Deep Semantic Clustering by Partition Confidence Maximisation

Jiabo Huang
Queen Mary University of London
jiabo.huang@qmul.ac.uk

Shaogang Gong
Queen Mary University of London
s.gong@qmul.ac.uk

Xiatian Zhu
University of Surrey
xiatian.zhu@surrey.ac.uk

Abstract

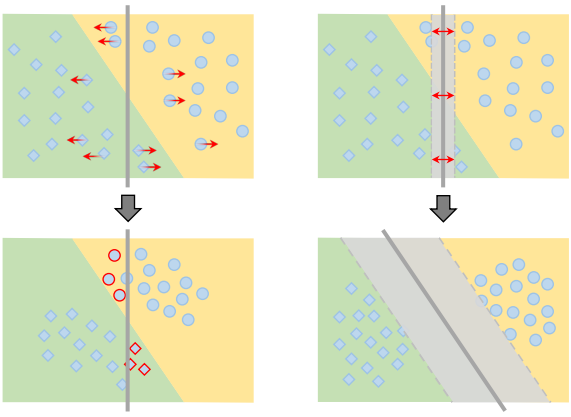
By simultaneously learning visual features and data grouping, deep clustering has shown impressive ability to deal with unsupervised learning for structure analysis of high-dimensional visual data. Existing deep clustering methods typically rely on local learning constraints based on inter-sample relations and/or self-estimated pseudo labels. This is susceptible to the inevitable errors distributed in the neighbourhoods and suffers from error-propagation during training. In this work, we propose to solve this problem by learning the most confident clustering solution from all the possible separations, based on the observation that assigning samples from the same semantic categories into different clusters will reduce both the intra-cluster compactness and inter-cluster diversity, i.e. lower partition confidence. Specifically, we introduce a novel deep clustering method named PartItion Confidence mAximisation (PICA). It is established on the idea of learning the most semantically plausible data separation, in which all clusters can be mapped to the ground-truth classes one-to-one, by maximising the "global" partition confidence of clustering solution. This is realised by introducing a differentiable partition uncertainty index and its stochastic approximation as well as a principled objective loss function that minimises such index, all of which together enables a direct adoption of the conventional deep networks and mini-batch based model training. Extensive experiments on six widely-adopted clustering benchmarks demonstrate our model's performance superiority over a wide range of the state-of-the-art approaches. The code is available online.

1. Introduction

As one of the most fundamental problems in machine learning, clustering has drawn extensive attention in a wide range of computer vision fields [5, 1, 25, 31] where the ground-truth labels are hard to acquire. Earlier clustering algorithms [32, 43, 38] are usually limited when dealing with high-dimensional imagery data due to the absence of discriminative visual representations [21, 22]. As a recent effort to solve this problem, deep clustering [19] is proposed to jointly optimise the objectives of representation learning and clustering with the help of the deep learning techniques [12, 30]. Although conducting cluster analysis with learnable representations holds the potential to benefit clustering on unlabelled data, how to improve the semantic plausibility of these clusters remains an open problem.

Recent deep clustering models either iteratively estimate cluster assignment and/or inter-sample relations which are then used as hypotheses in supervising the learning of deep neural networks [44, 6, 7, 50], or used in conjunction with clustering constraints [24, 16, 23]. In ideal cases, such alternation-learning methods can approach the performance of supervised models not the least benefiting from their robustness against noisy labels [18]. Nevertheless, they are susceptible to error-propagation as the process of alternating cluster assignment and representation learning is staged between two learning objectives and any error in local neighbourhoods is accumulated during this alternation (Fig. 1(a)). On the other hand, there are a few simultaneous learning methods that learn both the representation and clusters at the same time without explicit stages of cluster assignment and/or inter-sample relations estimation [24, 23]. They usually train with some pretext tasks that lead to unsatisfying solutions. Although such methods can avoid mostly the problem of error-propagation, they suffer from ambiguous learning constraints due to vague connection between the training supervision and clustering objective. Without global solution-level guidance to select from all the possible separations, the resulted clusters are usually semantically less plausible.

In this work, we propose a deep clustering method called *PartItion Confidence mAximisation* (PICA). Whilst



(a) Local Constraint (b) Global Constraint

Figure 1. **(a)** Local vs. **(b)** global learning constraints in deep clustering. Ground-truth classes are depicted by coloured backgrounds, decision boundaries and margins are indicated by grey solid lines and greyish shadow areas respectively. Arrow means learning supervision. With local learning constraints, a model is more likely to propagate errors in the neighbourhood due to lacking global structural guidance at the solution level as in (b).

the existence of intra-class visual discrepancy and inter-class affinity are common in most natural images, the majority of samples from the same semantic classes are still expected to share a high proportion of visual information. In this case, although a set of data can be separated in numerous ways according to various criteria, assigning samples from the same semantic categories to different clusters will reduce the resulted intra-cluster compactness and inter-cluster diversity [45] and lead to lower partition confidence. Based on this insight, PICA is designed specifically to encourage the model to learn the most *confident* clusters from all the possible solutions in order to find the most semantically plausible inter-class separation. This is in spirit of traditional maximal margin clustering [47, 9] which also seeks for most separable clustering solutions with shallow models (*e.g.* SVM), but differs notably in that both the feature representations and decision boundaries are end-to-end learned in our deep learning model. Specifically, we propose a partition uncertainty index that quantifies how confidently a deep model can make sense and separate a set of target images when performing both feature representation learning and cluster assignment concurrently. To fit the standard mini-batch based model learning, a stochastic approximation of the partition uncertainty index is introduced. We further formulate a novel objective loss function based on the stochastic partition uncertainty index to enable deep clustering with any off-the-shelf networks.

