Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

HHS Public Access

Author manuscript

IEEE Trans Image Process. Author manuscript; available in PMC 2023 November 23.

Published in final edited form as:

IEEE Trans Image Process. 2022 ; 31: 7264–7278. doi:10.1109/TIP.2022.3221290.

SPICE: Semantic Pseudo-Labeling for Image Clustering

Chuang Niu [Member, IEEE],

Department of Biomedical Engineering, Center for Biotechnology and Interdisciplinary Studies, Rensselaer Polytechnic Institute, Troy, NY USA, 12180

Hongming Shan [Senior Member, IEEE],

Institute of Science and Technology for Brain-inspired Intelligence and MOE Frontiers Center for Brain Science and Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence, Fudan University, Shanghai, 200433, China, and also with the Shanghai Center for Brain Science and Brain-inspired Technology, Shanghai 201210, China

Ge Wang [Fellow, IEEE]

Department of Biomedical Engineering, Center for Biotechnology and Interdisciplinary Studies, Rensselaer Polytechnic Institute, Troy, NY USA, 12180

Abstract

The similarity among samples and the discrepancy among clusters are two crucial aspects of image clustering. However, current deep clustering methods suffer from inaccurate estimation of either feature similarity or semantic discrepancy. In this paper, we present a Semantic Pseudo-labeling-based Image ClustEring (SPICE) framework, which divides the clustering network into a feature model for measuring the instance-level similarity and a clustering head for identifying the cluster-level discrepancy. We design two semantics-aware pseudo-labeling algorithms, prototype pseudo-labeling and reliable pseudo-labeling, which enable accurate and reliable self-supervision over clustering. Without using any ground-truth label, we optimize the clustering network in three stages: 1) train the feature model through contrastive learning to measure the instance similarity; 2) train the clustering head with the prototype pseudo-labeling algorithm to identify cluster semantics; and 3) jointly train the feature model and clustering head with the reliable pseudo-labeling algorithm to improve the clustering performance. Extensive experimental results demonstrate that SPICE achieves significant improvements (~10%) over existing methods and establishes the new state-of-the-art clustering results on six balanced benchmark datasets in terms of three popular metrics. Importantly, SPICE significantly reduces the gap between unsupervised and fully-supervised classification; *e.g.* there is only 2% (91.8% vs 93.8%) accuracy difference on CIFAR-10. Our code is made publicly available at https://github.com/niuchuangnn/SPICE.

Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

niuc@rpi.edu .

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material includes the analysis of double softmax. Contact niuc@rpi.edu, hmshan@fudan.edu.cn, wangg6@rpi.edu for further questions about this work.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Keywords

Deep clustering; self-supervised learning; semi-supervised learning; representation learning

I. Introduction

IMAGE clustering aims to group images into different meaningful clusters without human annotations, and is an essential task in unsupervised learning with applications in many areas. At the core of image clustering are the measurements of the similarity among samples (images) and the discrepancy among semantic clusters. Recently, deep learning based clustering methods have achieved great progress thanks to the strong representation capability of deep neural networks.

Initially, by combining autoencoders with clustering algorithms, some deep clustering methods were proposed to learn representation features and perform clustering simultaneously and alternatively [2], [3], [4], [5], [6], [7], [8], [9], [10], achieving better results than the traditional methods. Due to the overestimation of low-level features, these autoencoder-based methods hardly capture discriminative features of complex images. Thus, a number of methods were proposed to learn discriminative label features under various constraints [11], [12], [13], [14], [14], [15], [16]. However, these methods have limited performance when directly using the label features to measure the similarity among samples. This is because the category-level features lose too much instance-level information to accurately measure the instance similarity. Very recently, Van Gansbeke *et al.* [1] proposed to leverage the embedding features of the unsupervised representation learning model to search for similar samples across the whole dataset, and then encourage a clustering model to output the same labels for similar instances, which further improved the clustering performance. Considering the imperfect embedding features, the local nearest samples in the embedding space do not always have the same semantics especially when the samples lie around the borderlines between different clusters as shown in Fig. 1(a), which may compromise the performance. Essentially, SCAN only utilizes the instance similarity for training the clustering model without explicitly exploring the semantic discrepancy between clusters, as shown in Fig. 1(b), so that it cannot identify the semantically inconsistent samples.

To this end, we propose a Semantic Pseudo-labeling-based Image ClustEring (SPICE) framework that synergizes the similarity among instances and semantic discrepancy between clusters to generate accurate and reliable self-supervision over clustering. In SPICE, the clustering network is divided into two parts: a feature model and a subsequent clustering head, which is exactly a traditional classification network. To effectively measure the instance similarity and the cluster discrepancy, we split the training process into three stages: 1) training the feature model; 2) training the clustering head; and 3) jointly training the feature model and clustering head. We highlight that there is no any annotations throughout the training process. More specifically, in the first stage we adopt the self-supervised contrastive learning paradigm to train the feature model, which can accurately measure the similarity among instance samples. In the second stage, we propose a prototype

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

pseudo-labeling algorithm to train the clustering head in the expectation-maximization (EM) framework with the feature model fixed, which can take into account both the instance similarity and the semantic discrepancy for clustering. In the final stage, we propose a reliable pseudo-labeling algorithm to jointly optimize the feature model and the clustering head, which can boost both the clustering performance and the representation ability.

Compared with the instance similarity based method [1], the proposed prototype pseudo-labeling algorithm can leverage the predicted cluster labels, obtained from the clustering head, to identify cluster prototypes in the feature space, and then assign each prototype label to its neighbor samples for training the clustering head alternatively. Thus, the inconsistency among borderline samples can be avoided when the prototypes are well captured, as shown in Fig. 1(b). On the other hand, given the predicted labels, the reliable pseudo-labeling algorithm can identify unreliable samples in the embedding space, as the yellow circles in Fig. 1(c), which will be filtered out during joint training. Therefore, the proposed SPICE can generate more accurate and reliable supervision by synergizing the similarity and discrepancy.

Our contributions are summarized as follows. (1) We propose a novel SPICE framework for image clustering, which generates accurate and reliable self-supervision for clustering by synergizing the feature similarity among similar samples and the discrepancy between clusters. To this end, the clustering network is divided into a feature model for measuring the similarity and a clustering head for identifying the discrepancy. (2) We design a prototype pseudo-labeling algorithm to identity prototypes for training the clustering head in an EM framework, which effectively reduces the semantic inconsistency of the samples around borderlines. (3) We design a reliable pseudo-labeling algorithm to select reliable samples for jointly training the feature model and the clustering head, which significantly improves the clustering performance.

Extensive experimental results demonstrate that SPICE outperforms the state-of-the-art clustering methods on common image clustering benchmarks by a large margin (~10%), closing the gap between unsupervised and supervised classification (down to ~2% on CIFAR10). Being the same as most of the recent studies, we also assume that the image dataset is well balanced over different clusters. Nevertheless, the performance of SPICE on imbalanced datasets is also evaluated in Subsection IV-G.

II. Related work

In this section, we first analyze the deep image clustering methods systematically, and then briefly review the related unsupervised representation learning and semi-supervised classification methods.

A. Deep Clustering

Deep clustering methods have shown significant superiority over traditional clustering algorithms, especially in computer vision. In a data-driven fashion, deep clustering can effectively utilize the representation ability of deep neural networks. Initially, some methods were proposed to combine deep neural networks with traditional clustering algorithms, as

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

shown in Fig. 2(a). Most of these clustering methods combine the stacked auto-encoders (SAE) [17] with the traditional clustering algorithms, such as *k*-means [18], [2], [4], [19], [3], Gaussian mixture model [5], [8], [9], spectral clustering [20], subspace clustering [10], [6], and relative entropy constraint [7]. However, since the pixel-wise reconstruction loss of SAE tends to over-emphasize low-level features, these methods have inferior performance in clustering images of complex contents due to the lack of object-level semantics. Instead of using SAE, Yang *et al.* [21] alternately perform the agglomerative clustering [22] and train the representation model by enforcing the samples within a selected cluster and its nearest cluster having similar features while pushing away the selected cluster from its other neighbor clusters. However, the performance of JULE [21] can be compromised by the errors accumulated during the alternation, and their successes in online scenarios are limited as they need to perform clustering on the entire dataset.

Recently, novel methods emerged that directly learn to map images into label features, which are used as the representation features during training and as the one-hot encoded cluster indices during testing [23], [11], [24], [13], [12], [25], [14], [1], [15], [26], [16], as shown in Fig. 2(b). Actually, these methods aim to train the clustering model in the unsupervised setting while using multiple indirect loss functions, such as sample relations [11], invariant information [12], [15], mutual information [13], partition confidence maximisation [25], attention [14], and entropy [14], [25], [1], [15]. Gupta *et al.* [27] proposed to train an ensemble of deep networks and select the predicted labels that a large number of models agree on as the high-quality labels, which are then used to train a ladder network [28] in a semi-supervised learning mode. However, the performance of these methods may be sub-optimal when using such label features to compute the similarity and discrepancy between samples, as the category-level label features can hardly reflect the relations of instance-level samples accurately.

To improve the representation learning ability, a two-stage unsupervised clustering method (TSUC) [29] was proposed. In the first stage, the unsupervised representation learning model was trained for initialization. In the second stage, a mutual information loss [13] based on the label features and a contrastive loss based on the embedding features were simultaneously optimized for training the whole model, as shown in Fig. 2(c). Although better initialization and contrastive loss can help learn representation features, the supervision based on the similarity and the discrepancy are computed independently, and the end-to-end training without accurate discriminative supervision will even harm the representation features as analyzed in Section IV-F2. In contrast, Van Gansbeke *et al.* [1] proposed a method called SCAN to use embedding features of the representation learning model for computing the instance similarity, based on which the label features are learned by encouraging similar samples to have the same label, as shown in Fig. 2(d). Sharing the same idea with SCAN, NNM [30] was proposed to enhance SCAN by searching similar samples on both the entire dataset and the sub-dataset. However, since the embedding features are not perfect, similar instances do not always have the same semantics especially when the samples lie near the borderlines of different clusters. Therefore, only using the instance similarity and ignoring the semantic discrepancy between clusters to guide model training may limit the clustering performance. Recently, Park *et al.* [31] proposed an add-on

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

module for improving the off-the-shelf unsupervised clustering method based on the semi-supervised learning [32] and label smoothing techniques [33], [34].

Based on the comprehensive analysis of existing deep cluttering methods, we present a new framework for image clustering, as shown in Fig. 2(e), which can accurately measure the similarity among samples and the discrepancy between clusters for the model training, and effectively reduce the semantic inconsistency for similar samples. On the other hand, our method is also related to the prototype-based traditional clustering algorithms [35], [36] and prototype-based learning for robust supervised classification [37]. Different from these methods, our proposed prototype pseudo-labeling algorithm alternatively estimates cluster labels and optimizes clustering heads to explicitly leverage the feature similarity among similar samples and the discrepancy between clusters in an unsupervised learning manner. In addition to the image clustering methods mentioned above, there are also promising deep learning-based clustering methods focusing on special [38], [39], multi-view clustering datasets [40], [41], and segmentation applications [42].

B. Unsupervised Representation Learning

Unsupervised representation learning aims to map samples/images into semantically meaningful features without human annotations, which facilitates various downstream tasks, such as object detection and classification. Previously, various pretext tasks were heuristically designed for this purpose, such as colorization [45], rotation [46], jigsaw [47], etc. Recently, contrastive learning methods combined with data augmentation strategies have achieved great success, such as SimCLR [48], MOCO [49], and BYOL [43], just to name a few. On the other hand, clustering based representation learning methods have also achieved great progress. Caron *et al.* [50] proposed to alternatively perform the k-means clustering algorithm on the entire dataset and train the classification network using the cluster labels. Without explicitly computing cluster centers, Asano *et al.* [51] proposed a self-labeling approach that directly infers the pseudo-labels from the predicted cluster labels of the full dataset based on the Sinkhorn-Knopp algorithm [52], and then uses the pseudo labels to train the clustering network. Taking the advantages of contrastive learning, SwAV [53] was proposed to simultaneously cluster the data while enforcing different transformations of the same image having the same cluster assignment. It is worth emphasizing that, different from the unsupervised deep clustering methods in Section II-A, these clustering based representation learning methods aim to learn the representation features by clustering a much larger number of clusters than the number of ground-truth classes. This is consistent with our observation that directly clustering the target number of classes without accurate supervision will harm the representation learning due to the over-compression of instance-level features. Usually, to evaluate the quality of learned features, a linear classifier is independently trained with ground truth labels by freezing the parameters of representation learning models.

In this work, we aim to achieve unsupervised clustering with the exact number of real classes. On the other hand, we not only train the feature model but also the clustering head without using any annotations. Actually, any unsupervised representation learning methods

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

can be implemented as our feature model, which can be further improved via the joint training in SPICE.

C. Semi-Supervised Classification

Our method is also related to the semi-supervised classification methods as actually we reformulate the unsupervised clustering task into a semi-supervised learning paradigm in the joint training stage. Semi-supervised classification methods aim to reduce the requirement of labeled data for training a classification model by providing a means of leveraging unlabeled data. In this category, remarkable results were obtained with consistency regularization [54], [55] that constrains the model to output the same prediction for different transformations of the same image, pseudo-labeling [56] that uses confident predictions of the model as the labels to guide training processes, and entropy minimization [57], [56] that steers the model to output high-confidence predictions. MixMatch [32] algorithm combines these principles in a unified scheme and achieves an excellent performance, which is further improved by ReMixMatch [58] along this direction. Recently, FixMatch [59] proposed a simplified framework that uses the confident prediction of a weakly transformed image as the pseudo label when the model is fed a strong transformation of the same image, delivering superior results.

