and use this vector to capture the cluster-level information in x_i by leveraging the "ViewInfoMax" idea discussed in Section 2.1. It leads to the following loss for clustering:

$$\mathcal{L}_{\text{cluster}} = \mathbb{E}_{p(x_{1:M})p(\tilde{x}|x_1)}\left[-\log \frac{e^{f(\tilde{q},q_1)}}{\sum_{i=1}^{M} e^{f(\tilde{q},q_i)}}\right] - \lambda H(\tilde{Q}_{\text{avg}}) \tag{7}$$

$$= \mathcal{L}_{\text{PC}} - \lambda H(\tilde{Q}_{\text{avg}}) \tag{8}$$

where $\lambda \geq 0$ is a coefficient; $\tilde{q}$, q_i are probability vectors associated with $\tilde{x}$ and x_i, respectively. $\mathcal{L}_{\text{PC}}$ is the *proba-*

[1]Here, we use "views" as a generic term to indicate different transformations of the same data sample.

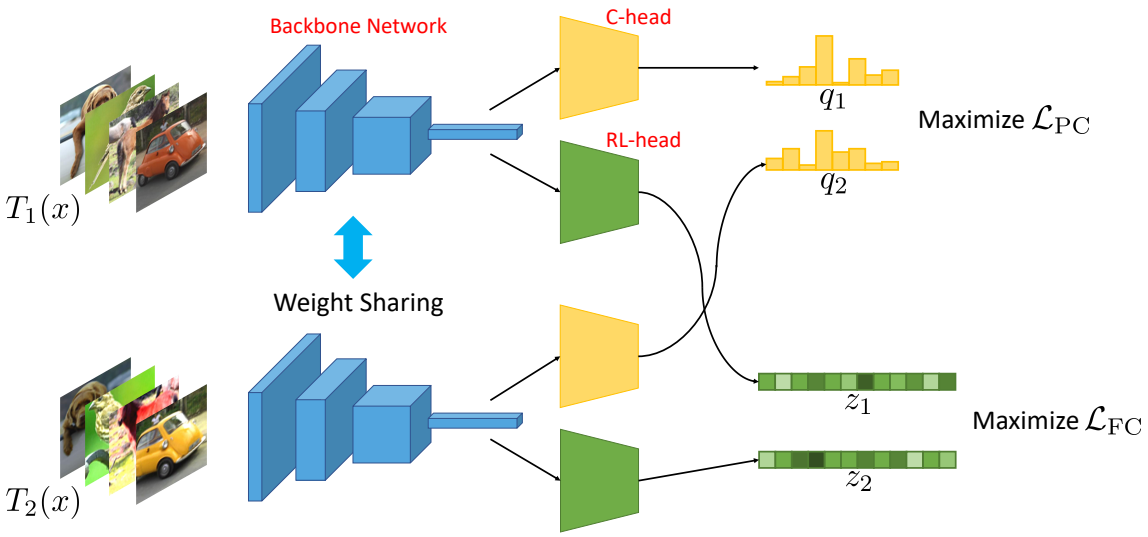


Figure 1: Overview of our proposed framework for Contrastive Representation Learning and Clustering (CRLC). Our framework consists of a "clustering" head and a "representation learning" head sharing the same backbone network. x denotes an input images and $T_1(x)$, $T_2(x)$ denote two different transformations of x.

bility contrastive loss similar to $\mathcal{L}_{\text{FC}}$ (Eq. 5) but with feature vectors replaced by probability vectors. H is the entropy of the *marginal* cluster-assignment probability $\tilde{q}_{\text{avg}} = \mathbb{E}_{p(x_1)p(\tilde{x}|x_1)}[\tilde{q}]$. Here, we maximize $H(\tilde{Q}_{\text{avg}})$ to avoid a degenerate solution in which all samples fall into the same cluster (e.g., $\tilde{q}$ is one-hot for all samples). However, it is free to use other regularizers on $\tilde{q}_{\text{avg}}$ rather than $-H(\tilde{Q}_{\text{avg}})$.