Our *contributions* are threefold: **(1)** We propose the idea of learning the most semantically plausible clustering solution by maximising partition confidence, which extends the classical maximal margin clustering idea [47, 9] to the deep learning paradigm. The proposed method makes no strong hypothesis on local inter-sample relations and/or cluster assignment which usually leads to error-propagation and inferior clustering solutions. **(2)** We introduce a novel deep clustering method, called *PartItion Confidence mAximisation* (PICA). PICA is built upon a newly introduced partition uncertainty index that is designed elegantly to quantify the global confidence of the clustering solution. To enable formulating a deep learning objective loss function, a novel transformation of the partition uncertainty index is further proposed. PICA can be trained end-to-end using a single objective loss function without whistles and bells (*e.g.* complex multi-stage alternation and multiple loss functions) to simultaneously learn a deep neural network and cluster assignment that can be mapped to the semantic category one-to-one. **(3)** A stochastic approximation of the partition uncertainty index is introduced to decouple it from the whole set of target images, therefore, enabling a ready adoption of the standard mini-batch model training.

The advantages of PICA over a wide range of the state-of-the-art deep clustering approaches have been demonstrated by extensive experiments on six challenging objects recognition benchmarks, including CIFAR-10/100 [28], STL-10 [8], ImageNet-10/Dogs [7] and Tiny-ImageNet [29].

2. Related Work

Existing deep clustering approaches generally fall into two categories according to the training strategy: **(1)** Alternate training [49, 46, 7, 44, 6, 50, 15] and **(2)** Simultaneous training [23, 36, 35, 16]

Alternate training strategy usually estimates the ground-truth membership according to the pretrained or up-to-date model and in return supervises the network training by the estimated information. DEC [46] initialises cluster centroids by conducting K-means [32] on pretrained image features and then fine-tunes the model to learn from the confident cluster assignments to sharpen the resulted prediction distribution. IDEC [15] shares a similar spirit and improves by a local structure preservation mechanism. JULE [50] combines the hierarchical agglomerative clustering idea with deep learning by a recurrent framework which merges the clusters that are close to each other. The method in [49] jointly optimises the objectives of Auto-Encoder [2] and K-means [32] and alternately estimates cluster assignment to learn a "clustering-friendly" latent space. DAC [7], DDC [6] and DCCM [44] exploit the inter-samples relations according to the pairwise distance between the latest sample features and train the model accordingly. Whilst explicit *local learning* constraints on cluster assignment computed from either the pretrained or up-to-date models usually lead

to a deterministic clustering solution, these approaches suffer from the problem of more severe error-propagation from the inconsistent estimations in the neighbourhoods during training. Comparing with them, our method makes no hard assumption on neither the cluster assignment nor the inter-samples relation so that it can refrain better from incorrectness accumulation.

Simultaneous training strategy often integrates deep representation learning [2, 42] with conventional cluster analysis [32, 10, 13] or other pretext objectives. These approaches do not explicitly learn from estimated cluster assignment but usually require good cluster structure to fulfil the training objectives. Methods in [23, 36, 35] train to optimise the objective of cluster analysis and use the reconstruction constraint from Auto-Encoder to avoid trivial solutions. ADC [16] formulates the optimisation objective to encourage consistent association cycles among cluster centroids and sample embeddings, while IIC [24] trains a model to maximise the mutual information between the predictions of positive sample pairs. Both of them randomly perturb the data distribution as a cue of positive relationships to improve model's robustness to visual transformations. Although the methods from this category alleviate the negative impact of inaccurate supervision from estimated information, their objectives are usually more ambiguous than that of the alternate approaches as they can be met by multiple different separations. Due to vague connection between the training supervision and clustering objective, this type of approaches tend to yield semantically less plausible clusters solutions when global solution-level guidance is absent. Our PICA model addresses this limitation by introducing a partition uncertainty index to quantify the global confidence of the clustering solution so as to select the most semantically plausible separation. Besides, it can also be trained in a concise and simultaneous manner without any ad-hoc strategy for pretraining or alternation.

3. Methodology

Problem Definition Given a set of N *unlabelled* target images $\mathcal{I} = \{I_1, I_2, ..., I_N\}$ drawn from K semantic classes $\mathcal{Y} = \{Y_1, Y_2, ..., Y_K\}$, the objective of deep clustering is learning to separate $\mathcal{I}$ into K clusters in an unsupervised manner by a convolutional neural network (CNN) model. There are typically two components learned jointly end-to-end: **(1)** feature extractor $f_{\boldsymbol{\theta}}(\cdot)$ that maps the target images into vector representations: $\boldsymbol{x} = f_{\boldsymbol{\theta}}(I)$, and **(2)** classifier $g_{\boldsymbol{\phi}}(\cdot)$ that assigns each feature representation $\boldsymbol{x}$ with a cluster membership distribution: $\boldsymbol{p} = g_{\boldsymbol{\phi}}(\boldsymbol{x}) = \{p_1; p_2; \cdots; p_K\}$.