In this work, we target a more challenging task of training the clustering network without using any annotations, sometimes achieving comparable or even better results than the state-of-the-art semi-supervised learning methods.

III. Method

We aim to cluster a set of N images $\mathfrak{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ into K classes by training a clustering network without using any annotations. The clustering network can be conceptually divided into two parts: a feature model that maps images to feature vectors, $\boldsymbol{f}_i = \mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\theta}_{\mathcal{F}})$, and a clustering head that maps feature vectors to the probabilities over K classes, $\boldsymbol{p}_i = \mathcal{C}(\boldsymbol{f}_i; \boldsymbol{\theta}_{\mathcal{C}})$, where $\boldsymbol{\theta}_{\mathcal{F}}$ and $\boldsymbol{\theta}_{\mathcal{C}}$ represent the trainable parameters of the feature model $\mathcal{F}$ and the clustering head $\mathcal{C}$, respectively. Different from the existing deep clustering methods, we use the outputs of the feature model to measure the similarity among samples and use the clustering head to identify the discrepancy between clusters for pseudo-labeling, as shown in Fig. 2. By effectively measuring both the similarity and discrepancy, we design two semantics-aware pseudo-labeling algorithms, prototype pseudo-labeling and reliable pseudolabeling, to generate accurate and reliable self-supervision.

Specifically, we split the network training into three stages as shown in Fig. 3. First, we optimize the feature model $\mathcal{F}$ through the instance-level contrastive learning that enforces the features from different transformations of the same image being similar and the features from different images being discriminative from each other. Second, we optimize the clustering head $\mathcal{C}$ with the proposed prototype pseudo-labeling algorithm while freezing the feature model learned in the first stage. Third, we optimize the feature model and the clustering head jointly with the proposed reliable pseudo-labeling algorithm.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

In the following subsections, we introduce each training stage in detail.

A. Feature Model Training with Contrastive Learning

To accurately measure the similarity of instance samples, here we adopt the instance discrimination based unsupervised representation learning method [49] for training the feature model. As shown in Fig. 3(a), there are two branches taking two random transformations of the same image as inputs, and each branch includes a feature model and a projection head that is a two-layer multilayer perceptron (MLP). During training, we only optimize the lower branch while the upper branch is updated as the moving average of the lower branch. As contrastive learning methods benefit from a large training batch, a memory bank [60] is used to maintain a queue of encoded negative samples for reducing the requirement of GPU memory size, which is denoted as $\left\{z_1^-, z_2^-, \ldots, z_{N_q}^-\right\}$, where N_q is the queue size.

Formally, given two transformations $\boldsymbol{x}'$ and $\boldsymbol{x}''$ of an image $\boldsymbol{x}$, the output of the upper branch is $\boldsymbol{z}^+ = \mathscr{P}(\mathscr{F}(x'; \boldsymbol{\theta}'_{\mathscr{F}}); \boldsymbol{\theta}'_{\mathscr{P}})$, and the output of the lower branch is $\boldsymbol{z} = \mathscr{P}(\mathscr{F}(\boldsymbol{x}''; \boldsymbol{\theta}_{\mathscr{F}}); \boldsymbol{\theta}_{\mathscr{P}})$, where $\mathscr{P}$ denotes the projection head with the parameters $\boldsymbol{\theta}_{\mathscr{P}}$, and $\boldsymbol{\theta}'_{\mathscr{F}}$ and $\boldsymbol{\theta}'_{\mathscr{P}}$ are the moving averaging versions of $\boldsymbol{\theta}_{\mathscr{F}}$ and $\boldsymbol{\theta}_{\mathscr{P}}$. The parameters $\boldsymbol{\theta}_{\mathscr{F}}$ and $\boldsymbol{\theta}_{\mathscr{P}}$ are optimized with the following loss function:

$$\mathscr{L}_{fea} = -\log\left(\frac{\exp\left(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z}^+/\tau\right)}{\sum_{i=1}^{N_q}\exp\left(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z}_i^-/\tau\right) + \exp\left(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z}^+/\tau\right)}\right), \tag{1}$$

where the negative sample z_i^- may be computed from any images other than the current image $\boldsymbol{x}$, and $\boldsymbol{\tau}$ is the temperature. Therefore, minimizing Eq. (1) is to make the representation features of different transformations from the same instance similar and the representation features of different instances dissimilar, which has been demonstrated as an effective pretext task to learn visual representations in the self-supervised learning setting [48], [49]. Then, the parameters of the upper branch is updated as $\boldsymbol{\theta}'_{\mathscr{F}} \leftarrow \mu\boldsymbol{\theta}'_{\mathscr{F}} + (1-\mu)\boldsymbol{\theta}_{\mathscr{F}}$, and $\boldsymbol{\theta}'_{\mathscr{P}} \leftarrow \mu\boldsymbol{\theta}'_{\mathscr{P}} + (1-\mu)\boldsymbol{\theta}_{\mathscr{P}}$, where $\mu \in [0, 1)$ is a momentum coefficient. The queue is updated by adding $\boldsymbol{z}^+$ to the end and removing the first item. All hyperparameters including $\boldsymbol{\tau} = 0.2$ and $\mu = 0.999$ are the same as those in [49]. The finally optimized feature model parameters are denoted as $\boldsymbol{\theta}^s_{\mathscr{F}}$, which will be used in the next stage.

Remark. In practice, any unsupervised representation learning methods and network architectures can be applied in the SPICE framework.

B. Clustering Head Training with Prototype Pseudo-Labeling

Based on the trained feature model, here we train the clustering head by explicitly exploring both the similarity among samples and the discrepancy between clusters. Formally, given the image dataset $\mathfrak{X}$ and the feature model parameters $\boldsymbol{\theta}^s_{\mathscr{F}}$ obtained in Section III-A, we aim

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

to train the clustering head only for predicting the cluster labels $\{y_i^s\}_{i=1}^N$. The clustering head $\mathscr{C}$ is a two-layer MLP mapping the features to the probabilities, $\boldsymbol{p}_i = \mathscr{C}(\boldsymbol{f}_i; \boldsymbol{\theta}_{\mathscr{C}})$, where $\boldsymbol{f}_i = \mathscr{F}(\boldsymbol{x}_i; \boldsymbol{\theta}_{\mathscr{F}}^s)$. However, in the unsupervised setting we do not have the ground truth for training. To address this issue, we propose a prototype pseudo-labeling algorithm that alternatively estimates the pseudo labels of batch-wise samples and optimizes the parameters of the clustering head in an EM framework.

Generally, this training stage is to solve two sets of variables, *i.e.*, the parameters of the clustering head $\mathscr{C}$, $\boldsymbol{\theta}_{\mathscr{C}}$, and the cluster labels $\{y_i^s\}$ of $\mathfrak{X}$ over K clusters. Analogous to k-means clustering algorithm [61], we solve two underlying subproblems alternatively in an EM framework: the expectation (E) step is solving $\{y_i^s\}$ given $\boldsymbol{\theta}_{\mathscr{C}}$, and the maximization (M) step is solving $\boldsymbol{\theta}_{\mathscr{C}}$ given $\{y_i^s\}$. Taking the advantages of contrastive learning, we clone the feature model into three branches as shown in Fig. 3(b):

- The top branch takes original images as inputs and outputs the embedding features $\boldsymbol{f}_i$;
- The middle branch takes the weakly transformed images as inputs and estimates the probabilities $\boldsymbol{p}_i$ over K clusters, which is then combined with $\boldsymbol{f}_i$ to generate the pseudo labels y_i^s through the proposed prototype pseudo-labeling algorithm;
- The bottom branch takes strongly transformed images as inputs and optimizes $\boldsymbol{\theta}_{\mathscr{C}}$ with the pseudo-labels.

The EM process for the clustering head training is detailed as follows.

Prototype Pseudo-Labeling (E-step).—The top branch computes the embedding features, $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_M]^{\mathrm{T}} \in \mathbb{R}^{M \times D}$, of a mini-batch of samples $\mathfrak{X}_b$, and the middle branch computes the corresponding probabilities, $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M]^{\mathrm{T}} \in \mathbb{R}^{M \times K}$, for the weakly transformed samples, $\alpha(\mathfrak{X}_b)$; here, M is mini-batch size, D is the dimension of the feature vector, and α denotes the weak transformation over the input image.

Given $\boldsymbol{P}$ and $\boldsymbol{F}$, the top confident samples are selected to estimate the prototypes for each cluster, and then the indices of the cluster prototypes are assigned to their nearest neighbors as the pseudo labels. Formally, the top confident samples for each cluster, taking the k-th cluster as an example, are selected as:

$$\mathfrak{F}_k = \{f_i \mid i \in \mathrm{argtopk}(\boldsymbol{P}_{:,k}, \frac{M}{K}), \forall i = 1, 2, \ldots, M\}, \tag{2}$$

where $\boldsymbol{P}_{:,k}$ denotes the k-th column of matrix $\boldsymbol{P}$ and argtopk $(\boldsymbol{P}_{:,k}, \frac{M}{K})$ returns the top $\frac{M}{K}$ confident sample indices from $\boldsymbol{P}_{:,k}$. Naturally, the cluster centers $\{\gamma_k\}_{k=1}^K$ in the embedding space are computed as

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\gamma_k = \frac{K}{M} \sum_{\boldsymbol{f}_i \in \mathfrak{F}_k} \boldsymbol{f}_i, \quad \forall k = 1, 2, \ldots, K. \tag{3}$$

By computing the cosine similarity between embedding features $\boldsymbol{f}_i$ and the cluster center γ_k, we select $\frac{M}{K}$ nearest samples to γ_k, denoted by $\mathfrak{X}^k$, to have the same cluster label, $y_i^s = k, \forall \boldsymbol{x}_i \in \mathfrak{X}^k$. Thus, a mini-batch of images with semantic pseudo-labels, $\mathfrak{X}^s$, is constructed as

$$\mathfrak{X}^s = \left\{ (\boldsymbol{x}_i, y_i^s) \mid \forall \boldsymbol{x}_i \in \mathfrak{X}^k, k = 1, 2, \ldots, K \right\}. \tag{4}$$

A toy example of the prototype pseudo-labeling process is shown in Fig. 4, where there is a batch of 10 samples and 3 clusters, 3 confident samples for each cluster are selected according to the predicted probabilities to calculate the prototypes in the feature space, and 3 nearest samples to each cluster are selected and labeled. Note that there may exist overlapped samples between different clusters, so there are two options to handle these labels: one is the overlap assignment that one sample may have more than one cluster labels as indicated by the blue and red circles in Fig. 4, and the other is the non-overlap assignment that all samples have only one cluster label as indicated by the dashed red circle. We found that the overlap assignment is better as analyzed in Section IV-F2.

Training Clustering Head (M-step).—Given the labeled samples $\mathfrak{X}^s$, the clustering head parameters are optimized in a supervised learning manner. Specifically, we compute the probabilities of strong transformations $\beta(\mathfrak{X}^s)$ in the forward pass, where β denotes the strong augmentation operator. Then, the clustering head $\mathscr{C}$ can be optimized in the backward pass by minimizing the following cross-entropy (CE) loss:

$$\mathscr{L}_{clu} = \frac{1}{M} \sum_{i=1}^{M} \mathscr{L}_{ce}(y_i^s, p_i'), \tag{5}$$

where $\boldsymbol{p}_i' = \mathrm{softmax}(\boldsymbol{p}_i)$ and $\boldsymbol{p}_i = \mathscr{A}\left(\mathscr{F}\left(\beta(\boldsymbol{x}_i); \boldsymbol{\theta}_{\mathscr{F}}^s\right); \boldsymbol{\theta}_{\mathscr{A}}\right)$, $\mathscr{L}_{ce}$ is the cross-entropy loss function.

In Eq. (5), we use double softmax functions before computing CE loss, as $\boldsymbol{p}_i$ is already the output of a softmax function. Considering that the pseudo-labels are not as accurate as the ground truth labels, the basic idea behind the double softmax implementation is to reduce the learning speed especially when the predictions are of low probabilities, which can benefit the dynamically clustering process during training (please see Appendix A for the detailed analysis of the double softmax implementation). The above process for training the clustering head is summarized in Algorithm 1.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Algorithm 1: Training Clustering Head.

Input: Dataset $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, $\boldsymbol{\theta}_{\mathcal{F}}^{s}$, K, M, m, T, α, β
Output: Cluster label y_i^s of $x_i \in \mathcal{X}$

Set feature model parameters to $\boldsymbol{\theta}_{\mathcal{F}}^{s}$, $t = 0$, and initialize $\boldsymbol{\theta}_{\mathcal{C}}$;
while $t < T$ **do**
  for $b = 1, 2, \ldots, \lfloor \frac{N}{M} \rfloor$ **do**
    ***E-step*:**
    Select M samples from $\mathcal{X}$ as $\mathcal{X}_b$;
    Compute embedding features $\boldsymbol{F} = \mathcal{F}(\mathcal{X}_b; \boldsymbol{\theta}_{\mathcal{F}}^{s})$;
    Predict probabilities $\boldsymbol{P} = \mathcal{C}(\mathcal{F}(\alpha(\mathcal{X}_b); \boldsymbol{\theta}_{\mathcal{F}}^{s}); \boldsymbol{\theta}_{\mathcal{C}})$;
    Construct labeled image set $\mathcal{X}^s$ with Eqs. (2), (3), and (4) ;
    ***M-step*:**
    Compute probabilities $\boldsymbol{P} = \mathcal{C}(\mathcal{F}(\beta(\mathcal{X}_b); \boldsymbol{\theta}_{\mathcal{F}}^{s}); \boldsymbol{\theta}_{\mathcal{C}})$;
    Optimize $\boldsymbol{\theta}_{\mathcal{C}}$ by minimizing Eq. (5) ;
  end
  $t \leftarrow t + 1$
end
Select the best clustering head with the minimum loss as $\boldsymbol{\theta}_{\mathcal{C}}^{s}$;
foreach $\boldsymbol{x}_i \in \mathcal{X}$ **do**
  $\boldsymbol{p}_i = \mathcal{C}(\mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\theta}_{\mathcal{F}}^{s}); \boldsymbol{\theta}_{\mathcal{C}}^{s}])$;
  $y_i^s = \arg\max_k(\boldsymbol{p}_i)$;
end

Remark. In this stage, we fix the parameters of feature models from the representation learning, and only optimize the lightweight clustering head. Thus, the computational burden is significantly reduced so that we can train multiple clustering heads simultaneously and independently. By doing so, the instability of clustering from the initialization can be effectively alleviated through selecting the best clustering head. Specifically, the best head with the parameters $\boldsymbol{\theta}_{\mathscr{C}}^{s}$ can be selected for the minimum loss value of $\mathscr{L}_{clu}$ over the whole dataset; *i.e.*, we set $M = N$ and follow E-step and M-step in Algorithm 1 to compute the loss value. During testing, the best clustering head is used to cluster the input images into different clusters.