Choosing a suitable critic It is possible to use the conventional "dot-product" critic for $\mathcal{L}_{\text{PC}}$ as for $\mathcal{L}_{\text{FC}}$ (Eq. 4). However, this will lead to suboptimal results (Section 5.3) since $\mathcal{L}_{\text{PC}}$ is applied to categorical probability vectors rather than continuous feature vectors. Therefore, we need to choose a suitable critic for $\mathcal{L}_{\text{PC}}$ so that the InfoNCE bound associated with $\mathcal{L}_{\text{PC}}$ is tightest. Ideally, $f(\tilde{x}, x_i)$ should match the theoretically optimal critic $f^*(\tilde{x}, x_i)$ which is proportional to $\log p(\tilde{x}|x_i)$ (detailed explanation in Appdx. A.3). Denoted by $\tilde{y}$ and y_i the cluster label of $\tilde{x}$ and x_i respectively, we then have:

$$\log p(\tilde{x}|x_i) \approx \log \sum_{c=1}^{C} p(\tilde{y}=c|y_i=c)$$
$$\propto \log \sum_{c=1}^{C} \tilde{q}_c q_{i,c} = \log(\tilde{q}^\top q_i) \qquad (9)$$

Thus, the most suitable critic is $f(\tilde{q}, q_i) = \log(\tilde{q}^\top q_i)$ which we refer to as the *"log-of-dot-product"* critic. This critic achieves its maximum value when $\tilde{q}$ and q_i are the same one-hot vectors and its minimum value when $\tilde{q}$ and q_i are different one-hot vectors. Apart from this critic, we also list other nonoptimal critics in Appdx. A.1. Empirical comparison of the "log-of-dot-product" critic with other critics is provided in Section 5.3.

In addition, to avoid the *gradient saturation* problem of minimizing $\mathcal{L}_{\text{PC}}$ when probabilities are close to one-hot (explanation in Appdx. A.5), we smooth out the probabilities as follows:

$$q = (1-\gamma)q + \gamma r$$

where $r = \left(\frac{1}{C}, ..., \frac{1}{C}\right)$ is the uniform probability vector over C classes; $0 \leq \gamma \leq 1$ is the smoothing coefficient set to 0.01 if not otherwise specified.

Implementing the contrastive probability loss To implement $\mathcal{L}_{\text{PC}}$, we can use either the SimCLR framework [8] or the MemoryBank framework [36]. If the SimCLR framework is chosen, both $\tilde{q}$ and q_i ($i \in \{1, ..., M\}$) are computed directly from $\tilde{x}$ and x_i respectively via the parametric classifier s_θ. On the other hand, if the MemoryBank framework is chosen, we maintain a nonparametric memory bank $\mathcal{M}$ - a matrix of size $N \times C$ containing the cluster-assignment probabilities of all N training samples, and update its rows once a new probability is computed as follows:

$$q_{n,t+1} = \alpha q_{n,t} + (1-\alpha)\hat{q}_n \qquad (10)$$

where α is the momentum, which is set to 0.5 in our work if not otherwise specified; $q_{n,t}$ is the probability vector of the training sample x_n at step t corresponding to the n-th row of $\mathcal{M}$; $\hat{q}_n = s_\theta(x_n)$ is the new probability vector. Then, except $\tilde{q}$ computed via s_θ as normal, all q_i in Eq. 7 are sampled uniformly from $\mathcal{M}$. At step 0, all the rows of $\mathcal{M}$ are

initialized with the same probability of $\left(\frac{1}{C}, ..., \frac{1}{C}\right)$. We also tried implementing $\mathcal{L}_{\text{PC}}$ using the MoCo framework [14] but found that it leads to unstable training. The main reason is that during the early stage of training, the EMA model in MoCo often produces inconsistent cluster-assignment probabilities for different views.

3.2. Incorporating representation learning

Due to the limited representation capability of categorical probability vectors, models trained by minimizing the loss $\mathcal{L}_{\text{cluster}}$ in Eq.7 are not able to discriminate objects in the same cluster. Thus, they may capture suboptimal cluster-level patterns, which leads to unsatisfactory results.

To overcome this problem, we propose to combine clustering with contrastive representation learning into a unified framework called CRLC[2]. As illustrated in Fig. 1, CRLC consists of a "clustering" head (C-head) and a "representation learning" head (RL-head) sharing the same backbone network. The backbone network is usually a convolutional neural network which maps an input image x into a hidden vector h. Then, h is fed to the C-head and the RL-head to produce a cluster-assignment probability vector q and a continuous feature vector z, respectively. We simultaneously apply the clustering loss $\mathcal{L}_{\text{cluster}}$ (Eq. 8) and the feature contrastive loss $\mathcal{L}_{\text{FC}}$ (Eq. 6) on q and z respectively and train the whole model with the weighted sum of $\mathcal{L}_{\text{cluster}}$ and $\mathcal{L}_{\text{FC}}$ as follows:

$$\begin{aligned}\mathcal{L}_{\text{CRLC}} &= \mathcal{L}_{\text{cluster}} + \lambda \mathcal{L}_{\text{FC}} \\ &= \mathcal{L}_{\text{PC}} - \lambda_1 H(\tilde{Q}_{\text{avg}}) + \lambda_2 \mathcal{L}_{\text{FC}} \end{aligned} \tag{11}$$

where $\lambda_1, \lambda_2 \geq 0$ are coefficients.