Once the CNN model is trained, the cluster assignment can be predicted in a maximum likelihood manner as:

$$k^* = \arg\max_k(p_k), \quad k \in \{1, 2, \cdots, K\}. \tag{1}$$

Ideally, all the samples of a cluster would share the same target class labels. That being said, we aim to discover the underlying semantic class decision boundary directly from the raw data samples.

Approach Overview In general, image clustering is *not* a well-defined problem as multiple different solutions can all make sense of the input data [48]. This makes deep clustering extremely challenging due to totally lacking high-level guidance knowledge. Given this, in this work, we assume that the most *confident* data partition is the most promising and semantically plausible solution we are seeking for, as interpreted earlier.

Motivated by this consideration, we formulate a novel deep clustering method, called *PartItion Confidence mAximisation* (**PICA**). PICA is based on a partition uncertainty index (PUI) that measures how a deep CNN is capable of interpreting and partitioning the target image data. It is therefore a global clustering solution measurement (Fig. 1 (b)), fundamentally different to most existing deep clustering methods that leverage some local constraints on individual samples or sample pairs (Fig. 1 (a)) without global solution-wise learning guidance. This index is differentiable, hence, PICA simply needs to optimise it without whistles and bells using any off-the-shelf CNN models. An overview of PICA is depicted in Fig. 2.

3.1. Partition Uncertainty Index

We start by formulating a partition uncertainty index, a key element of our PICA. Given an input image I_i, suppose the cluster prediction of a CNN model is denoted as:

$$\boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{p}_{i,1} \\ \cdots \\ \boldsymbol{p}_{i,K} \end{bmatrix} \in \mathbb{R}^{K \times 1} \tag{2}$$

where $p_{i,j}$ specifies the predicted probability of the i-th image assigned to the j-th cluster, and there are a total of K clusters ($j \in [1, 2, \cdots, K]$). We then obtain the cluster prediction matrix of all the N target images as

$$P = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N] \in \mathbb{R}^{K \times N} \tag{3}$$

For presentation ease, we denote the j-th row of P as:

$$\boldsymbol{q}_j = [p_{1,j}, p_{2,j}, \cdots, p_{N,j}] \in \mathbb{R}^{1 \times N}, \quad j \in [1, 2, \cdots, K]. \tag{4}$$

Clearly, $\boldsymbol{q}_j$ collects the probability values of all the images for the j-th cluster, which summarises the *assignment statistics* of that cluster over the whole target data. Hence, we call it as a ***cluster-wise Assignment Statistics Vector*** (ASV).

Ideally, each image is assigned to only one cluster, *i.e.* each $\boldsymbol{p}$ is a one-hot vector (same as the ground-truth label vectors in supervised image classification). It is intuitive that this corresponds to the *most confident* clustering

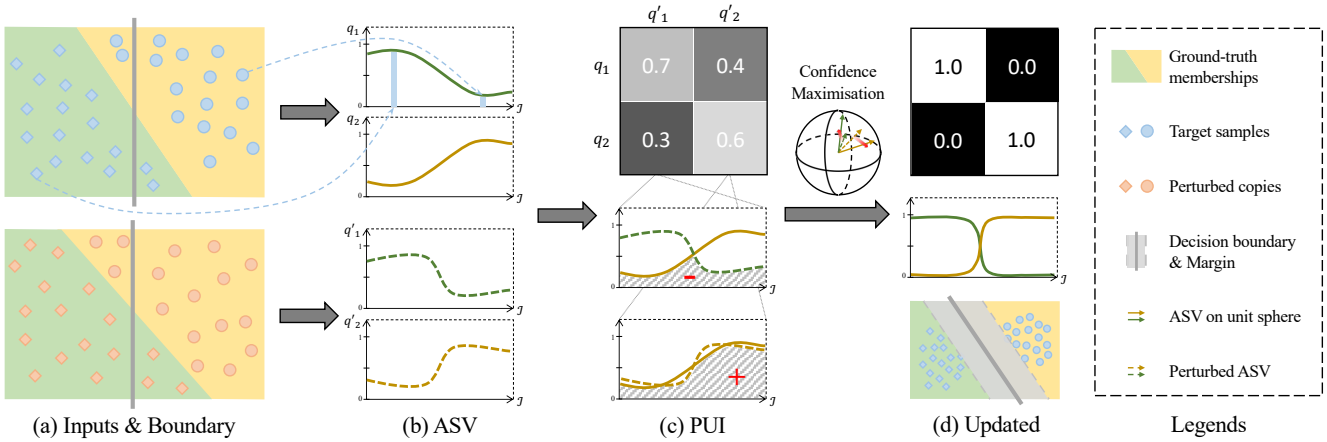


Figure 2. Overview of the proposed *PartItion Confidence mAximisation* (PICA) method for unsupervised deep clustering. **(a)** Given the input data and the decision boundaries determined by the CNN model, **(b)** PICA computes the *cluster-wise Assignment Statistics Vector* (ASV) in the forward pass using a mini-batch data and its randomly perturbed copy. **(c)** To minimise the *partition uncertainty index* (PUI), **(d)** PICA is trained to discriminate the ASV of all clusters on the hypersphere through a dedicated objective loss function, so as to learn the most confident and potentially promising clustering solution.