C. Joint Training with Reliable Pseudo-Labeling

The feature model and the clustering head are optimized separately so far, which tends to be a sub-optimal solution. On the one hand, the imperfect feature model may lead to some similar features corresponding to really different clusters; thus, assigning neighbor samples with the same pseudo-label is not always reliable. On the other hand, the imperfect clustering head may assign really dissimilar samples with the same cluster label, such that only using the predicted labels for fine-tuning is also not always reliable. To overcome these problems, we design a reliable pseudo-labeling algorithm to train the feature model and clustering head jointly for further improving the clustering performance.

Reliable Pseudo-Labeling.—Given the embedding features and the predicted labels $\{(\boldsymbol{x}_i, \boldsymbol{f}_i, y_i^s)\}_{i=1}^{N}$ obtained in Section III-B, we select N_s nearest samples for each sample $\boldsymbol{x}_i$ according to the cosine similarity between embedding features. The corresponding labels of these nearest samples are denoted by $\mathfrak{L}_i$. Then, the semantically consistent ratio r_i of the sample $\boldsymbol{x}_i$ is defined as

$$r_i = \frac{1}{N_s} \sum_{y \in \mathfrak{L}_i} \mathbb{1}(y = y_i^s) . \tag{6}$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Given a predefined threshold λ, if $r_i > \lambda$, the sample $(\boldsymbol{x}_i, y_i^s)$ is identified as the reliably labeled for joint training, and otherwise the corresponding label is ignored. Through the reliable pseudo-labeling, a subset of reliable samples $\mathfrak{X}^r$ are selected as:

$$\mathfrak{X}^r = \{(\boldsymbol{x}_i, y_i^s) \mid r_i > \lambda, \quad \forall i = 1, 2, \ldots, N\}. \tag{7}$$

Joint Training.—Given the above partially labeled samples, the clustering problem can be converted into a semi-supervised learning paradigm to train the clustering network jointly. Here we adapt a simple semi-supervised learning method [59]. During training, the subset of reliably labeled samples keep fixed. On the other hand, all training samples should be consistently clustered, *i.e.*, different transformations of the same image are constrained to have the consistent prediction. To this end, as shown in Fig. 3(c), the confidently predicted label of weak transformations is used as the pseudo-label for strong transformations of the same image. Formally, the consistency pseudo label y_j^u of the sample $\boldsymbol{x}_j$ is calculated as in Eq. (8):

$$y_j^u = \begin{cases} \arg\max(\boldsymbol{p}_j) & \text{if } \max(\boldsymbol{p}_j) \geq \eta \\ -1 & \text{otherwise} \end{cases}, \tag{8}$$

where $\boldsymbol{p}_j = \mathscr{C}(\mathscr{F}(\alpha(\boldsymbol{x}_j); \boldsymbol{\theta}_{\mathscr{F}}); \boldsymbol{\theta}_{\mathscr{C}}))$, and η is the confidence threshold.

Then, the whole network parameters $\boldsymbol{\theta}_{\mathscr{F}}$ and $\boldsymbol{\theta}_{\mathscr{C}}$ are optimized with the following loss function:

$$\begin{aligned} \mathscr{L}_{joint} = & \frac{1}{L}\sum_{i=1}^{L} \underbrace{\mathscr{L}_{ce}(y_i^s, \mathscr{C}(\mathscr{F}(\alpha(\boldsymbol{x}_i); \boldsymbol{\theta}_{\mathscr{F}}); \boldsymbol{\theta}_{\mathscr{C}}))}_{\text{partial samples with reliable pseudo-labels}} \\ & + \frac{1}{U}\sum_{j=1}^{U} \underbrace{\mathbb{1}(y_j^u \geq 0)\mathscr{L}_{ce}(y_j^u, \mathscr{C}(\mathscr{F}(\beta(\boldsymbol{x}_j); \boldsymbol{\theta}_{\mathscr{F}}); \boldsymbol{\theta}_{\mathscr{C}}))}_{\text{all samples with consistency pseudo-labels}}, \end{aligned} \tag{9}$$

where the first term is computed with reliably labeled samples $(\boldsymbol{x}_i, y_i^s)$ drawn from $\mathfrak{X}^r$, and the second term is computed with pseudo-labeled samples $(\boldsymbol{x}_i, y_i^u)$ drawn from the whole dataset $\mathfrak{X}$, which is dynamically labeled by thresholding the confident predictions as in Eq. (8). L and U denote the numbers of labeled and unlabeled images in a mini-batch.

Remark. Although we adapt the FixMatch [59] semi-supervised classification method for image clustering in this work, we highlight that other semi-supervised algorithms can also be used here with reliable samples generated by the proposed reliable pseudo-labeling algorithm.

Note that SPICE sheds light on the importance of utilizing both instance-level similarity and semantic-level discrepancy for clustering. With this key idea in mind, this study focuses on developing the semantic pseudo-labeling algorithms that can actually estimate the instance

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

similarity and semantic discrepancy for better clustering results. Actually, SPICE presents a general unsupervised clustering framework that gradually trains the feature model, clustering head, and whole model end-to-end. This framework is able to organically unify advanced unsupervised representation learning and semi-supervised learning methods for clustering through the proposed semantic pseudo-labeling method.

IV. Experiments and Results

A. Benchmark Datasets and Evaluation Metrics

We evaluated the performance of SPICE on six commonly used image clustering datasets, including STL10, CIFAR-10, CIFAR-100-20, ImageNet-10, ImageNet-Dog, and Tiny-ImageNet. The key details of each dataset are summarized in Table I, where the datasets reflect a diversity of image sizes, the number of images, and the number of clusters. Different existing methods used different image sizes for training and testing; for example, CC [15] resizes all images of these six datasets into 224 × 224, and GATCluster [14] studies the effectiveness of different image sizes on ImageNet-10 and ImageNet-Dog, showing that too large or too small may harm the clustering performance. In this work, we naturally use the original size of images without resizing to a larger size of images. For the ImageNet, we adopt the commonly used image size of 224 × 224.

Three popular metrics are used to evaluate clustering results, including Adjusted Rand Index (ARI) [62], Normalized Mutual Information (NMI) [63], and clustering Accuracy (ACC) [64].

B. Implementation Details

For a fair comparison, we mainly adopted two backbone networks, *i.e.* ResNet18 and ResNet34 [73], for representation learning. The clustering head in SPICE consists of two fully-connected layers, where the dimensions of the input and intermediate layer are the same and denoted as D, and the dimension of the output is the number of clusters and denoted as K. Specifically, $D = 512$ for both ResNet18 and ResNet34 backbone networks, and the cluster number K is predefined as the number of classes on the target dataset as shown in Table I. To show how the joint training (third stage) improves the clustering performance, we refer to SPICE without joint training as SPICE_{s}, where the subscript s indicates the separate training.

For representation learning, we use MoCo-v2 [49] in all our experiments, which was also used in SCAN [1]. For weak augmentation, a standard flip-and-shift augmentation strategy is implemented as in FixMatch [59]. For strong augmentation, we adopt the same strategies used in SCAN [1]. Specifically, the images were strongly augmented by composing Cutout [74] and four randomly selected transformations from RandAugment [75].

In the first training stage, we duplicated the optimization settings in [49] that the stochastic gradient descent (SGD) optimizer was used with weight decay of 0.0001, momentum of 0.9 and the cosine learning rate schedule, batch size was set to 128, and the initial learning rate to 0.015. Note that the memory bank size in [49] was set to 65,536 by default. However, the datasets CIFAR-10, CIFAR-100-20, ImageNet-10, and ImageNet-Dog contain less than

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

65,536 images, and thus the memory back size for these datasets was empirically set to 10,240. In the second training stage, the Adam optimizer was used with the constant learning rate of 0.005, and the batch size M was set to 1,000. Here 10 clustering heads were simultaneously trained, and the best head with the minimum loss was selected as the final head in each trial. In the third training stage, we used the same optimization settings in [59] that the SGD optimizer was used with weight decay of 0.0005, momentum of 0.9 and the cosine learning rate schedule, batch size was set to 128 (16 reliable labeled samples and 112 unlabeled samples), the initial learning rate to 0.03, and the threshold η to 0.95. To select the reliably labeled images for SPICE through reliable pseudo-labeling, we empirically selected $N_s = 100$ and $\lambda = 0.95$.

In all our experiments, feature model training and joint training were distributed across 4 GPUs, and clustering heads were trained on a single GPU. The training time depends on various factors, such as the number of images, image size, model size, training epochs, etc. By default, we set the number of training epochs for the feature model and the clustering head to 1000 and 100 epochs respectively in our experiments, which are the same as those used for training the competing models [1], and the number of training iterations for joint learning to 1,048,576, the same as that in [59]. As an example, clustering STL10 images with SPICE took about 18 hours (800 epochs for feature learning, 30 epochs for training clustering heads, and 500,000 iterations for joint learning) to achieve state-of-the-art results.

C. Clustering Performance Comparison

The existing methods can be divided into two groups according to their training and testing settings. One is to train and test the clustering model on the whole dataset combining the train and test splits as one. The other is to train and test the clustering model on the separate train and test datasets. For a fair comparison, the proposed SPICE is evaluated under both two settings and compared with the existing methods accordingly.

Table II shows the comparison results of clustering on the whole dataset. The results obtained using classic clustering algorithms (*i.e.*, *k*-means, spectral clustering, agglomerative clustering, NMF) and the unsupervised deep feature models (*i.e.*, AE, SDAE, DCGAN, DeCNN, VAE) were originally reported in [11], [25]. In reference to the recently developed methods [12], [14], [15], the same backbone, *i.e.* ResNet34, was used during learning the feature model and clustering head. Duplicating the original settings in FixMatch [59], we used WideResNet-28-2 for CIFAR-10, WideResNet-28-8 for CIFAR-100-20, and WideResNet-37-2 for STL-10. For ImageNet-10, ImageNet-Dog, and Tiny-ImageNet datasets that were not used in FixMatch, we simply used the same ResNet34 during joint learning. The results show that SPICE improves ACC, NMI, and ARI by 8.8%, 12.6%, and 14.4% respectively over the previous best results that were recently reported by CC [15] on STL10. On average, our proposed method also improves ACC, NMI, and ARI by about 10% on ImageNet-Dog-15, CIFAR-10, CIFAR-100-20, and Tiny-ImageNet-200. It is worth emphasizing that, without the joint training stage, SPICE_s still performs better than the existing deep clustering methods using the same network architecture on most of the datasets. *These results convincingly show the superior performance of the proposed method using exactly the same backbone networks and datasets.* The final joint training results

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

are obviously better than those from the separate training on all datasets, especially on CIFAR-10 (improved by 8.8% for ACC) and CIFAR-100-20 (improved by 7.0% for ACC). In clustering the images on Tiny-ImageNet-200, our results are significantly better than the existing results. These results are, however, still relatively low. This is mainly due to the class hierarchies, *i.e.*, some classes share the same supper class, as analyzed in [1]. Due to this issue, some clusters cannot be reliably labeled based on the reliable pseudo-labeling algorithm so that end-to-end training cannot further boost clustering performance when directly using the produced labels in the second stage. Thus, it is still an open problem for clustering images which form a large number of hierarchical clusters.

Table III shows the comparison results of clustering on the split train and test datasets. For a fair comparison, we reimplement SCAN with MoCo [49] for representation learning, denoted as $\text{SCAN}_{\text{MoCo}}$. It can be seen that the results of $\text{SCAN}_{\text{MoCo}}$ on SLT10 are obviously better than SCAN, while the performance on CIFAR-10 and CIFAR-100-20 drops slightly. Compared with the baseline method $\text{SCAN}_{\text{MoCo}}$, SPICE improves ACC, NMI, and ARI by 6.5%, 9.4%, and 11.5% on STL10, by 4.4%, 6.4%, and 8.0% on CIFAR-10, and by 8.0%, 9.3%, and 9.4% on CIFAR-100-20, under the exactly the same setting. Without joint learning, SPICE_{s} already performs better than $\text{SCAN}_{\text{MoCo}}$ that contains the pretraining, clustering, and finetuning stages on STL10 and CIFAR100-20. Moreover, we evaluate the SPICE using larger backbone networks as used in the whole dataset setting, the results on STL10 and CIFAR-10 are very similar while the results on CIFAR-100-20 are significantly improved. Remarkably, SPICE significantly reduces the gap between unsupervised and supervised classification. On the STL10 that includes both labeled and unlabeled images, the results of unsupervised methods are better than the supervised as the unsupervised methods can leverage the unlabeled images for representation learning while the supervised cannot. On CIFAR-10 and CIFAR-100-20, all methods used the same images for training and the proposed SPICE further reduces the performance gap compared with the supervised counterpart, particularly, only 2% for ACC gap on CIFAR-10.