3.3. A simple extension to semi-supervised learning

Although CRLC is originally proposed for unsupervised clustering, it can be easily extended to semi-supervised learning (SSL). There are numerous ways to adjust CRLC so that it can incorporate labeled data during training. However, within the scope of this work, we only consider a simple approach which is adding a crossentropy loss on labeled data to $\mathcal{L}_{\text{CRLC}}$. The new loss is given by:

$$\begin{aligned}\mathcal{L}_{\text{CRLC-semi}} &= \mathcal{L}_{\text{CRLC}} + \lambda \mathbb{E}_{(x_l, y_l)\sim\mathcal{D}_l} \left[-\log p(y_l|x_l)\right] \\ &= \mathcal{L}_{\text{PC}} - \lambda_1 H(\tilde{Q}_{\text{avg}}) + \lambda_2 \mathcal{L}_{\text{FC}} + \lambda_3 \mathcal{L}_{\text{xent}} \end{aligned} \tag{12}$$

We call this variant of CRLC "CRLC-semi". Despite its simplicity, we will empirically show that CRLC-semi outperforms many state-of-the-art SSL methods when only few labeled samples are available. We conjecture that the clustering objective arranges the data into disjoint clusters, making classification easier.

[2]CRLC stands for Contrastive Representation Learning and Clustering.

4. Related Work

There are a large number of clustering and representation learning methods in literature. However, within the scope of this paper, we only discuss works in two related topics, namely, contrastive learning and deep clustering.

4.1. Contrastive Learning

Despite many recent successes in learning representations, the idea of contrastive learning appeared long time ago. In 2006, Hadsell et. al. [13] proposed a max-margin contrastive loss and linked it to a mechanical spring system. In fact, from a probabilistic view, contrastive learning arises naturally when working with energy-based models. For example, in many problems, we want to maximize $\log p(y|x) = \log \frac{e^{f(y,x))}}{\sum_{y'\in\mathcal{Y}} e^{f(y',x)}}$ where y is the output associated with a context x and $\mathcal{Y}$ is the set of all possible outputs or vocab. This is roughly equivalent to maximizing $f(y, x)$ and minimizing $f(y', x)$ for all $y' \neq y$ but in a normalized setting. However, in practice, the size of $\mathcal{Y}$ is usually very large, making the computation of $p(y|x)$ expensive. This problem was addressed in [27, 36] by using Noise Contrastive Estimation (NCE) [12] to approximate $p(y|x)$. The basic idea of NCE is to transform the density estimation problem into a binary classification problem: "Whether samples are drawn from the data distribution or from a known noise distribution?". Based on NCE, Mikolov et. al. [25] and Oord et. al. [28] derived a simpler contrastive loss which later was referred to as the InfoNCE loss [30] and was adopted by many subsequent works [8, 11, 14, 26, 33, 43] for learning representations.

Recently, there have been several attempts to leverage inter-sample statistics obtained from clustering to improve representation learning on a large scale [1, 4, 47]. PCL [22] alternates between clustering data via K-means and contrasting samples based on their views and their assigned cluster centroids (or prototypes). SwAV [5] does not contrast two sample views directly but uses one view to predict the code of assigning the other view to a set of learnable prototypes. InterCLR [38] and ODC [45] avoid offline clustering on the entire training dataset after each epoch by storing a pseudo-label for every sample in the memory bank (along with the feature vector) and maintaining a set of cluster centroids. These pseudo-labels and cluster centroids are updated on-the-fly at each step via mini-batch K-means.

4.2. Deep Clustering

Traditional clustering algorithms such as K-means or Gaussian Mixture Model (GMM) are mainly designed for low-dimensional vector-like data, hence, do not perform well on high-dimensional structural data like images. Deep clustering methods address this limitation by leveraging the representation power of deep neural networks (e.g., CNNs,