case which is the objective that PICA aims to achieve. For enabling the learning process of a deep clustering model towards this ideal (most confident) case, an objective loss function is typically needed. To that end, we design a partition uncertainty index as the learning target. Specifically, we observe that in the ideal case above, the ASV quantities of any two clusters, $\boldsymbol{q}_{j_1}$ and $\boldsymbol{q}_{j_2}$, are *orthogonal* to each other. Mathematically, this means that their cosine similarity (Eq. (5)) is 0 (due to not negative values in $\boldsymbol{q}_{j_1}$ and $\boldsymbol{q}_{j_2}$).

$$\cos(\boldsymbol{q}_{j_1}, \boldsymbol{q}_{j_2}) = \frac{\boldsymbol{q}_{j_1} \cdot \boldsymbol{q}_{j_2}}{\|\boldsymbol{q}_{j_1}\|_2 \|\boldsymbol{q}_{j_2}\|_2}, \quad j_1, j_2 \in [1, \cdots, K] \quad (5)$$

In the worst clustering cases where all the cluster prediction $\boldsymbol{p}$ are the same (*e.g.* the uniform distribution vector), we also have a constant value: $\cos(\boldsymbol{q}_{j_1}, \boldsymbol{q}_{j_2}) = 1$, since all the ASV quantities are the same. For any case in-between, the ASV cosine similarity of two clusters will range from 0 (most confident) to 1 (least confident).

In light of the above analysis, we formulate a partition uncertainty index (PUI) as the ASV cosine similarity set of all the cluster pairs as:

$$\mathcal{M}_{\text{PUI}}(j_1, j_2) = \cos(\boldsymbol{q}_{j_1}, \boldsymbol{q}_{j_2}), \quad j_1, j_2 \in [1, \cdots, K] \quad (6)$$

In form, $\mathcal{M}_{\text{PUI}}$ is a $K \times K$ matrix. By doing so, the learning objective of PICA is then to minimise the PUI (except the diagonal elements), which is supposed to provide the most confident clustering solution at its minimum.

A Stochastic Approximation The PUI as formulated in Eq. (6) requires using the entire target data which is often at large scale. This renders it *unsuited* to stochastic mini-batch based deep learning. To address this problem, we propose a stochastic approximation of PUI. Specifically, instead of using all the images (which is deterministic), at each training iteration we use a random subset $\mathcal{I}^t$ of them. In probability theory and statistics, this is sampling from a discrete uniform distribution in the whole target data space [20]. We call this approximation as *Stochastic PUI*. In practice, this allows to fit easily the mini-batch training of the standard deep learning, *e.g.* simply setting $\mathcal{I}^t$ as a mini-batch.

Formally, at the t-th training iteration, we have a mini-batch B of N_b samples to train the model and set $\mathcal{I}^t = B$. Let us denote the cluster prediction matrix of $\mathcal{I}^t$ made by the up-to-date model as:

$$P^t = \begin{bmatrix} \boldsymbol{q}_1^t \\ \cdots \\ \boldsymbol{q}_K^t \end{bmatrix} \in \mathbb{R}^{K \times N_b} \quad (7)$$

where $\boldsymbol{q}_j^t \in \mathbb{R}^{1 \times N_b}$ is the ASV of j-th cluster on mini-batch $\mathcal{I}^t$. As Eq. (6), we obtain the Stochastic PUI:

$$\mathcal{M}_{\text{S-PUI}}(j_1, j_2) = \cos(\boldsymbol{q}_{j_1}^t, \boldsymbol{q}_{j_2}^t), \quad j_1, j_2 \in [1, \cdots, K]. \quad (8)$$

The Stochastic PUI is in a spirit of dropout [40]. Instead of *neurons*, we randomly drop *data samples* in this case and realised in the standard mini-batch sampling process.

3.2. Learning Objective Function

Given the Stochastic PUI $\mathcal{M}_{\text{S-PUI}}$, as discussed earlier PICA is then trained to minimise it excluding the diagonal elements. To derive a typical objective loss function, we usually need a *scalar* measure. However, $\mathcal{M}_{\text{S-PUI}}$ is a $K \times K$ matrix. There is hence a need to transform it.

Recall that for any two different clusters, we want to minimise their ASV cosine similarity. This is actually reinforcing self-attention [41] by treating each cluster as a

data sample and suppressing all the inter-sample attention. Hence, we apply a softmax operation as self-attention to each cluster j and obtain a probabilistic measurement as:

$$m_{j,j'} = \frac{\exp(\mathcal{M}_{\text{S-PUI}}(j,j'))}{\sum_{k=1}^{K}\exp(\mathcal{M}_{\text{S-PUI}}(j,k))}, \quad j' \in [1,\cdots,K] \quad (9)$$

With this transformation, the learning objective is further simplified into maximising $\{m_{j,j}\}_{j=1}^{K}$.