Table IV provides more detailed comparison results on STL10, where all these three methods use MoCo [49] for representation learning and were conducted five times for computing the mean and standard deviation of the results. Compared with $\text{SCAN}^{*}_{\text{MoCo}}$ that explores the instance similarity only without fine-tuning, SPICE_{s} explicitly leverages both the instance similarity and semantic discrepancy for learning clusters. On the other hand, different from k-means that infers the cluster labels with cluster centers, SPICE_{s} uses the nonlinear clustering head to predict the cluster labels. It can be seen that SPICE_{s} is significantly better than MoCo+k-means and $\text{SCAN}^{*}_{\text{MoCo}}$ in terms of both the mean and standard deviation metrics, demonstrating the superiority of the proposed prototype pseudo-labeling algorithm. The final results of SPICE is significantly better than those of $\text{SCAN}_{\text{MoCo}}$ in terms of mean performance and the stability.

Overall, our comparative results systematically demonstrate the superiority of the proposed SPICE method on both the whole and split dataset settings.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

D. Semi-Supervised Classification Comparison

In this subsection, we further compare SPICE with the recently proposed semi-supervised learning methods including II-Model [28], Pseudo-Labeling [77], Mean Teacher [78] Mix-Match [32], UDA [79], ReMinMatch [58], and FixMatch [59], as shown in Table V. Here the semi-supervised learning methods use 250 and 1,000 samples with ground truth labels on CIFAR-10 and STL10, respectively. Here the semi-supervised learning results on the commonly used CIFAR-10 and STL10 datasets are from [59]. SPICE used the same backbone network in FixMatch for a fair comparison and was conducted five times for reporting the mean and standard deviation results. It is found in our experiments that SPICE is comparable to and even better than these state-of-the-art semi-supervised learning methods. Actually, these results demonstrate that SPICE_S with the reliable pseudo-labeling algorithm can accurately label a set of images without human interaction.

E. Unsupervised Representation Learning

Here we aim to study the effects of the reliable pseudolabeling-based joint training in SPICE on representation features. As in all unsupervised representation learning methods [49], the quality of representation features is evaluated by training a linear classifier while freezing the learned feature model using the ground-truth labels. The results in Table VI show that the quality of representation features on CIFAR-10 and STL10 is clearly improved after joint training, while keeping similar performance on CIAFR-100-20. The interpretation is that the feature improvement with joint learning requires very accurate pseudo-labels (the ACC of reliable labels on STL10 and CIFAR-10 are 97.7% and 96.5%), so that the improvement is not observed on CIFAR-100-20 (the ACC of reliable labels on CIFAR-100-20 is 67.7%). In comparison with SCAN, SPICE also achieved better results in the unsupervised representation learning. Thus, given more prior of the target dataset, *e.g.*, the number of different semantic clusters with a roughly balanced distribution, the proposed framework has the potential to improve the unsupervised representation learning by generating accurate and reliable pseudo-labels.

F. Empirical Analysis

In this subsection, we empirically analyze the effectiveness of different components and options in the proposed SPICE framework.

1) Visualization of cluster semantics: We visualize the semantic clusters learned by SPICE_S in terms of the prototype samples and the discriminative regions, as shown in Fig. 5. Specifically, Fig. 5(a) shows the top three nearest samples of the cluster centers representing the cluster prototypes, and in Fig. 5(b), each cluster includes three samples and each sample is visualized with an original image and an attention map to highlight the discriminative regions. The attention maps of each cluster is computed by computing the cosine similarity between the cluster center (with Eqs. (2) and (3)) of the whole dataset and the convolutional feature maps of individual images, and then resized and normalized into [0, 1]. It shows that the prototype samples exactly match the human annotations, and the discriminative regions focus on the semantic objects. For example, the cluster with label ‘1’ captures the ‘dog’ class, and its most discriminative regions exactly capture the dogs at different locations,

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

and similar results can be observed for all other clusters. The visual results indicate that semantically meaningful clusters are learned, and the cluster center vectors can extract the discriminative features.

2) Ablation study: We evaluate the effectiveness of different components of SPICE_S in an ablation study, as shown in Table VII. In each experiment, we replaced one component of SPICE_S with another option, and five trials were conducted to report the mean and standard deviation for each metric.

We first evaluated the effectiveness of the overlap assignment and non-overlap assignment as described in Section III-B and Fig. 4. The results show that the overlap assignment is preferred over the non-overlap assignment, which may be explained by the fact that the non-overlap assignment may introduce extra local inconsistency when assigning the label to a sample far away from the cluster center, as shown in Fig. 4 (dashed red circle).

During training SPICE_S, we only optimize the clustering head while freezing the parameters of the feature model. To demonstrate the effectiveness of this separate training strategy, we compared it with two variants, *i.e.*, jointly training the feature model and a single clustering head (Joint-SH) and jointly training the feature model and multiple clustering heads (Joint-MH). Note that the feature model in the first branch in Fig. 3(b) is still fixed for accurately measuring the similarity. The results show that the clustering performance for these two variants became significantly worse. on the one hand, the quality of pseudo labels not only depends on the similarity measurement but also the predictions of the clustering head. On the other hand, the performance of the clustering head is also determined by the quality of representation features. When tuning all parameters during training, the feature model tends to be degraded without accurate labels and the clustering head tends to output incorrect predictions in the initial stage, which will harm the pseudo-labeling quality and get trapped in a bad cycle.

Usually, maximizing the entropy over different clusters is the necessary to avoid assign all samples into a single or a few clusters, as demonstrated in GATCluster [14]. Thus, we added another entropy loss during training, and the results were not changed. It indicates that our pseudo-labeling process with balance assignment has the ability to prevent trivial solutions. However, when clustering a large number of hierarchical clusters, *e.g.* 200 clusters in Tiny-ImageNet, the entropy loss was found necessary to avoid the empty clusters.

To evaluate the effectiveness of applying double softmax before CE, we replaced it with the plain CE or the CE with a temperature parameter (TCE) [80]. For TCE, we evaluated different temperature values including 0.01, 0.05, 0.07, 0.1, 0.2, 0.5, 0.8, 2, and found 0.2 achieve the best results that were included for comparison. The results show that TCE is better than CE, which is consistent with the literature [49]. On the other hand, the results of TCE are inferior to that with the double softmax CE. These experimental results are consistent with the detailed analysis in terms of the derivatives in Appendix A.

3) Clustering head selection: In unsupervised learning, the training process is hard to converge to the best state without the ground truth supervision, such that usually there

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

is a large standard deviation among different trials. Thus, how to estimate the performance of models in the unsupervised training process to select the potential best model is very important. As introduced in Subsection III-B, we use the clustering loss defined in Eq. (5) on the whole test dataset to approximate the clustering performance; *i.e.*, the smaller loss, the better clustering performance. The model selection process is shown in Fig. 6, it can be seen that the performance of the selected clustering head is very close to that of the ground truth selection, proving the effectiveness of this loss metric. In this way, the bad performance clustering heads can be filtered out. The results in Table IV show that SPICE has a lower standard deviation compared with the competing methods. Importantly, the lightweight clustering heads can be independently and simultaneously trained without affecting each other and without extra training time.

4) Effect of reliable labeling: In Section III-C, we introduced a reliable pseudo-labeling algorithm to select the reliably labeled images. Here we show the effectiveness of this algorithm in Fig. 7, where t-SNE was used to map the representation features of images in CIFAR-10 to 2D vectors for visualization. In Fig. 7(a), some obvious semantically inconsistent samples are evident, and the predicted ACC of SPICE_S on all samples is 83.8%. Using these samples directly for joint training, the ACC is not boosted. Fig. 7(b) shows the selected reliable samples, where the ratio of local inconsistency samples is significantly decreased, and the ACC is increased to 95.9% correspondingly. Using the reliable samples for joint training, the ACC of SPICE after joint training is significantly boosted (92.6% v.s. 83.8%) compared with that of SPICE_S.

5) Effect of data augmentation: We evaluated the effects of different data augmentations on SPICE_S, as shown in Table VIII, where Aug1 and Aug2 correspond to data augmentations of the second and the third branches in Fig. 3(b). The results show that when the second branch used the weak augmentation and the third branch used the strong augmentation, the model achieved the best performance. Moreover, the model had relatively worse performance when the second branch in labeling process uses the strong augmentation, which is due to that the labeling process aims to generate reliable pseudo labels that will be compromised by the strong augmentation. The model performs better when the third branch used the strong augmentation, as it will drive the model to output consistent predictions of different transformations. Overall, the data augmentation has a small impact on the results, because the pre-trained feature model had been already equipped with the transformation invariance ability.

G. Performance on Imbalanced Dataset—This study assumes that the target image dataset can be clustered into a set of balanced groups, and thus the pseudo-labels in Eq. (2) are generated in a balanced manner. Here we aim to evaluate the performance and robustness of SPICE on imbalanced datasets. To this end, we simulated imbalanced datasets from the training set of CIFAR10. Specifically, we randomly selected a subset of images for each class k according to a predefined ratio ζ_k. Then a set of ratios for different classes are calculated in a linear way; *i.e.*, $\zeta_k = \zeta + k \times (1 - \zeta)/K$, where $k = 1, 2, \ldots, K$, the number of clusters $K = 10$ on the CIFAR10 dataset, and the minimum ratio ζ defines the imbalanced level. We generated five levels of imbalanced datasets with $\zeta = 1.0, 0.8,$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

0.6, 0.4, 0.2 respectively, where 1.0 means a balanced dataset and the smaller ζ the more imbalanced. The cluster distributions for $\zeta = 0.4$ and 0.2 are shown in Fig. 8. Both SPICE and SCAN methods using MoCo representations were evaluated and the results are reported in Table IX. Similar to the results on balanced datasets, SPICE achieved consistently better performance on different levels of imbalanced datasets than SCAN. Although using a balanced assignment, SPICE performs relatively well for $\zeta \geq 0.4$ and shows robustness to some degree. However, the clustering performance is degraded on the highly imbalanced dataset $\zeta = 0.2$. The clustering distributions shown in Fig. 8 further illustrate the numerical results. Therefore, SPICE still needs to be improved further in the case of the imbalanced dataset.

V. Discussions

In this study, the SPICE network significantly improves the image clustering performance over the competing methods and reduces the performance gap between unsupervised and fully-supervised classification. However, there are opportunities for further refinements. First, existing deep clustering methods assume the clustering number K is known. In real applications, we do not always have such a prior. Therefore, how to automatically determine the number of semantically meaningful clusters is an open problem for deep clustering research. Second, to avoid trivial solutions, almost all existing clustering methods assume that the target dataset contains a similar number of samples in each and every cluster, which may or may not be the case in a real-world application. usually, there are at least two constraints that can be applied to implement this prior, including maximizing the entropy [14] and balancing assignment [77] (and SPICE) that is an optimal solution for maximizing entropy. On the other hand, if we do have a prior distribution of samples as a function of the cluster index, the constraints for these methods can be adapted from the uniform distribution to a specific one. The ideal clustering method should work well when neither the number of clusters nor the prior distribution of samples per cluster is known, which is a holy grail in this field. Finally, although the SPICE method achieved the superior results over the existing methods, the progressive training process through the three stages is computationally complicated based on multiple algorithmic ingredients. We recognize new opportunities to reduce the complexity and improve the performance of SPICE in a unified framework. For example, we could simultaneously train the feature model and clustering head in a single stage by combining the proposed semantic pseudo-labeling and the recently proposed representation learning methods [44] [81] that do not require any specific network architecture or particular optimization algorithm, being compatible with the current pseudo-labeling algorithm. Also, the single prototype pseudolabeling could be extended to multiple prototypes as studied in [36], [37] for further improvement. Nevertheless, our method is both effective and efficient, and easy to optimize and apply, because each training stage only has a single cross-entropy function and is fairly straightforward. Importantly, SPICE is an exemplary workflow to synergize both the instance similarity and semantic discrepancy for superior image clustering.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

VI. Conclusion

We have presented a semantic pseudo-labeling framework for image clustering, with the acronym "SPICE". To accurately measure both the similarity among samples and the discrepancy between clusters for clustering, we divide the clustering network into a feature model and a clustering head, which are first trained separately with the unsupervised representation learning algorithm and the prototype pseudo-labeling algorithm, and then jointly trained with the reliable pseudolabeling algorithm. Extensive experiments have demonstrated the superiority of SPICE over the competing methods on balanced close-set benchmarks with an average performance boost of 10% in terms of adjusted rand index, normalized mutual information, and clustering accuracy. The SPICE is comparable to or even better than the state-of-the-art semi-supervised learning methods and has the ability to improve the representation features. Remarkably, SPICE significantly reduces the gap between unsupervised and fully-supervised classification; *e.g.*, only 2% gap on CIFAR-10. We believe the basic idea behind SPICE has the potential to help cluster other domain datasets, and apply to other learning tasks.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

This work was supported in part by Natural Science Foundation of Shanghai (21ZR1403600), National Natural Science Foundation of China (62101136), Shanghai Municipal Science and Technology Major Project (2018SHZDZX01), ZJ Lab, Shanghai Municipal of Science and Technology Project (20JC1419500), Shanghai Sailing Program (21YF1402800), Shanghai Center for Brain Science and Brain-inspired Technology, NIH under Award numbers R01CA237267, R01HL151561, R01EB031102, and R01EB032716.