RNNs) to effectively transform data into low-dimensional feature vectors which are then used as inputs for a clustering objective. For example, DCN [40] applies K-means to the latent representations produced by an auto-encoder. The reconstruction loss and the K-means clustering loss are minimized simultaneously. DEC [37], by contrast, uses only an encoder rather than a full autoencoder like DCN to compute latent representations. This encoder and the cluster centroids are learned together via a clustering loss proposed by the authors. JULE [41] uses a RNN to implement agglomerative clustering on top of the representations outputted by a CNN and trains the two networks in an end-to-end manner. VaDE [19] regards clustering as an inference problem and learns the cluster-assignment probabilities of data using a variational framework [20]. Meanwhile, DAC [7] treats clustering as a binary classification problem: "Whether a pair of samples belong to the same cluster or not?". To obtain a pseudo label for a pair, the cosine similarity between the cluster-assignment probabilities of the two samples in that pair is compared with an adaptive threshold. IIC [18] learns cluster assignments via maximizing the mutual information between clusters under two different data augmentations. PICA [17], instead, minimizes the contrastive loss derived from the the mutual information in IIC. While the cluster contrastive loss in PICA is cluster-oriented and can have at most C negative pairs (C is the number of clusters). Our probability contrastive loss, by contrast, is sample-oriented and can have as many negative pairs as the number of training data. Thus, in theory, our proposed model can capture more information than PICA. In real implementation, in order to gain more information from data, PICA has to make use of the "over-clustering" trick [18]. It alternates between minimizing $\mathcal{L}_{\text{PICA}}$ for C clusters and minimizing $\mathcal{L}_{\text{PICA}}$ for kC clusters ($k > 1$ denotes the "over-clustering" coefficient). DRC [46] and CC [23] enhances PICA by combining clustering with contrastive representation learning, which follows the same paradigm as our proposed CRLC. However, like PICA, DRC and CC uses cluster-oriented representations rather than sample-oriented representations.

In addition to end-to-end deep clustering methods, some multi-stage clustering methods have been proposed recently [29, 34]. The most notable one is SCAN [34]. This method uses representations learned via contrastive learning during the first stage to find nearest neighbors for every sample in the training set. In the second stage, neighboring samples are forced to have similar cluster-assignment probabilities. Our probability contrastive loss can easily be extended to handle neighboring samples (see Section 5.1.2).

5. Experiments

Dataset We evaluate our proposed method on 5 standard datasets for image clustering which are CIFAR10/20 [21], STL10 [9], ImageNet10 [10, 7], and ImageNet-Dogs [10, 7], and on 3 big ImageNet subsets namely ImageNet50/100/200 with 50/100/200 classes, respectively [10, 34]. A description of these datasets is given in Appdx. A.6. Our data augmentation setting follows [14, 36]. We first randomly crop images to a desirable size (32×32 for CIFAR, 96×96 for STL10, and 224×224 for ImageNet subsets). Then, we perform random horizontal flip, random color jittering, and random grayscale conversion. For datasets which are ImageNet subsets, we further apply Gaussian blurring at the last step [8]. Similar to previous works [7, 18, 17], both the training and test sets are used for CIFAR10, CIFAR20 and STL10 while only the training set is used for other datasets. We also provide results where only the training set is used for CIFAR10, CIFAR20 and STL10 in Appdx. A.8. For STL10, 100,000 auxiliary unlabeled samples are additionally used to train the "representation learning" head. However, when training the "clustering" head, these auxiliary samples are not used since their classes may not appear in the training set.

Model architecture and training setups Following previous works [17, 18, 34, 46], we adopt ResNet34 and ResNet50 [15] as the backbone network when working on the 5 standard datasets and on the 3 big ImageNet subsets, respectively. The "representation learning" head (RL-head) and the "clustering" head (C-head) are two-layer neural networks with ReLU activations. The length of the output vector of the RL-head is 128. The temperature τ (Eq. 5) is fixed at 0.1. To reduce variance in learning, we train our model with 10 C-subheads[3] similar to [18]. This only adds little extra computation to our model. However, unlike [17, 18, 46], we do *not* use an auxiliary "over-clustering" head to exploit additional information from data since we think our RL-head can do that effectively.

Training setups for end-to-end and two-stage clustering are provided in Appdx. A.7.

Evaluation metrics We use three popular clustering metrics namely Accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI) for evaluation. For unlabeled data, ACC is computed via the Kuhn-Munkres algorithm. All of these metrics scale from 0 to 1 and higher values indicate better performance. In this work, we convert the [0, 1] range into percentage.

5.1. Clustering

5.1.1 End-to-end training

Table 1 compares the performance of our proposed CRLC with a wide range of state-of-the-art deep clustering methods. CRLC clearly outperforms all baselines by a large

[3] The final $\mathcal{L}_{\text{cluster}}$ in Eq. 8 is the average of $\mathcal{L}_{\text{cluster}}$ of these C-subheads.