By further treating $m_{j,j}$ as the model prediction probability on the ground-truth class of a training sample (a cluster j in this context), a natural choice is then to exploit the common cross-entropy loss function:

$$\mathcal{L}_{\text{ce}} = \frac{1}{K}\sum_{j=1}^{K} -\log(m_{j,j}) \quad (10)$$

As such, we formulate a scalar objective loss function $\mathcal{L}_{\text{ce}}$ that minimises effectively the matrix $\mathcal{M}_{\text{S-PUI}}$.

In clustering, there are algorithm-agnostic trivial solutions that assign a majority of samples into a minority of clusters. To avoid this, we introduce an extra constraint that minimises *negative* entropy of the cluster size distribution:

$$\mathcal{L}_{\text{ne}} = \log(K) - H(\mathcal{Z}), \text{ with } \mathcal{Z} = [z_1, z_2, \cdots, z_K] \quad (11)$$

where $H(\cdot)$ is the entropy of a distribution, and $\mathcal{Z}$ is L_1 normalised soft cluster size distribution with each element computed as $z_j = \frac{\sum(\boldsymbol{q}_j^t)}{\sum_{k=1}^{K}\sum(\boldsymbol{q}_k^t)}$. Using $\log(K)$ is to ensure non-negative loss values.

The overall objective function of PICA is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda\mathcal{L}_{\text{ne}} \quad (12)$$

where λ is a weight parameter.

3.3. Model Training

The objective function (Eq. (12)) of PICA is differentiable end-to-end, enabling the conventional stochastic gradient descent algorithm for model training. To improve the model robustness to visual transformations we use data augmentation to randomly perturb the training data distribution. We enforce the clustering invariance against image perturbations at the global solution level. More specifically, we use the original data to compute $\boldsymbol{q}_{j_1}^t$ and the transformed data to compute $\boldsymbol{q}_{j_2}^t$ in Eq. (8) at each iteration. The training procedure is summarised in Algorithm 1.

Algorithm 1 Deep clustering by PICA.

Input: Training data $\mathcal{I}$, training epochs N_{ep}, iterations per epoch N_{it}, target cluster number K;
Output: A deep clustering model;
for epoch $= 1$ **to** N_{ep} **do**
  for iter $= 1$ **to** N_{it} **do**
    Sampling a random mini-batch of images;
    Feeding the mini-batch into the deep model;
    Computing per-cluster ASV (Eq. (4));
    Computing the Stochastic PUI matrix (Eq. (8));
    Computing the objective loss (Eq. (12));
    Model weights update by back-propagation.
  end for
end for

4. Experiments

Datasets In evaluation, we used six object recognition datasets. **(1) CIFAR-10(/100)** [28]: A natural image dataset with 50,000/10,000 samples from 10(/100) classes for training and testing respectively. **(2) STL-10** [8]: An ImageNet [39] sourced dataset containing 500/800 training/test images from each of 10 classes and additional 100,000 samples from several unknown categories. **(3) ImageNet-10 and ImageNet-Dogs** [7]: Two subsets of ImageNet: the former with 10 random selected subjects and the latter with 15 dog breeds. **(4) Tiny-ImageNet** [29]: A subset of ImageNet with 200 classes. There are 100,000/10,000 training/test images evenly distributed in each category.

We adopted the clustering setup same as [24, 44, 7]: Using both the training and test sets (without labels) for CIFAR10/100 and STL-10, and only the training set for ImageNet-10, ImageNet-Dogs and Tiny-ImageNet; Taking the 20 super-classes of CIFAR-100 as the ground-truth.

Evaluation Metrics We used three standard clustering performance metrics: (a) Accuracy (**ACC**) is computed by assigning each cluster with the dominating class label and taking the average correct classification rate as the final score, (b) Normalised Mutual Information (**NMI**) quantifies the normalised mutual dependence between the predicted labels and the ground-truth, and (c) Adjusted Rand Index (**ARI**) evaluates the clustering result as a series of decisions and measures its quality according to how many positive/negative sample pairs are correctly assigned to the same/different clusters. All of these metrics scale from 0 to 1 and higher values indicate better performance.

Implementation Details For fair comparisons with previous works, we followed the same settings and most of the implementation choices as [24]. Specifically, we conducted all the experiments with a ResNet-like backbone. We used the auxiliary over-clustering strategy in a separate clustering head to exploit the additional data from irrelevant classes if available. For the over-clustering head, we set 700 clusters for Tiny-ImageNet (due to more ground-truth classes) and 70 clusters for all the others. The over-clustering head, discarded finally in test, was trained alternatively with the primary head. In case of no auxiliary data, we used the target data in over-clustering head, which plays a role of auxiliary learning. For training, we used Adam optimiser [26]