Biographies

Chuang Niu (Member, IEEE) is currently a Postdoctoral Research Associate with Department of Biomedical Engineering, Center for Biotechnology and Interdisciplinary Studies, Rensselaer Polytechnic Institute. He serves as an Associate Director in AXIS Lab at Rensselaer Polytechnic Institute. He received his B.S. in biomedical engineering in 2015, and his Ph.D. in pattern recognition and machine intelligence in 2020, from Xidian University. He was a visiting student at Rensselaer Polytechnic Institute from 2019 to 2020. His research interests include unsupervised/self-supervised learning, weakly-supervised learning, representation learning, clustering, and biomedical imaging and analysis.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Hongming Shan (Senior Member, IEEE) is currently an Associate Professor with the Institute of Science and Technology for Brain-inspired Intelligence, Fudan University, and also a "Qiusuo" Research Leader with the Shanghai Center for Brain Science and Brain-inspired Technology. He received the Ph.D. degree in machine learning from Fudan University in 2017. From 2017 to 2020, he was a Postdoctoral Research Associate and Research Scientist at Rensselaer Polytechnic Institute, USA. His research interests include machine learning, medical imaging, and computer vision. He was recognized with Youth Outstanding Paper Award at World Artificial Intelligence Conference 2021.

Ge Wang (Fellow, IEEE) is Clark-Crossan Chair Professor and Director of Biomedical Imaging Center, RPI, USA. He focuses on AI-based medical imaging. He published the first spiral cone-beam CT method in the 1990s. There are 200 million CT scans yearly, with a majority in the spiral cone-beam mode. He published the first perspective on deep imaging in 2016 and many follow-up papers. He is Fellow of IEEE, SPIE, AAPM, OSA, AIMBE, AAAS, and NAI, and recognized with various other honors such as IEEE R1 Outstanding Teaching Award, EMBS Career Achievement Award, SPIE Meinel Technology Award, and Sigma Xi Chubb Award for Innovation.

References

[1]. Van Gansbeke W, Vandenhende S, Georgoulis S, Proesmans M, and Van Gool L, "SCAN: Learning to classify images without labels," in ECCV, 2020, pp. 268–285.

[2]. Xie J, Girshick R, and Farhadi A, "Unsupervised deep embedding for clustering analysis," in ICML, 2016, pp. 478–487.

[3]. Li F and et al. , "Discriminatively boosted image clustering with fully convolutional auto-encoders," PR, vol. 83, pp. 161 – 173, 2018.

[4]. Yang B, Fu X, Sidiropoulos ND, and Hong M, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in ICML, vol. 70, 2017, pp. 3861–3870.

[5]. Tian K and et al. , "DeepCluster: A general clustering framework based on deep learning," in ECML PKDD, 2017, pp. 809–825.

[6]. Zhang J, Li C-G, You C, Qi X, Zhang H, Guo J, and Lin Z, "Self-supervised convolutional subspace clustering network," in CVPR, 2019.

[7]. Dizaji KG, Herandi A, Deng C, Cai W, and Huang H, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in ICCV, 2017, pp. 5747–5756.

[8]. Jiang Z, Zheng Y, Tan H, Tang B, and Zhou H, "Variational deep embedding: An unsupervised and generative approach to clustering," in IJCAI, 2017, pp. 1965–1972.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

[9]. Dilokthanakul N and et al. , “Deep unsupervised clustering with Gaussian mixture variational autoencoders,” arXiv:1611.02648, 2017.

[10]. Zhou P, Hou Y, and Feng J, “Deep adversarial subspace clustering,” in CVPR, 2018.

[11]. Chang J, Wang L, Meng G, Xiang S, and Pan C, “Deep adaptive image clustering,” in ICCV, 2017, pp. 5880–5888.

[12]. Ji X, Henriques JF, and Vedaldi A, “Invariant information clustering for unsupervised image classification and segmentation,” in ICCV, 2019.

[13]. Wu J, Long K, Wang F, Qian C, Li C, Lin Z, and Zha H, “Deep comprehensive correlation mining for image clustering,” in ICCV, 2019.

[14]. Niu C and et al. , “GATCluster: Self-supervised Gaussian-attention network for image clustering,” in ECCV, 2020, pp. 735–751.

[15]. Li Y, Hu P, Liu Z, Peng D, Zhou JT, and Peng X, “Contrastive clustering,” in AAAI, 2021.

[16]. Tao Y and et al. , “Clustering-friendly representation learning via instance discrimination and feature decorrelation,” in ICLR, 2021.

[17]. Vincent P and et al. , “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” JMLR, vol. 11, no. 12, pp. 3371–3408, 2010.

[18]. Huang P, Huang Y, Wang W, and Wang L, “Deep embedding network for clustering,” in ICPR, 2014, pp. 1532–1537.

[19]. Chen D, Lv J, and Zhang Y, “Unsupervised multi-manifold clustering by learning deep representation.” in AAAI Workshops, 2017.

[20]. Ji P, Zhang T, Li H, Salzmann M, and Reid I, “Deep subspace clustering networks,” in NeurIPS, 2017, pp. 23–32.

[21]. Yang J, Parikh D, and Batra D, “Joint unsupervised learning of deep representations and image clusters,” in CVPR, 2016.

[22]. Chidananda Gowda K and Krishna G, “Agglomerative clustering using the concept of mutual nearest neighbourhood,” Pattern Recognition, vol. 10, no. 2, pp. 105–112, 1978.

[23]. Hong S, Choi J, Feyereisl J, Han B, and Davis LS, “Joint image clustering and labeling by matrix factorization,” TPAMI, vol. 38, no. 7, pp. 1411–1424, 2016.

[24]. Chang J, Meng G, Wang L, Xiang S, and Pan C, “Deep self-evolution clustering,” TPAMI, vol. 42, no. 4, pp. 809–823, 2020.

[25]. Huang J, Gong S, and Zhu X, “Deep semantic clustering by partition confidence maximisation,” in CVPR, 2020.

[26]. Jabi M, Pedersoli M, Mitiche A, and Ayed IB, “Deep clustering: On the link between discriminative models and k-means,” TPAMI, vol. 43, no. 6, pp. 1887–1896, 2021.

[27]. Gupta D, Ramjee R, Kwatra N, and Sivathanu M, “Unsupervised clustering using pseudo-semi-supervised learning,” in International Conference on Learning Representations, 2020.

[28]. Rasmus A, Berglund M, Honkala M, Valpola H, and Raiko T, “Semi-supervised learning with ladder networks,” in NeurIPS, vol. 28, 2015.

[29]. Han S and et al. , “Mitigating embedding and class assignment mismatch in unsupervised image classification,” in ECCV, 2020, pp. 768–784.

[30]. Dang Z, Deng C, Yang X, Wei K, and Huang H, “Nearest neighbor matching for deep clustering,” in CVPR, 2021, pp. 13 693–13 702.

[31]. Park S, Han S, Kim S, Kim D, Park S, Hong S, and Cha M, “Improving unsupervised image clustering with robust learning,” in CVPR, 2021, pp. 12278–12287.

[32]. Berthelot D and et al. , “MixMatch: A holistic approach to semi-supervised learning,” in NeurIPS, vol. 32, 2019.

[33]. Li J, Socher R, and Hoi SC, “DivideMix: Learning with noisy labels as semi-supervised learning,” in ICLR, 2020.

[34]. Lukasik M and et al. , “Does label smoothing mitigate label noise?” in ICML, vol. 119, 2020, pp. 6448–6458.

[35]. Mollineda R, Ferri F, and Vidal E, “An efficient prototype merging strategy for the condensed 1-nn rule through class-conditional hierarchical clustering,” Pattern Recognition, vol. 35, no. 12, pp. 2771–2782, 2002.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

[36]. Liu M, Jiang X, and Kot AC, “A multi-prototype clustering algorithm,” Pattern Recognition, vol. 42, no. 5, pp. 689–698, 2009.

[37]. Yang H-M, Zhang X-Y, Yin F, and Liu C-L, “Robust classification with convolutional prototype learning,” in CVPR, 2018, pp. 3474–3482.

[38]. Wang Q, Lian H, Sun G, Gao Q, and Jiao L, “icmsc: Incomplete cross-modal subspace clustering,” TIP, vol. 30, pp. 305–317, 2021.

[39]. Lv J, Kang Z, Lu X, and Xu Z, “Pseudo-supervised deep subspace clustering,” TIP, vol. 30, pp. 5252–5263, 2021.

[40]. Qin Y, Wu H, Zhang X, and Feng G, “Semi-supervised structured subspace learning for multi-view clustering,” TIP, vol. 31, pp. 1–14, 2022.

[41]. Huang A, Chen W, Zhao T, and Chen CW, “Joint learning of latent similarity and local embedding for multi-view clustering,” TIP, vol. 30, pp. 6772–6784, 2021.

[42]. Yan M, Chen Y, Chen Y, Zeng G, Hu X, and Du J, “A lightweight weakly supervised learning segmentation algorithm for imbalanced image based on rotation density peaks,” Knowledge-Based Systems, vol. 244, p. 108513, 2022.

[43]. Grill J-B and et al. , “Bootstrap your own latent - a new approach to self-supervised learning,” in NeurIPS, vol. 33, 2020, pp. 21 271–21 284.

[44]. Niu C and Wang G, “Self-supervised representation learning with multi-segmental informational coding (music),” arXiv preprint arXiv:2206.06461, 2022.

[45]. Zhang R, Isola P, and Efros AA, “Colorful image colorization.” in ECCV, vol. 9907. Springer, 2016, pp. 649–666.

[46]. Gidaris S, Singh P, and Komodakis N, “Unsupervised representation learning by predicting image rotations,” in ICLR, 2018.

[47]. Noroozi M and Favaro P, “Unsupervised learning of visual representations by solving jigsaw puzzles.” in ECCV, 2016, pp. 69–84.

[48]. Chen T, Kornblith S, Norouzi M, and Hinton G, “A simple framework for contrastive learning of visual representations,” in ICML, vol. 119, 2020, pp. 1597–1607.

[49]. He K, Fan H, Wu Y, Xie S, and Girshick R, “Momentum contrast for unsupervised visual representation learning,” in CVPR, 2020.

[50]. Caron M, Bojanowski P, Joulin A, and Douze M, “Deep clustering for unsupervised learning of visual features.” in ECCV, vol. 11218, 2018, pp. 139–156.

[51]. Asano YM, Rupprecht C, and Vedaldi A, “Self-labelling via simultaneous clustering and representation learning,” in ICLR, 2020.

[52]. Cuturi M, “Sinkhorn distances: Lightspeed computation of optimal transport,” in NeurIPS, vol. 26, 2013.

[53]. Caron M and et al. , “Unsupervised learning of visual features by contrasting cluster assignments,” in NeurIPS, 2020.

[54]. Sajjadi M, Javanmardi M, and Tasdizen T, “Regularization with stochastic transformations and perturbations for deep semi-supervised learning,” in NeurIPS, 2016.

[55]. Laine S and Aila T, “Temporal ensembling for semi-supervised learning,” in ICLR, 2017.

[56]. Lee D-H, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in ICML Workshop on Challenges in Representation Learning, 2013.

[57]. Grandvalet Y and Bengio Y, “Semi-supervised learning by entropy minimization,” in NeurIPS, vol. 17, 2005.

[58]. Berthelot D, Carlini N, Cubuk ED, Kurakin A, Sohn K, Zhang H, and Raffel C, “ReMixMatch: Semi-supervised learning with distribution matching and augmentation anchoring,” in ICLR, 2020.

[59]. Sohn K and et al. , “FixMatch: Simplifying semi-supervised learning with consistency and confidence,” in NeurIPS, vol. 33, 2020, pp. 596–608.

[60]. Wu Z, Xiong Y, Stella XY, and Lin D, “Unsupervised feature learning via non-parametric instance discrimination,” in CVPR, 2018.

[61]. Macqueen J, “Some methods for classification and analysis of multivariate observations,” in In 5-th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

[62]. Hubert L and Arabie P, “Comparing partitions,” Journal of classification, vol. 2, no. 1, pp. 193–218, 1985.

[63]. Strehl A and Ghosh J, “Cluster ensembles – a knowledge reuse framework for combining multiple partitions,” Journal on Machine Learning Research, vol. 3, pp. 583–617, December 2002.

[64]. Li T and Ding CHQ, “The relationships among various nonnegative matrix factorization methods for clustering.” in ICDM. IEEE Computer Society, 2006, pp. 362–371.

[65]. Ng AY, Jordan MI, and Weiss Y, “On spectral clustering: Analysis and an algorithm,” in NeurIPS, 2002, pp. 849–856.

[66]. Franti P, Virmajoki O, and Hautamaki V, “Fast agglomerative clustering using a k-nearest neighbor graph,” TPAMI, vol. 28, no. 11, pp. 1875–1881, Nov 2006.

[67]. Cai D, He X, Wang X, Bao H, and Han J, “Locality preserving nonnegative matrix factorization,” in IJCAI, 2009, pp. 1010–1015.

[68]. Bengio Y, Lamblin P, Popovici D, Larochelle H, and Montreal U, “Greedy layer-wise training of deep networks,” in NeurIPS, vol. 19, 2007, pp. 153–160.

[69]. Radford A, Metz L, and Chintala S, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2016.

[70]. Zeiler MD, Krishnan D, Taylor GW, and Fergus R, “Deconvolutional networks,” in CVPR, 2010.

[71]. Kingma DP and Welling M, “Auto-encoding variational bayes,” 2014.

[72]. Chang J, Guo Y, Wang L, Meng G, Xiang S, and Pan C, “Deep discriminative clustering analysis,” in arXiv:1905.01681, 2019.

[73]. He K, Zhang X, Ren S, and Sun J, “Deep residual learning for image recognition,” in CVPR, 2016, pp. 770–778.