Dataset	CIFAR10			CIFAR20			STL10			ImageNet10			ImageNet-Dogs		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
JULE [41]	27.2	19.2	13.8	13.7	10.3	3.3	27.7	18.2	16.4	30.0	17.5	13.8	13.8	5.4	2.8
DEC [37]	30.1	25.7	16.1	18.5	13.6	5.0	35.9	27.6	18.6	38.1	28.2	20.3	19.5	12.2	7.9
DAC [7]	52.2	39.6	30.6	23.8	18.5	8.8	47.0	36.6	25.7	52.7	39.4	30.2	27.5	21.9	11.1
DDC [6]	52.4	42.4	32.9	-	-	-	48.9	37.1	26.7	57.7	43.3	34.5	-	-	-
DCCM [35]	62.3	49.6	40.8	32.7	28.5	17.3	48.2	37.6	26.2	70.1	60.8	55.5	38.3	32.1	18.2
IIC [18]	61.7	-	-	25.7	-	-	61.0	-	-	-	-	-	-	-	-
MCR2 [44]	68.4	63.0	50.8	34.7	36.2	16.7	49.1	44.6	29.0	-	-	-	-	-	-
PICA [17]	69.6	59.1	51.2	33.7	31.0	17.1	71.3	61.1	53.1	87.0	80.2	76.1	35.2	35.2	20.1
DRC [46]	72.7	62.1	54.7	36.7	35.6	20.8	74.7	64.4	56.9	**88.4**	83.0	**79.8**	38.9	38.4	23.3
C-head only	66.9	56.9	47.5	37.7	35.7	21.6	61.2	52.7	43.4	80.0	75.2	67.6	36.3	37.5	19.8
CRLC	**79.9**	**67.9**	**63.4**	**42.5**	**41.6**	**26.3**	**81.8**	**72.9**	**68.2**	85.4	**83.1**	75.9	**46.1**	**48.4**	**29.7**

Table 1: End-to-end clustering results on 5 standard image datasets. Due to space limit, we only show the means of the results. For the standard deviations, please refer to Appdx. A.8.

ImageNet	50 classes				100 classes				200 classes			
Metric	ACC	ACC5	NMI	ARI	ACC	ACC5	NMI	ARI	ACC	ACC5	NMI	ARI
K-means [34]	65.9	-	77.5	57.9	59.7	-	76.1	50.8	52.5	-	75.5	43.2
SCAN [34]	75.1	91.9	80.5	**63.5**	66.2	88.1	78.7	54.4	56.3	80.3	75.7	44.1
two-stage CRLC	**75.4**	**93.3**	**80.6**	63.4	**66.7**	**88.3**	**79.2**	**55.0**	**57.9**	**80.6**	**76.4**	**45.9**

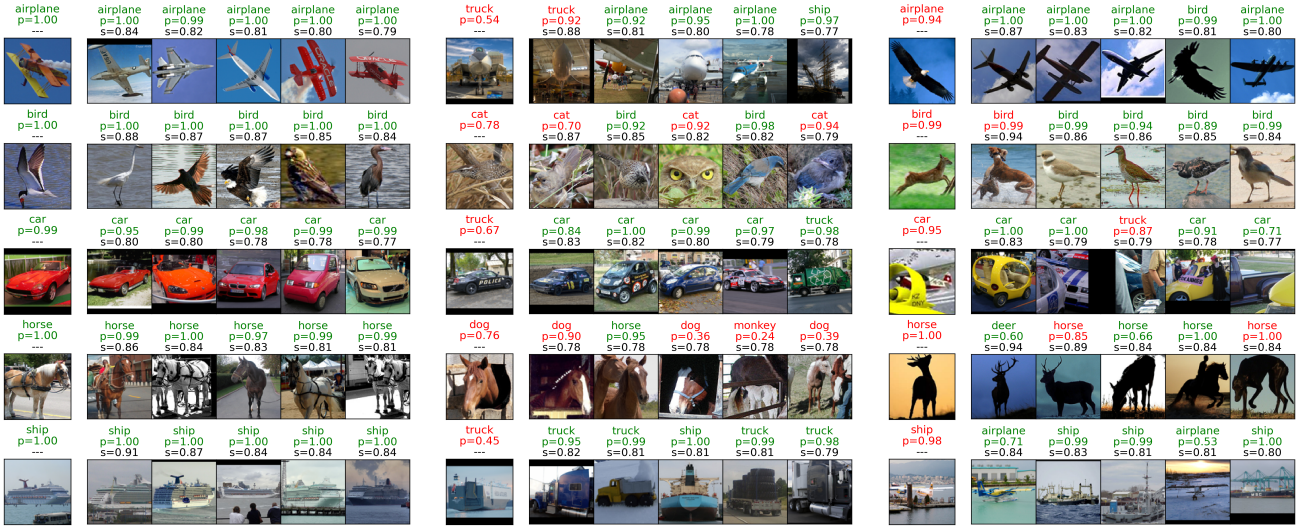
Table 2: Two-stage clustering results on ImageNet50/100/200.