Dataset	CIFAR-10			CIFAR-100			STL-10			ImageNet-10			ImageNet-Dogs			Tiny-ImageNet		
Metrics	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
K-means	0.087	0.229	0.049	0.084	0.130	0.028	0.125	0.192	0.061	0.119	0.241	0.057	0.055	0.105	0.020	0.065	0.025	0.005
SC [52]	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 [14]	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 [4]	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 [2]	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 [42]	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 [37]	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 [51]	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 [27]	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 [50]	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 [46]	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 [7]	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 [16] †	-	0.325	-	-	0.160	-	-	0.530	-	-	-	-	-	-	-	-	-	
DDC [6]	0.424	0.524	0.329	-	-	-	0.371	0.489	0.267	0.433	0.577	0.345	-	-	-	-	-	-
DCCM [44]	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 [24] †	-	0.617	-	-	0.257	-	-	0.610	-	-	-	-	-	-	-	-	-	
PICA: (Mean)	0.561	0.645	0.467	0.296	0.322	0.159	0.592	0.693	0.504	0.782	0.850	0.733	0.336	0.324	0.179	**0.277**	0.094	0.016
PICA: (Best) †	**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**

Table 1. The clustering performance on six challenging object image benchmarks. The 1st/2nd best results are indicated in **red**/blue. The results of previous methods are taken from [44, 24]. †: The best result among multiple trials.

with a fixed learning rate 0.0001. All the models were randomly initialised and trained with 200 epochs. Considering no generalisation to test data in clustering, the regularisation penalties to model weights were deprecated. We set the batch size to 256 and repeated each in-batch sample 3 times. Three operations, including random rescale, horizontal flip and colour jitters, were adopted for data perturbations and augmentation. We applied the sobel filter for restraining the model from capturing meaningless patterns of trivial colour cues. The weight of entropy regularisation in Eq (12) was set to 2 empirically. We used the same hyper-parameters for all the experiments *without* exhaustive per-dataset tuning (which is unscalable, inconvenient nor unfriendly in deployment). To reflect the performance stability, we tested our model with 5 trials and reported the average and best results separately.

4.1. Comparisons to State-of-the-Art Methods

Table 1 compares the object image clustering performance of the proposed PICA method with a wide range of the state-of-the-art clustering approaches. We have the following observations: **(1)** PICA surpasses all the strong competitors in most cases, sometimes by a large margin. Taking the clustering accuracy (ACC) for example, PICA† outperforms the best competitor [44] on CIFAR-10 and ImageNet-10 with 7.3% and 16.0% respectively while the performance gain over IIC [24] on STL-10 is 10.3%. This demonstrates the overall remarkable ability of our PICA for image clustering. **(2)** DCCM [44] serves as a strong state-of-the-art model on most datasets except for STL-10, on which it is outperformed by both IIC [24] and our PICA by more significant margins. We attribute this to the capability of exploiting auxiliary data of both winner methods. Also, PICA is clearly superior to IIC for clustering the images of STL-10, suggesting the outstanding potential of our method for capitalising extra data during deep clustering. **(3)** The absolute margins obtained by PICA over existing methods on the more challenging CIFAR-100 and ImageNet-Dogs benchmarks are relatively smaller. This is not surprised, since these datasets present fine-grained object classes which give very subtle differences in-between; Without rich knowledge, even humans may make mistakes when differentiating such classes. The relative performance improvements of our method is more consistent.

4.2. Ablation Study

We conducted ablation studies to investigate the effect of different design choices in PICA with a fixed random seed.

Partition Confidence Dynamics We started with examining the clustering confidence dynamics during training, which underpins the key idea of our PICA. In this examination, we used the maximum prediction probability (Eq. (1)) of every image to measure the clustering confidence, and summarised their 50-bins histogram statistics. We performed this test on CIFAR-10 at four accuracy performance milestones: 0.10 (random guess), 0.30, 0.50 and 0.70. As shown in Fig. 3, **(a)** the model started with random clustering close to a uniform prediction; **(b,c)** Along with the training process, an increasing number of samples get more confident cluster assignment; **(d)** At the end of training, a majority of samples can be assigned into clusters with 0.98+ probability confidence, nearly one-hot predictions.

Avoiding under-clustering We examined how important PICA needs to solve the generic "under-clustering" prob-

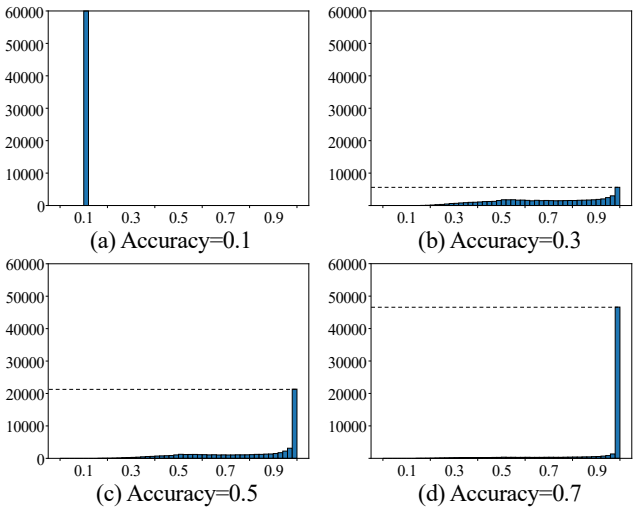


Figure 3. Partition confidence evolution in training on CIFAR-10.

lem (assigning most samples to a few clusters, *i.e*. trivial solutions) in our context. The results in Table 2 indicate that it is highly necessary to take into account this problem in model design otherwise the model will be trivially guided to such undesired solutions. This also verifies that the proposed PICA idea is compatible well with the entropy regularisation of the cluster size distribution (Eq. (11)), enabling to eliminate simply trivial results without resorting to complex designs or tricks.