[74]. DeVries T and Taylor GW, “Improved regularization of convolutional neural networks with cutout,” arXiv preprint arXiv:1708.04552, 2017.

[75]. Cubuk ED, Zoph B, Shlens J, and Le Q, “RandAugment: Practical automated data augmentation with a reduced search space,” in NeurIPS, vol. 33, 2020, pp. 18 613–18 624.

[76]. Haeusser P, Plapp J, Golkov V, Aljalbout E, and Cremers D, “Associative deep clustering: Training a classification network with no labels,” in Pattern Recognition, 2019, pp. 18–32.

[77]. Lee D-H, “Pseudo-Label : The simple and efficient semi-supervised learning method for deep neural networks,” in Workshop on challenges in ICML, 2013.

[78]. Tarvainen A and Valpola H, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in NeurIPS, 2017, p. 11951204.

[79]. Xie Q, Dai Z, Hovy EH, Luong M, and Le QV, “Unsupervised data augmentation,” CoRR, vol. abs/1904.12848, 2019.

[80]. Hinton G, Vinyals O, and Dean J, “Distilling the knowledge in a neural network,” in NIPS Deep Learning and Representation Learning Workshop, 2015.

[81]. Niu C and Wang G, “Home: High-order mixed-moment-based embedding for representation learning,” arXiv preprint arXiv:2207.07743, 2022.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

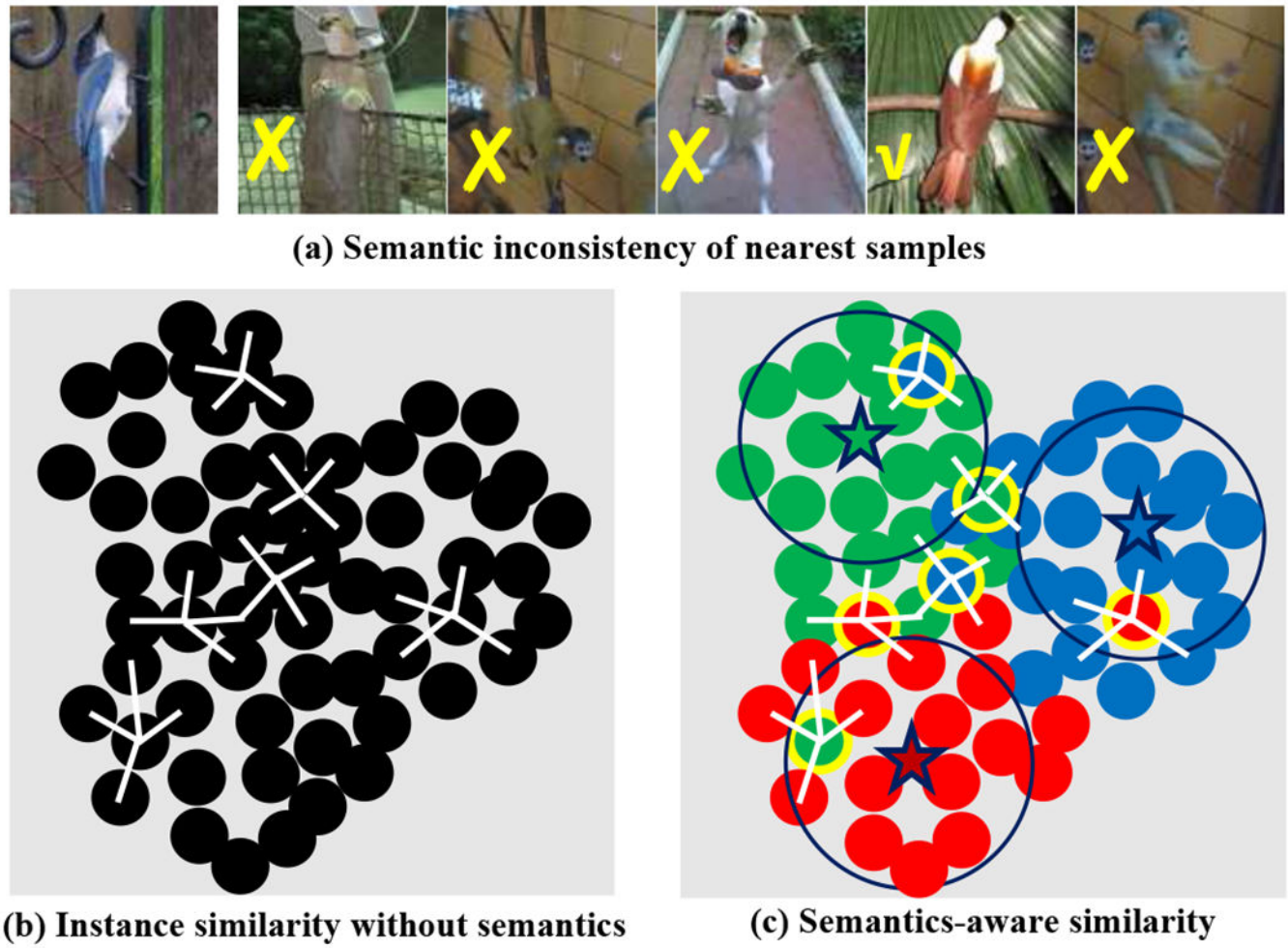

Fig. 1.

Semantic relevance in the feature space. (a) Neighboring samples of different semantics, where the first image is the query image and the other images are the nearest images with the closest features provided by SCAN [1]. (b) Instance similarity without semantics, where each point denotes a sample in the feature space, and white lines link a sample to its four nearest similar samples measured by cosine similarity. (c) Semantics-aware similarity, where different colors denote different semantics/classes that can be estimated through pseudo-labeling, stars denote cluster centers, the points within large black circles are similar to the cluster centers, and the points within yellow circles are semantically inconsistent to neighbor samples. By synergizing instance similarity and semantic discrepancy, more accurate and reliable supervision can be generated for deep clustering.

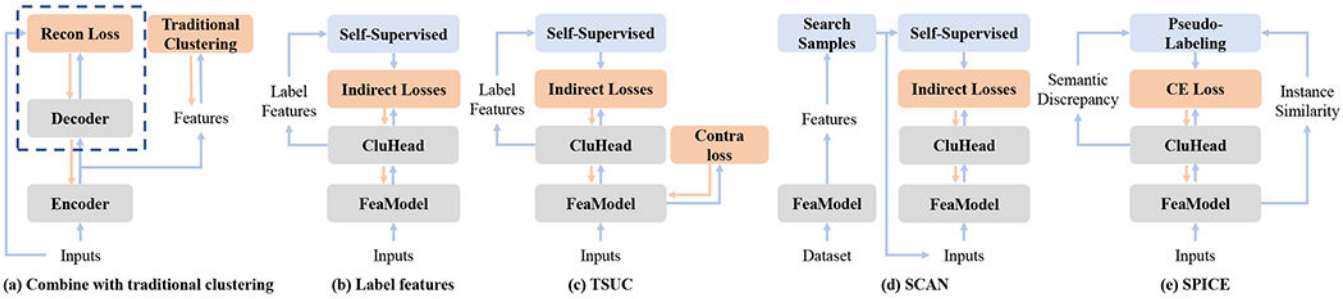

Fig. 2.

Training framework of different deep clustering methods. (a) The initial deep clustering methods that combine traditional clustering algorithms with the deep neural networks, most of them combine with the autoencoders and some combine with an encoder only. (b) The label feature based methods that directly map images to cluster labels, where self-supervision is calculated based on the label features only. (c) A two-stage unsupervised clustering method that first trains the feature model and then trains the whole clustering work with a label feature based self-supervision and an embedding feature based contrastive loss simultaneously and separately. (d) The SCAN learning framework that constrains the similar samples in the embedding space having the same cluster label. (e) The proposed SPICE framework that synergizes both the similarity among samples and the discrepancy between clusters for training a clustering network through semantics-aware pseudo-labeling. Note that the indirect losses here refer to measuring the similarity or distance between samples, the entropy over clusters, etc., while the cross-entropy (CE) loss directly computes the difference between predicted cluster labels and pseudo labels.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

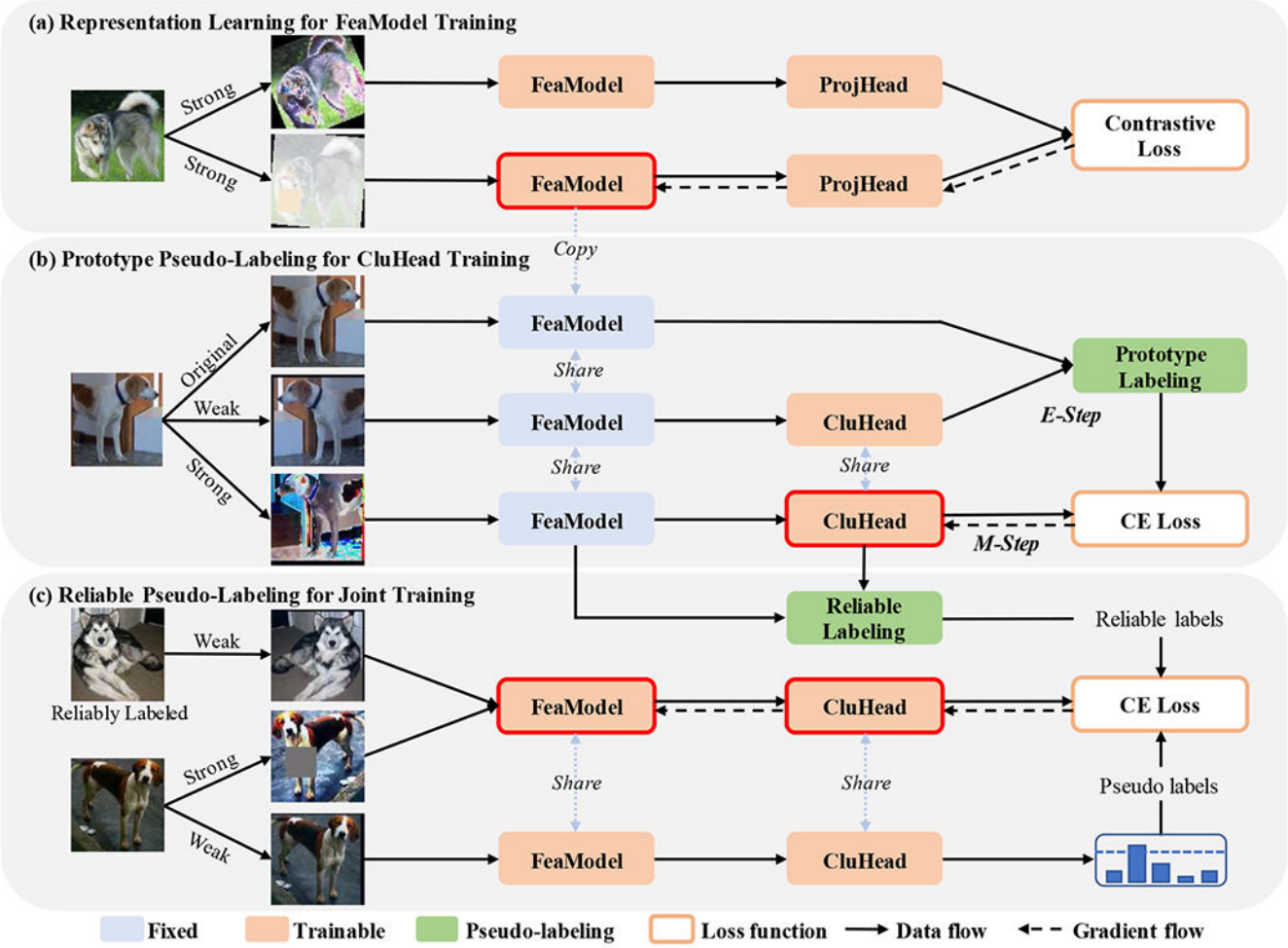

Fig. 3.

Illustration of the SPICE framework. (a) Train the feature model with contrastive learning based unsupervised representation learning. In this study, MoCo was implemented for representation learning. Note that the memory bank for storing negative samples is not shown, and other methods [43], [44] without requiring negative samples could be implemented here for better performance. (b) Train the clustering head via the prototype pseudo-labeling algorithm in an EM framework. (c) Jointly train the feature model and the clustering head through the reliable pseudo-labeling algorithm.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

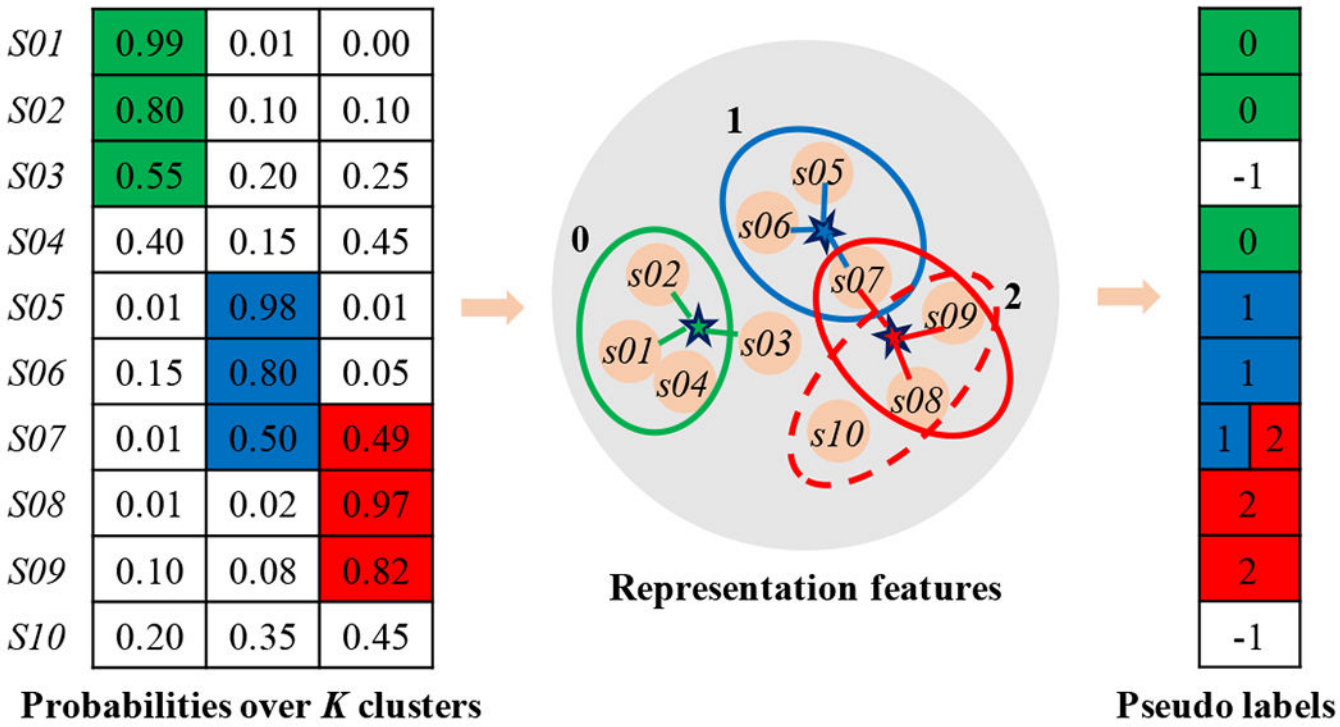

Fig. 4.