margin on most datasets. For example, in term of clustering accuracy (ACC), our method improves over the best baseline (DRC [46]) by 5-7% on CIFAR10/20, STL10, and ImageNet-Dogs. Gains are even larger if we compare with methods that do not explicitly learn representations such as PICA [17] and IIC [18]. CRLC only performs worse than DRC on ImageNet10, which we attribute to our selection of hyperparameters. In addition, even when only the "clustering" head is used, our method still surpasses most of the baselines (e.g., DCCM, IIC). These results suggest that: i) we can learn semantic clusters from data just by minimizing the probability contrastive loss, and ii) combining with contrastive representation learning improves the quality of the cluster assignment.

To have a better insight into the performance of CRLC, we visualize some success and failure cases in Fig. 2 (and also in Appdx. A.11). We see that samples predicted correctly with high confidence are usually representative for the cluster they belong to. It suggests that CRLC has learned coarse-grained patterns that separate objects at the cluster level. Besides, CRLC has also captured fine-grained instance-level information, thus, is able to find nearest neighbors with great similarities in shape, color and texture to the original image. Another interesting thing from Fig. 2 is that the predicted label of a sample is often strongly correlated with that of the majority of its neighbors. It means that: i) CRLC has learned a smooth mapping from images to cluster assignments, and ii) CRLC tends to make "collective" errors (the first and third rows in Fig. 2c). Other kinds of errors may come from the closeness between classes (e.g., horse vs. dog), or from some adversarial signals in the input (e.g., the second row in Fig. 2b). Solutions for fixing these errors are out of scope of this paper and will be left for future work.

5.1.2 Two-stage training

Although CRLC is originally proposed as an end-to-end clustering algorithm, it can be easily extended to a two-stage clustering algorithm similar to SCAN [34]. To do that, we first pretrain the RL-head and the backbone network with $\mathcal{L}_{\text{FC}}$ (Eq. 6). Next, for every sample in the training data, we find a set of K nearest neighbors based on the cosine similarity between feature vectors produced by the pretrained network. In the second stage, we train the C-head by minimizing $\mathcal{L}_{\text{cluster}}$ (Eq. 8) with the positive pair consisting of a sample and its neighbor drawn from a set of K nearest neighbors. We call this variant of CRLC "two-stage" CRLC. In fact, we did try training both the C-head and the RL-head in the second stage by minimizing $\mathcal{L}_{\text{CRLC}}$ but could not achieve good results compared to training only the C-head. We hypothesize that finetuning the RL-head causes the model to capture too much fine-grained informa-



(a) correct (b) false negative (c) false positive

Figure 2: STL10 samples of 5 classes correctly (green) and incorrectly (red) predicted by CRLC. For each subplot, we show reference samples on the leftmost column and their nearest neighbors on the right. Neighbors are retrieved based on the normalized cosine similarity ("s") between the feature vectors of two samples. "p" denotes the confidence probability.

Dataset	CIFAR10		
Labels	10	20	40
MixMatch [3]	-	-	47.54±11.50
UDA [39]	-	-	29.05±5.93
ReMixMatch [2]	-	-	19.10±9.64
ReMixMatch†[4]	59.86±9.34	41.68±8.15	28.31±6.72
CRLC-semi	**46.75±8.01**	**29.81±1.18**	**19.87±0.82**

Table 3: Classification errors on CIFAR10. Lower values are better. Results of baselines are taken from [31]. †: Results obtained from external implementations of models.

tion and ignore important cluster-level information, which hurts the clustering performance.

In Table 2, we show the clustering results of "two-stage" CRLC on ImageNet50/100/200. Results on CIFAR10/20 and STL10 are provided in Appdx. A.9. For fair comparison with SCAN, we use the same settings as in [34] (details in Appdx. A.7). It is clear that "two-stage" CRLC outperforms SCAN on all datasets. A possible reason is that besides pushing neighboring samples close together, our proposed probability contrastive loss also pulls away samples that are not neighbors (in the negative pairs) while the SCAN's loss does not. Thus, by experiencing more pairs of samples, our model is likely to form better clusters.