Entropy	CIFAR-10	CIFAR-100	ImageNet-10
✗	0.246	0.168	0.650
✓	**0.696**	**0.330**	**0.829**

Table 2. Necessity of avoiding under-clustering using the entropy regularisation of the cluster size distribution. *Metric*: ACC.

Effect of over-clustering We examined the performance contribution of over-clustering in PICA which serves two purposes: (1) Leveraging extra auxiliary data from irrelevant classes for mining more information (*e.g*. on STL-10); (2) In case of no auxiliary data, playing a role of ensemble learning (*e.g*. on CIFAR-10). The results are given in Table 3. It is clear that over-clustering helps in both cases, and interestingly the margin on CIFAR-10 is even bigger than that on STL-10. Also note that without using over-clustering, our PICA can still achieve competitive performances (cf. Table 1).

Over-clustering	CIFAR-10	STL-10
✗	0.582	0.633
✓	**0.696**	**0.687**

Table 3. Effect of over-clustering. STL-10 has auxiliary data whilst CIFAR-10 does not. *Metric*: ACC.

Robustness to data perturbation We tested the clustering robustness against image perturbation in PICA. Unlike existing methods typically using data augmentation by random perturbation at the local sample level, we exploit it at the global clustering solution level. The results in Table 4 show that our PICA requires data augmentation for offering strong performances. While seemingly surprised, this is also rational/sensitive since our method performs the robustness enhance in a solution-wise manner; This effectively accumulate the augmentation effect of individual samples and potentially resulting in amplified effects eventually. However, this does not affect the use of PICA in general since data augmentation is just a standard necessary element of almost all deep learning methods.

Permutation	CIFAR-10	CIFAR-100	ImageNet-10
✗	0.310	0.147	0.734
✓	**0.696**	**0.330**	**0.829**

Table 4. Robustness to data perturbation in PICA. *Metric*: ACC.

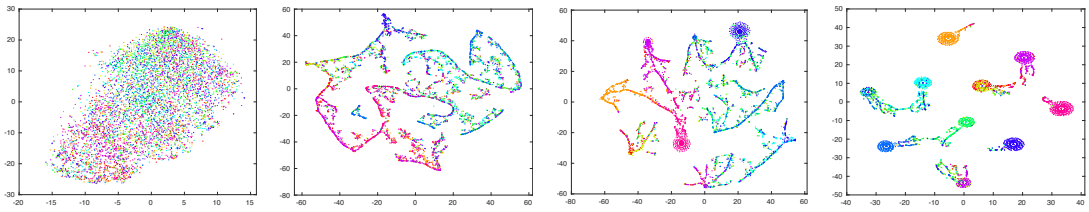
Sensitivity to model initialisation Model initialisation is an important part of both deep neural networks and clustering [2, 3]. We tested its sensitivity in our PICA w.r.t. model performance on CIFAR-10. Apart from the default initialisation of ResNet as delivered in PyTorch [34], we evaluated three more initialisation ways: Gaussian, Xavier [11], Kaiming [17]. Table 5 shows that PICA can work stably without clear variation in the overall performance when using different initialisation methods. This verifies that our method is insensitive to network initialisation.

Initialisation	NMI	ACC	ARI
Default	0.591	0.696	0.512
Gaussian	0.603	0.681	0.511
Xavier	0.610	0.676	0.511
Kaiming	0.617	0.680	0.525

Table 5. Model performance sensitivity to network initialisation.

4.3. Qualitative Study

Evolution of cluster assignment To provide a better understanding of how does PICA work, we analysed it qualitatively by visualising the evolution of cluster assignment across the whole training process. This enables us to find out how does our model gradually attain the final result. We tracked the model status during the whole training process on CIFAR-10 and evaluated at four accuracy performance milestones: 0.10 (random guess), 0.30, 0.50 and 0.70. Using t-SNE [33], we plotted the predictions of 6,000 randomly selected samples with the ground-truth classes colour encoded. Fig. 4 shows that, **(a)** the model started from a chaotic status where all the samples were assigned



(a) Accuracy=0.1 (b) Accuracy=0.3 (c) Accuracy=0.5 (d) Accuracy=0.7

Figure 4. Prediction dynamics of PICA across the training process on CIFAR-10. A total of 6,000 randomly sampled images were used. Ground-truth classes are colour coded.

Figure 5. Cases studies of 5 classes on STL-10. (**Left**) Successful cases; (**Middle**) False negative and (**Right**) false positive failure cases.

into each cluster with similar probabilities; **(b)** With the supervision from the proposed objective, easy samples with most salient observations were separated gradually while the remaining were still indecisive; **(c)** As the training proceeded, easy samples served as references for the others and attracted those with high visual similarities; **(d)** Finally, the model converged to a stable clustering solution which separated samples from different classes with some confusion around decision boundaries.