A toy example for prototype pseudo-labeling. First, given the predicted probabilities of 10 samples over 3 clusters, top 3 confident samples are selected for each cluster, marked as green, blue, and red colors respectively. Then, the selected samples are mapped into the corresponding features (denoted by dots) to estimate the prototypes (denoted by stars) for each cluster, where stars are estimated with connected dots. Finally, the top 3 nearest samples to each cluster prototype (the dots within the same ellipse) are selected and assigned with the index of the corresponding prototype. Other unselected samples are signed with −1 and will not be used for training. The dashed ellipse denotes the non-overlap assignment.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

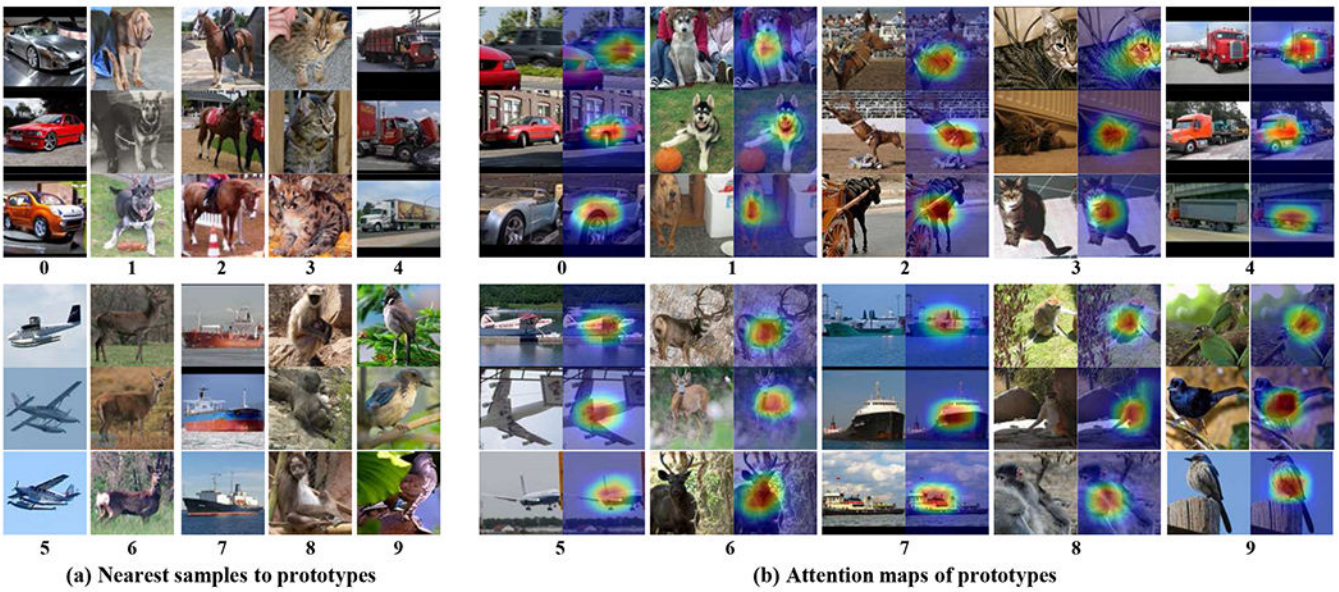

Fig. 5.

Visualization of learned semantic clusters on STL10. (a) The top three nearest samples to the cluster centers. (b) The attention maps of cluster center on individual images.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

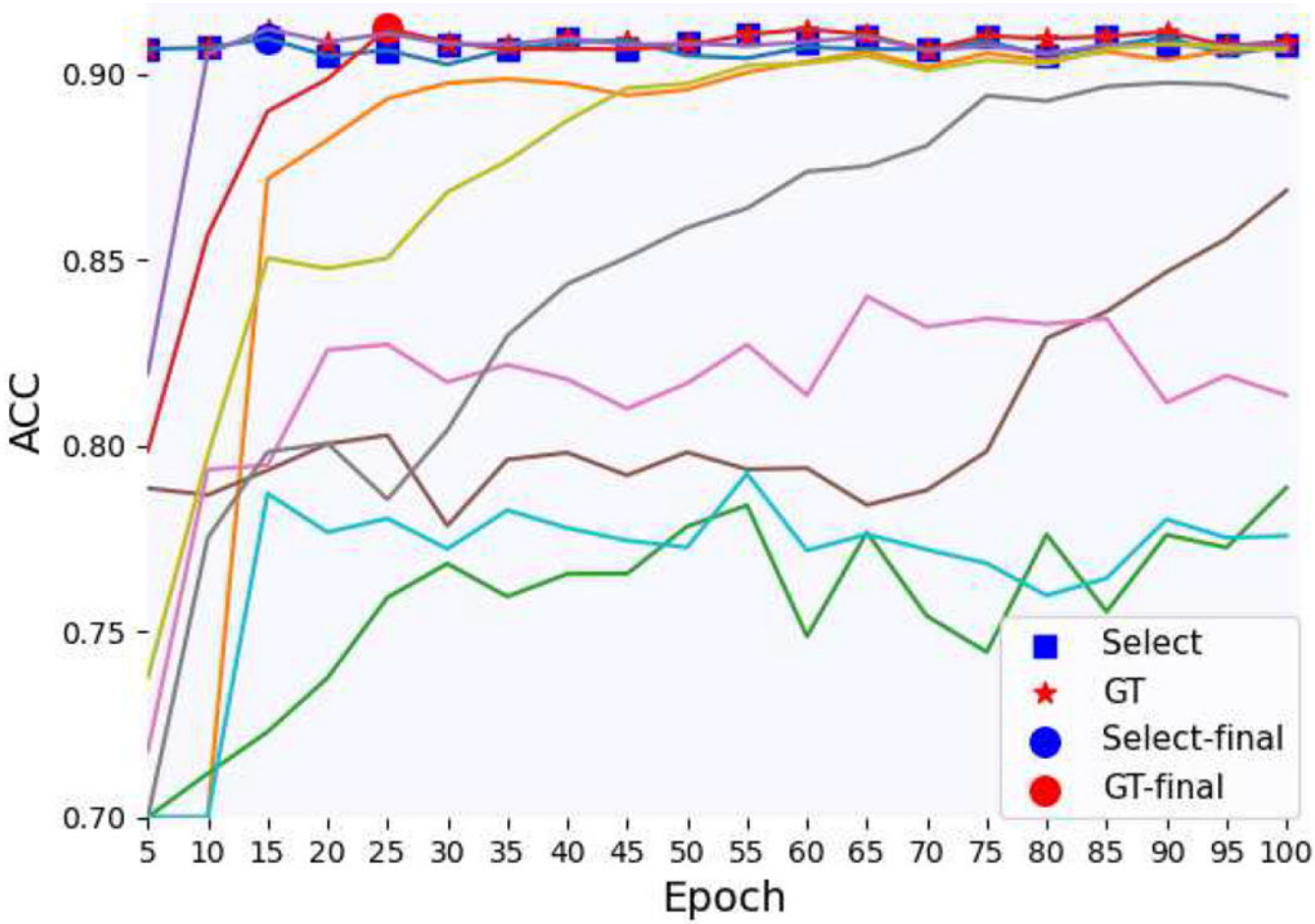

Fig. 6.

Clustering head selection. Each curve represents the changing process of ACC v.s. epoch of a specific clustering head. The blue squares mark the selected best clustering head for each epoch, and the red stars represent the corresponding best head evaluated with the ground truth. The blue circle denotes the finally selected head. The red circle is the ground truth best head.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

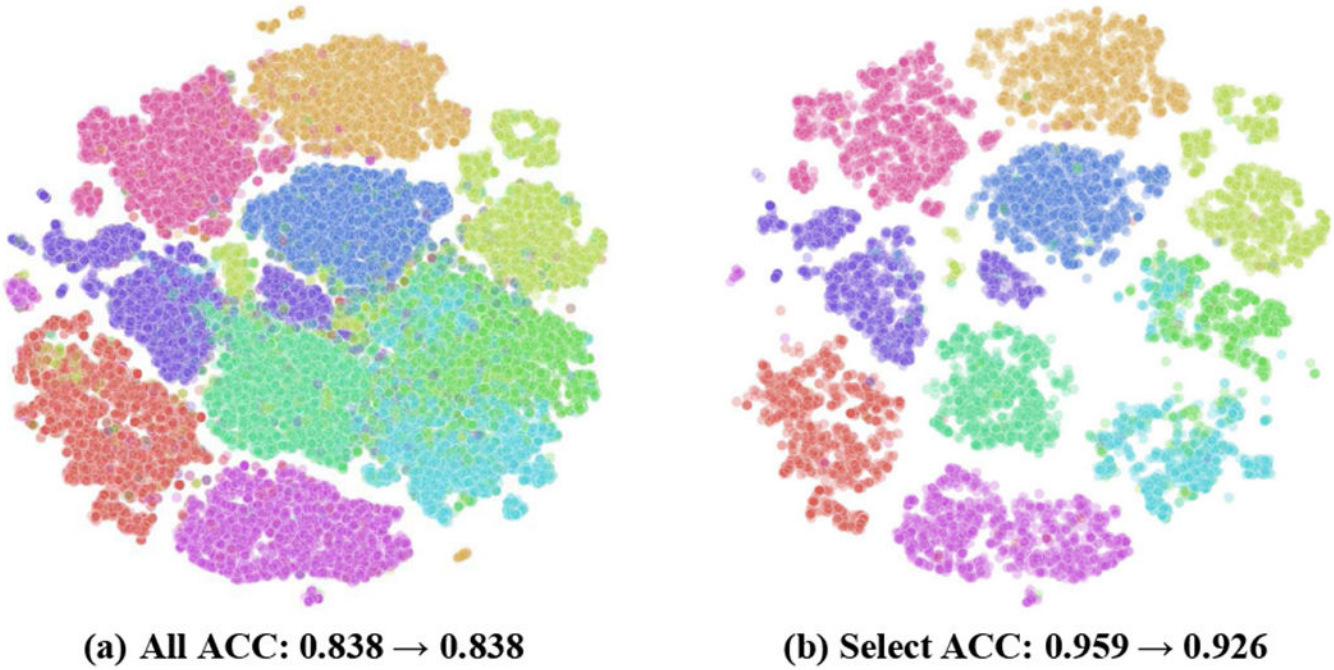

Fig. 7.

Visualization of reliable labels on CIFAR-10 dataset. Each point denotes a sample in the embedding space, different colors are rendered by the ground-truth labels. (a) The ACC of all samples is 0.838 and the ACC of the jointly trained model using all pseudo-labeled samples is also 0.838. (b) The ACC of the selected reliable labels is 0.959 and the ACC of the jointly trained model using the reliable samples is 0.926.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

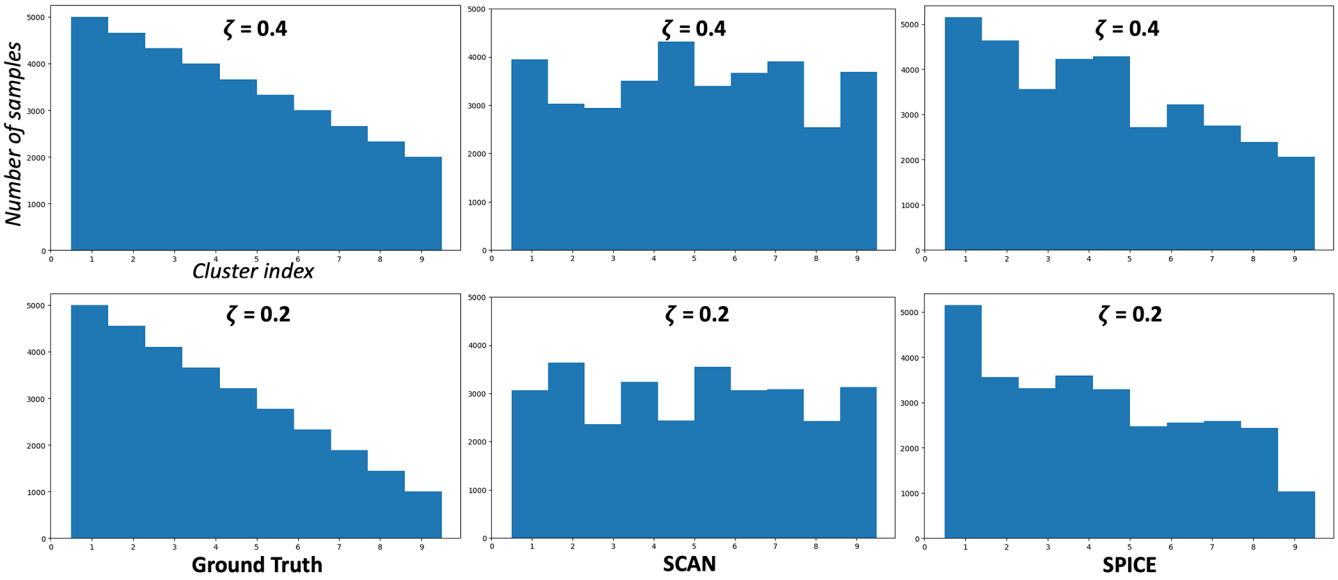

Fig. 8.