[4]https://github.com/google-research/remixmatch

5.2. Semi-supervised Learning

Given the good performance of CRLC on clustering, it is natural to ask whether this model also performs well on semi-supervised learning (SSL) or not. To adapt for this new task, we simply train CRLC with the new objective $\mathcal{L}_{\text{CLRC-semi}}$ (Eq. 12). The model architecture and training setups remain almost the same (changes in Appdx. A.13).

From Table 3, we see that CRLC-semi, though is not designed especially for SSL, significantly outperforms many state-of-the-art SSL methods (brief discussion in Appdx. A.12). For example, CRLC-semi achieves about 30% and 10% lower error than MixMatch [3] and UDA [39] respectively on CIFAR10 with 4 labeled samples per class. Interestingly, the power of CRLC-semi becomes obvious when the number of labeled data is pushed to the limit. While most baselines cannot work with 1 or 2 labeled samples per class, CRLC-semi still performs *consistently well* with *very low standard deviations*. We hypothesize the reason is that CRLC-semi, via minimizing $\mathcal{L}_{\text{FC}}$, models the "smoothness" of data better than the SSL baselines. For more results on SSL, please check Appdx. A.14.

5.3. Ablation Study

Comparison of different critics in the probability contrastive loss In Fig. 3 left, we show the performance of CRLC on CIFAR10 and CIFAR20 w.r.t. different critic functions. Apparently, the theoretically sound "log-of-dot-product" critic (Eq. 9) gives the best results. The "negative-L2-distance" critic is slightly worse than the "log-of-dot-

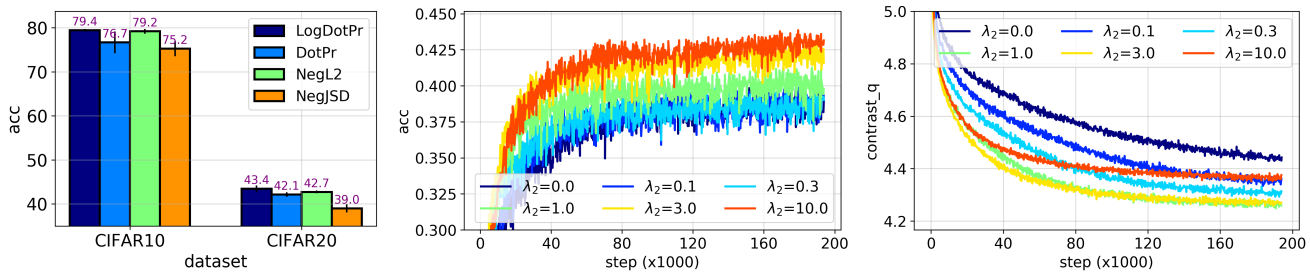


Figure 3: **Left**: Clustering accuracies of CRLC w.r.t. different critics on CIFAR10/20 (training set only). **Middle**, **Right**: Accuracy and $\mathcal{L}_{\text{PC}}$ curves of CRLC on CIFAR20 w.r.t. different coefficients of $\mathcal{L}_{\text{FC}}$ (λ_2 in Eq. 11).

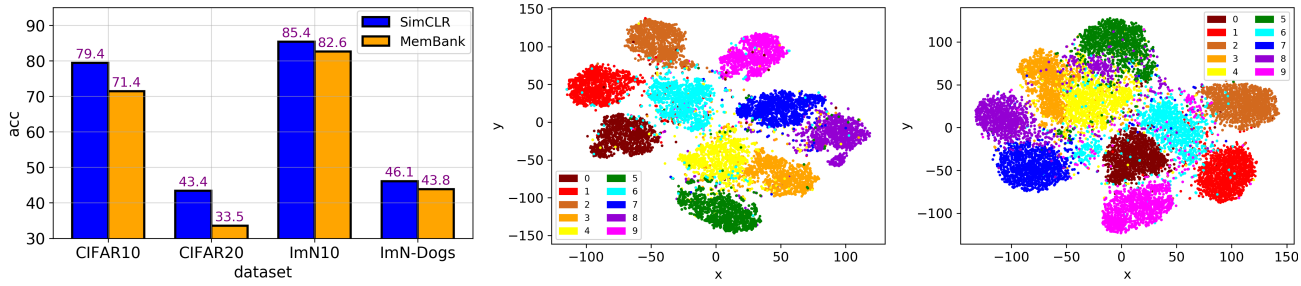


Figure 4: **Left**: Clustering accuracies of CRLC w.r.t. SimCLR [8] and MemoryBank [36] implementations. For CIFAR10/20, only the training set is used. **Middle**, **Right**: tSNE visualizations of the feature vectors learned by CRLC and SimCLR on the ImageNet10 train set, repectively.

product" critic while the "dot-product" and the "negative-JS-divergence" critics are the worst.