Success vs. failure cases Investigating both success and failure cases would offer extra insights into our method. We studied three cases for each one of 5 categories from STL-10: **(1)** Success cases: samples are correctly assigned to a target class, **(2)** False negative failure cases: samples of a target class are mis-assigned to other classes with high probability and **(3)** False positive failure cases: in terms of a target class, samples of other classes are wrongly assigned to it with high probability. As shown in Fig. 5, PICA can group together images of the same class with unconstrained variations. It also made mistakes in distinguishing the samples with high similarity in foreground or background. An explanation is that at the absence of ground-truth labels, the positive cluster pairs constructed by PICA are purely based on visual transformations without any information on either visual discrepancy within classes or affinity across classes. How to distinguish fine-grained classes remains unsolved, particularly for unsupervised learning and clustering.

5. Conclusion

We proposed a novel deep semantic clustering method, *PartItion Confidence mAximisation* (PICA). Specifically, we introduced a novel idea of learning the most promising and semantically plausible clustering solution from the partition confidence perspective, and formulated an elegant objective loss function based on a partition uncertainty index. This extends the idea of maximal margin clustering used in previous shallow models to the stronger deep learning paradigm with significant loss function formulation. PICA can be introduced in standard deep network models and end-to-end trainable without bells and whistles. Extensive experiments on six challenging datasets demonstrated the performance superiority of the proposed PICA method over a wide range of the state-of-the-art deep clustering methods.

6. Acknowledgements

This work was supported by the China Scholarship Council, Vision Semantics Limited, the Alan Turing Institute Turing Fellowship, and the Innovate UK Industrial Challenge Project on Developing and Commercialis-

ing Intelligent Video Analytics Solutions for Public Safety (98111-571149)

References

[1] Radhakrishna Achanta and Sabine Susstrunk. Superpixels and polygons using simple non-iterative clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4651–4660, 2017. 1

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 153–160, 2007. 2, 3, 6, 7

[3] Paul S Bradley and Usama M Fayyad. Refining initial points for k-means clustering. In *Proceedings of the International Conference on machine learning (ICML)*, volume 98, pages 91–99. Citeseer, 1998. 7

[4] Deng Cai, Xiaofei He, Xuanhui Wang, Hujun Bao, and Jiawei Han. Locality preserving nonnegative matrix factorization. 2009. 6

[5] 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 1–18, 2018. 1

[6] Jianlong Chang, Yiwen Guo, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep discriminative clustering analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–11, 2019. 1, 2, 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 (ICCV)*, pages 5879–5887, 2017. 1, 2, 5, 6

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AIStats)*, pages 215–223, 2011. 2, 5

[9] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 2

[10] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(11):2765–2781, 2013. 3

[11] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 7

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 1

[13] K Chidananda Gowda and G Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition (PR)*, 10(2):105–112, 1978. 3

[14] K Chidananda Gowda and G Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition (PR)*, 10(2):105–112, 1978. 6

[15] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. pages 1753–1759, 2017. 2

[16] Philip Haeusser, Johannes Plapp, Vladimir Golkov, Elie Aljalbout, and Daniel Cremers. Associative deep clustering: Training a classification network with no labels. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, pages 18–32. Springer, 2018. 1, 2, 3, 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 7

[18] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 10456–10465, 2018. 1

[19] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2016. 1

[20] Robert V Hogg, Joseph McKean, and Allen T Craig. *Introduction to mathematical statistics*. Pearson Education, 2005. 4

[21] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *Proceedings of the International Conference on machine learning (ICML)*, 2019. 1

[22] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning via affinity diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 1

[23] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 24–33, 2017. 1, 2, 3

[24] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–10, 2019. 1, 3, 5, 6

[25] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1943–1950. IEEE, 2010. 1

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[28] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 2, 5

[29] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 2, 5

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. 1

[31] Zichuan Liu, Guosheng Lin, Sheng Yang, Jiashi Feng, Weisi Lin, and Wang Ling Goh. Learning markov clustering networks for scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–11. IEEE, 2018. 1

[32] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 1, 2, 3

[33] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 7

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 7

[35] Xi Peng, Jiashi Feng, Jiwen Lu, Wei-Yun Yau, and Zhang Yi. Cascade subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 2, 3

[36] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. Deep subspace clustering with sparsity prior. pages 1925–1931, 2016. 2, 3

[37] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. 6

[38] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005. 1

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 4

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4

[42] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research (JMLR)*, 11(Dec):3371–3408, 2010. 3, 6

[43] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 1

[44] 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 International Conference on Computer Vision (ICCV)*, pages 1–12, 2019. 1, 2, 5, 6

[45] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[46] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on machine learning (ICML)*, pages 478–487, 2016. 2, 6

[47] Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans. Maximum margin clustering. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 1537–1544, 2005. 2

[48] Rui Xu and Donald C Wunsch. Survey of clustering algorithms. 2005. 3

[49] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the International Conference on machine learning (ICML)*, pages 3861–3870. JMLR. org, 2017. 2

[50] 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 (CVPR)*, pages 5147–5156, 2016. 1, 2, 6

[51] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE, 2010. 6

[52] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 1601–1608, 2005. 6