Cluster distributions on imbalanced datasets. From the first to third columns are the cluster distributions of ground-truth labels, SCAN predicted labels, and SPICE predicted labels on the most two imbalanced datasets (ζ=0.4 and 0.2). In each sub-figure, axes x and y represent the cluster index and the number of samples respectively.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE I

SPECIFICATIONS AND PARTITIONS OF SELECTED DATASETS.

Dataset	Image size	# Training	# Testing	# Classes (K)
STL10	96 × 96	5,000	8,000	10
CIFAR-10	32 × 32	50,000	10,000	10
CIFAR-100-20	32 × 32	50,000	10,000	20
ImageNet-10	224 × 224	13,000	N/A	10
ImageNet-Dog	224 × 224	19,500	N/A	15
Tiny-ImageNet	64 × 64	100,000	10,000	200

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

TABLE II

COMPARISON WITH COMPETING METHODS THAT WERE TRAINED AND TESTED ON THE WHOLE DATASET (TRAIN AND TEST SPLIT DATASETS ARE MERGED AS ONE). NOTE THAT SPICE WAS TRAINED ON THE TARGET DATASETS ONLY WITHOUT USING EXTRA DATA AND WITHOUT USING ANY ANNOTATIONS, WHICH ARE THE EXACTLY THE SAME AS DATA USED IN THE EXISTING METHODS. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**.

Method	STL10			ImageNet-10			ImageNet-Dog-15			CIFAR-10			CIFAR-100-20			Tiny-ImageNet-200		
	ACC	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**
k-means [61]	0.192	0.125	0.061	0.241	0.119	0.057	0.105	0.055	0.020	0.229	0.087	0.049	0.130	0.084	0.028	0.025	0.065	0.005
SpectralCluster [65]	0.159	0.098	0.048	0.274	0.151	0.076	0.111	0.038	0.013	0.247	0.103	0.085	0.136	0.090	0.022	0.022	0.063	0.004
AgglomCluster [66]	0.332	0.239	0.140	0.242	0.138	0.067	0.139	0.037	0.021	0.228	0.105	0.065	0.138	0.098	0.034	0.027	0.069	0.005
NMF [67]	0.180	0.096	0.046	0.230	0.132	0.065	0.118	0.044	0.016	0.190	0.081	0.034	0.118	0.079	0.026	0.029	0.072	0.005
AE [68]	0.303	0.250	0.161	0.317	0.210	0.152	0.185	0.104	0.073	0.314	0.239	0.169	0.165	0.100	0.048	0.041	0.131	0.007
SDAE [17]	0.302	0.224	0.152	0.304	0.206	0.138	0.190	0.104	0.078	0.297	0.251	0.163	0.151	0.111	0.046	0.039	0.127	0.007
DCGAN [69]	0.298	0.210	0.139	0.346	0.225	0.157	0.174	0.121	0.078	0.315	0.265	0.176	0.151	0.120	0.045	0.041	0.135	0.007
DeCNN [70]	0.299	0.227	0.162	0.313	0.186	0.142	0.175	0.098	0.073	0.282	0.240	0.174	0.133	0.092	0.038	0.035	0.111	0.006
VAE [71]	0.282	0.200	0.146	0.334	0.193	0.168	0.179	0.107	0.079	0.291	0.245	0.167	0.152	0.108	0.040	0.036	0.113	0.006
JULE [21]	0.277	0.182	0.164	0.300	0.175	0.138	0.138	0.054	0.028	0.272	0.192	0.138	0.137	0.103	0.033	0.033	0.102	0.006
DEC [2]	0.359	0.276	0.186	0.381	0.282	0.203	0.195	0.122	0.079	0.301	0.257	0.161	0.185	0.136	0.050	0.037	0.115	0.007
DAC [11]	0.470	0.366	0.257	0.527	0.394	0.302	0.275	0.219	0.111	0.522	0.396	0.306	0.238	0.185	0.088	0.066	0.190	0.017
DeepCluster [50]	0.334	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.374	N/A	N/A	0.189	N/A	N/A	N/A	N/A	N/A
DDC [72]	0.489	0.371	0.267	0.577	0.433	0.345	N/A	N/A	N/A	0.524	0.424	0.329	N/A	N/A	N/A	N/A	N/A	N/A
IIC [12]	0.610	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.617	N/A	N/A	0.257	N/A	N/A	N/A	N/A	N/A
DCCM [13]	0.482	0.376	0.262	0.710	0.608	0.555	0.383	0.321	0.182	0.623	0.496	0.408	0.327	0.285	0.173	0.108	0.224	0.038
DSEC [24]	0.482	0.403	0.286	0.674	0.583	0.522	0.264	0.236	0.124	0.478	0.438	0.340	0.255	0.212	0.110	0.066	0.190	0.017
GATCluster [14]	0.583	0.446	0.363	0.762	0.609	0.572	0.333	0.322	0.200	0.610	0.475	0.402	0.281	0.215	0.116	N/A	N/A	N/A
PICA [25]	0.713	0.611	0.531	0.870	0.802	0.761	0.352	0.352	0.201	0.696	0.591	0.512	0.337	0.310	0.171	0.098	0.277	0.040
CC [15]	0.850	0.746	0.726	0.893	0.859	0.822	0.429	0.445	0.274	0.790	0.705	0.637	0.429	0.431	0.266	0.140	0.340	0.071
IDFD [16]	0.756	0.643	0.575	0.954	0.898	0.901	0.591	0.546	0.413	0.815	0.711	0.663	0.425	0.426	0.264	N/A	N/A	N/A
SPICE$_s$	0.908	0.817	0.812	0.921	0.828	0.836	0.646	0.572	0.479	0.838	0.734	0.705	0.468	0.448	0.294	**0.305**	**0.449**	**0.161**
SPICE	**0.938**	**0.872**	**0.870**	**0.959**	**0.902**	**0.912**	**0.675**	**0.627**	**0.526**	**0.926**	**0.865**	**0.852**	**0.538**	**0.567**	**0.387**	0.291	0.427	0.147

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

TABLE III

Comparison with competing methods on split datasets (training and testing images are mutually exclusive). For a fair comparison, both $SCAN_{MoCo}$ and SPICE used MOCO for feature learning, and ResNet18 as backbone in all training stages. Here the best results for all methods were used for comparison. $SPICE_{RES34}$ used the ResNet34 as backbone. The best two unsupervised results are highlighted in **bold**.

Method	STL10			CIFAR-10			CIFAR-100-20		
	ACC	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**
ADC [76]	0.530	N/A	N/A	0.325	N/A	N/A	0.160	N/A	N/A
TSUC [29]	0.665	N/A	N/A	0.810	N/A	N/A	0.353	N/A	N/A
NNM [30]	0.808	0.694	0.650	0.843	0.748	0.709	0.477	0.484	0.316
SCAN [1]	0.809	0.698	0.646	0.883	0.797	0.772	0.507	0.486	0.333
RUC_{SCAN} [31]	0.867	N/A	N/A	0.903	N/A	N/A	0.533	N/A	N/A
$SCAN_{MoCo}$ [1]	0.855	0.758	0.721	0.874	0.786	0.756	0.455	0.472	0.310
$SPICE_s$	0.862	0.756	0.732	0.845	0.739	0.709	0.468	0.457	0.321
SPICE	**0.920**	**0.852**	**0.836**	**0.918**	**0.850**	**0.836**	**0.535**	**0.565**	**0.404**
$SPICE_{Res34}$	**0.929**	**0.860**	**0.853**	**0.917**	**0.858**	**0.836**	**0.584**	**0.583**	**0.422**
Supervised	0.806	0.659	0.631	0.938	0.862	0.870	0.800	0.680	0.632

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE IV

More detailed comparison results on STL10. Here all methods were trained and tested on the split train and test datasets respectively. Both the mean and standard deviation results were reported. Each method was conducted five times. Here all methods used the ResNet18 backbone, $\text{SCAN}_{\text{MoCo}}$ and SPICE used MoCo for feature learning with STL10 images only. $\text{SCAN}^{*}_{\text{MoCo}}$ means no self-labeling.

Method	ACC	NMI	ARI
Supervised	0.806	0.659	0.631
MoCo+*k*-means	0.797±0.046	0.768±0.021	0.624±0.041
$\text{SCAN}^{*}_{\text{MoCo}}$	0.787±0.036	0.697±0.026	0.639±0.041
$\text{SCAN}_{\text{MoCo}}$	0.797±0.034	0.701±0.032	0.649±0.044
SPICE_{s}	0.852±0.011	0.749±0.008	0.719±0.015
SPICE	0.918±0.002	0.849±0.003	0.836±0.002

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

TABLE V

Comparison results with semi-supervised learning on STL10 and CIFAR-10. Top-1 classification accuracy and standard deviation for different methods are reported. The best three results are in **bold**.

	Semi-supervised							Unsupervised	
Method	**Π-Model [28]**	**Pseudo-Labeling [77]**	**Mean Teacher [78]**	**MixMatch [32]**	**UDA [79]**	**ReMixMatch [58]**	**FixMatch [59]**	**SCAN [1]**	**SPICE (ours)**
STL10	0.748±0.008	0.720±0.008	0.786±0.024	0.896±0.006	**0.923±0.005**	**0.948±0.005**	0.920±0.015	0.767±0.019	**0.929±0.001**
CIFAR-10	0.457±0.040	0.502±0.004	0.677±0.009	0.890±0.009	0.912±0.011	**0.947±0.001**	**0.949±0.003**	0.876±0.004	**0.917±0.002**

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE VI

FEATURE QUALITY BEFORE AND AFTER JOINT TRAINING. THE FORMAT OF RESULTS FOR BEFORE JOINT TRAINING MEANS THE ACC OF *supervised / unsupervised / reliable labels*, FOR AFTER JOINT TRAINING MEANS THE ACC OF *supervised / unsupervised*. THE SUPERVISED RESULTS WERE OBTAINED BY TRAINING A LINEAR CLASSIFIER WHILE FIXING THE FEATURE MODEL.

Method	STL-10	CIFAR-10	CIFAR-100-20
Before	0.908/0.862/0.977	0.893/0.845/0.965	0.729/0.468/0.677
After (SPICE)	**0.938/0.920**	**0.922/0.918**	**0.723/0.535**
After (SCAN)	0.877/0.855	0.901/0.874	0.693/0.455

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE VII

ABLATION STUDIES OF $SPICE_S$ ON THE WHOLE STL10 DATASET.

Variants	acc	NMI	ARI
Non-overlap	0.885±0.002	0.788±0.003	0.771±0.003
Joint-SH	0.622±0.061	0.513±0.037	0.437±0.053
Joint-MH	0.687±0.037	0.577±0.029	0.512±0.033
Entropy	0.907±0.001	0.817±0.003	0.810±0.003
CE	0.875±0.031	0.784±0.017	0.764±0.033
TCE	0.895±0.005	0.794±0.010	0.787±0.010
$SPICE_S$	**0.908±0.001**	**0.817±0.002**	**0.812±0.002**

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE VIII

RESULTS OF $SPICE_S$ WITH DIFFERENT DATA AUGMENTATION STRATEGIES ON THE WHOLE STL10 DATASET, WHERE RESNET34 WAS USED AS BACKBONE, AND EACH VARIANT WAS CONDUCTED FIVE TIMES.

Aug1	Aug2	ACC	NMI	ARI
Weak	Weak	0.905±0.002	0.815±0.003	0.808±0.003
Strong	Weak	0.883±0.029	0.799±0.019	0.781±0.031
Strong	Strong	0.902±0.008	0.812±0.009	0.803±0.013
Weak	Strong	**0.908±0.001**	**0.817±0.002**	**0.812±0.002**

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE IX

CLUSTERING PERFORMANCE ON IMBALANCED DATASETS. ζ REPRESENTS THE IMBALANCE LEVEL, THE SMALLER ζ THE MORE IMBALANCED.

Method	Metric	ζ=1.0	ζ=0.8	ζ=0.6	ζ=0.4	ζ=0.2
$SCAN_{MoCo}$	ACC	0.882	0.887	0.864	0.758	0.702
	NMI	0.800	0.807	0.775	0.716	0.677
	ARI	0.772	0.707	0.738	0.601	0.542
SPICE	ACC	**0.923**	**0.922**	**0.920**	**0.905**	**0.850**
	NMI	**0.863**	**0.859**	**0.847**	**0.832**	**0.639**
	ARI	**0.847**	**0.842**	**0.835**	**0.810**	**0.778**