Contribution of the feature contrastive loss We investigate by how much our model's performance will be affected if we change the coefficient of $\mathcal{L}_{\text{FC}}$ (λ_2 in Eq. 11) to different values. Results on CIFAR20 are shown in Fig. 3 middle, right. Interestingly, minimizing both $\mathcal{L}_{\text{PC}}$ and $\mathcal{L}_{\text{FC}}$ simultaneously results in lower values of $\mathcal{L}_{\text{PC}}$ than minimizing only $\mathcal{L}_{\text{PC}}$ ($\lambda_2 = 0$). It implies that $\mathcal{L}_{\text{FC}}$ provides the model with more information to form better clusters. In order to achieve good clustering results, λ_2 should be large enough relative to the coefficient of $\mathcal{L}_{\text{PC}}$ which is 1. However, too large λ_2 results in a high value of $\mathcal{L}_{\text{PC}}$, which may hurts the model's performance. For most datasets including CIFAR20, the optimal value of λ_2 is 10.

Nonparametric implementation of CRLC Besides using SimCLR [8], we can also implement the two contrastive losses in CRLC using MemoryBank [36] (Section 3.1). This reduces the memory storage by about 30% and the training time by half (on CIFAR10 with ResNet34 as the backbone and the minibatch size of 512). However, MemoryBank-based CRLC usually takes longer time to converge and is poorer than the SimCRL-based counterpart as shown in Fig. 4 left. The contributions of the number of negative samples and the momentum coefficient to the performance of MemoryBank-based CRLC are analyzed in Appdx. A.10.2.

Mainfold visualization We visualize the manifold of the continous features learned by CRLC in Fig. 4 middle. We observe that CRLC usually groups features into well-separate clusters. This is because the information captured by the C-head has affected the RL-head. However, if the RL-head is learned independently (e.g., in SimCLR), the clusters also emerge but are usually close together (Fig. 4 right). Through both cases, we see the importance of contrastive representation learning for clustering.

6. Conclusion

We proposed a novel clustering method named CRLC that exploits both the fine-grained instance-level information and the coarse-grained cluster-level information from data via a unified sample-oriented contrastive learning framework. CRLC showed promising results not only in clustering but also in semi-supervised learning. In the future, we plan to enhance CRLC so that it can handle neighboring samples in a principled way rather than just views. We also want to extend CRLC to other domains (e.g., videos, graphs) and problems (e.g., object detection).

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 4

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 7

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019. 7

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 4

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 4

[6] Jianlong Chang, Yiwen Guo, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep discriminative clustering analysis. *arXiv preprint arXiv:1905.01681*, 2019. 6

[7] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887, 2017. 1, 5, 6

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 3, 4, 5, 8

[9] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[11] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 1*, pages 766–774, 2014. 4

[12] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 4

[13] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 4

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 4, 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[16] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 2

[17] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8849–8858, 2020. 1, 5, 6

[18] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019. 1, 5, 6

[19] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1965–1972, 2017. 1, 5

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 5

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[22] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 4

[23] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. *arXiv preprint arXiv:2009.09687*, 2020. 5

[24] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR, 2020. 2

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119, 2013. 4

[26] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 4

[27] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 419–426, 2012. 4

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 4

[29] Sungwon Park, Sungwon Han, Sundong Kim, Danu Kim, Sungkyu Park, Seunghoon Hong, and Meeyoung Cha. Improving unsupervised image clustering with robust learning. *arXiv preprint arXiv:2012.11150*, 2020. 5

[30] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180, 2019. 1, 2, 4

[31] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 7

[32] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019. 2

[33] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2, 4

[34] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer, 2020. 5, 6, 7

[35] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8150–8159, 2019. 6

[36] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2, 3, 4, 5, 8

[37] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016. 1, 5, 6

[38] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Delving into inter-image invariance for unsupervised visual representations. *arXiv preprint arXiv:2008.11702*, 2020. 4

[39] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. 2019. 7

[40] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR, 2017. 1, 5

[41] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5147–5156, 2016. 1, 5, 6

[42] Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6440–6449, 2019. 1

[43] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019. 4

[44] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in Neural Information Processing Systems*, 33, 2020. 6

[45] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6688–6697, 2020. 4

[46] Huasong Zhong, Chong Chen, Zhongming Jin, and Xian-Sheng Hua. Deep robust clustering by contrastive learning. *arXiv preprint arXiv:2008.03030*, 2020. 5, 6

[47] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012, 2019. 4