RPSC: Robust Pseudo-Labeling for Semantic Clustering

Sihang Liu[1], **Wenming Cao**[2], **Ruigang Fu**[3], **Kaixiang Yang**[1*], **Zhiwen Yu**[1,4]
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[2]School of Mathematics and Statistics, Chongqing Jiaotong University, Chongqing, China
[3]College of Electronic Science and Technology, National University of Defense Technology, Changsha, China
[4]Peng Cheng Laboratory, Shenzhen, China
202221044888@mail.scut.edu.cn, wenmincao2-c@my.cityu.edu.hk, furuigang08@nudt.edu.cn,
yangkx@scut.edu.cn, zhwyu@scut.edu.cn

Abstract

Clustering methods achieve performance improvement by jointly learning representation and cluster assignment. However, they do not consider the confidence of pseudo-labels which are not optimal as supervised information, resulting into error accumulation. To address this issue, we propose a Robust Pseudo-labeling for Semantic Clustering (RPSC) approach, which includes two stages. In the first stage (RPSC-Self), we design a semantic pseudo-labeling scheme by using the consistency of samples, *i.e.*, samples with same semantics should be close to each other in the embedding space. To exploit robust semantic pseudo-labels for self-supervised learning, we propose a soft contrastive loss (SCL) which encourage the model to believe high-confidence sematic pseudo-labels and be less driven by low-confidence pseudo-labels. In the second stage (RPSC-Semi), we first determine the semantic pseudo-label of a sample based on the distance between itself and cluster centers, followed by screening out reliable semantic pseudo-label by exploiting the consistency. These reliable pseudo-labels are used as supervised information in the pseudo-semi-supervised learning algorithm to further improve the performance. Experimental results show that RPSC outperforms 18 competitive clustering algorithms significantly on six challenging image benchmarks. In particular, RPSC achieves an accuracy of 0.688 on ImageNet-Dogs, which is an up to 24% improvement, compared with the second-best method. We conduct ablation studies to investigate effects of different augmented strategies on RPSC as well as contributions of terms in SCL to clustering performance. Experimental results indicate that SCL can be easily integrated into existing clustering methods and bring performance improvement.

Introduction

Clustering algorithms divide unlabeled data into groups via using the similarity, so that data in the same group are more similar to those from different groups. Traditional clustering algorithms like hierarchical clustering (Johnson 1967), K-means (Hartigan and Wong 1979), DBSCAN (Schubert et al. 2017) require high-quality features to obtain desirable performance. However, when dealing with high-dimensional and complex data, traditional clustering algorithms may lead to poor results due to the difficulty in extracting high-quality features. With the improvement of representation learning capability of neural networks, deep learning has come to the fore in clustering tasks, referred to as deep clustering.

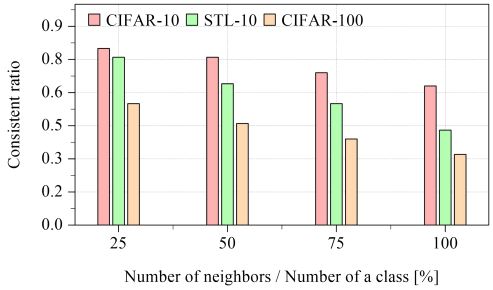


Figure 1: The ratio of neighboring samples that are semantically identical for CIFAR-10, STL-10 and CIFAR-100.

Existing deep clustering methods are divided into single-stage methods (Yang et al. 2017; Xie, Girshick, and Farhadi 2016; Yang, Parikh, and Batra 2016; Guo et al. 2017; Li et al. 2021) and two-stage methods (Van Gansbeke et al. 2020; Dang et al. 2021; Niu, Shan, and Wang 2022). The former ones learn representations and cluster assignment simultaneously. For example, DEC (Xie, Girshick, and Farhadi 2016) learns the mapping from data space to embedding space through a pre-trained network, and uses K-means and Kullback-Leibler (KL) divergence loss in the embedding space to optimize the clustering objective. JULE (Yang, Parikh, and Batra 2016) obtains the label sequence by merging clusters in the forward propagation, and uses the label sequence as supervisory information to learn representations. CC (Li et al. 2021) proposes a comparative learning framework combining instance level and cluster level.

Unlike one-stage methods, two-stage methods (Van Gansbeke et al. 2020; Niu, Shan, and Wang 2022) first use pretext tasks for representation learning and then perform cluster assignment. For example, SCAN (Van Gansbeke et al. 2020) exploits contrastive learning to mine nearest neighbors in the first stage and force the model to output the same label for similar samples in the second stage. SPICE (Niu, Shan, and Wang 2022) proposed a deep clustering frame-

*Corresponding author.

work for images based on semantic pseudo-labels through contrastive learning, prototype clustering, and pseudo-label semi-supervised training. Although two-stage clustering algorithms have achieved performance gains, they lack mechanisms for screening robust semantic pseudo-labels. Specifically, in the first stage, SCAN assigns semantic pseudo-labels by predicting beyond the threshold simply, and SPICE assigns semantic pseudo-labels via the clustering of samples and prototypes. These methods encourage the model to output the same label for samples with similar embeddings. However, this is not the case when the samples are located near the boundaries of different clusters, as shown in Fig. 1. The horizontal axis of Fig. 1 denotes the proportion of the number of samples selected to be neighbors of a prototype for a cluster in data sets. The vertical axis denotes the proportion of the selected samples that share the same semantic with this prototype. The best-case results are selected in Fig. 1, from which we find that even in the best case, when the number of considered neighbors is close to the number of a class, the nearest neighbor samples often do not belong to the same semantic class.

In this work, we propose a new semantic pseudo-labeling method that jointly considers the confidence of semantic pseudo-labels and the consistency of similar samples in the embedding space. Specifically, the semantic pseudo-labeling method first determines the cluster centers in the pre-trained embedding space. Based on the consistency of similar samples in the embedding space, a corresponding pseudo-label is attached to samples around the cluster center, and the confidence of the pseudo-label is determined based on the distance between the sample and the cluster center. Finally, we propose soft contrastive loss (SCL) to rationally utilize semantic pseudo-labels to supervise model training. The numerator of SCL encourages samples to approach pseudo-labels in according with confidence, and the denominator of SCL forces samples with different pseudo-labels to have different predictions. See the Method section for details. In summary, the main contributions of our work are as follows:

- We propose a novel RPSC approach to extract semantic pseudo-labels for image clustering, which considers the confidence of semantic pseudo-labels and the consistency of similar samples in the embedding space.
- We propose a soft contrastive loss that encourages the model to make high-confident predictions for robust semantic pseudo-labels while preventing from learning wrong predictions, which can be easily integrated into existing clustering methods for improving performance.
- The proposed method shows superior performance on six challenging image data sets. In particular, it achieves up to 24% improvement in terms of accuracy on ImageNet-Dogs, compared to the most competitive baseline.

Related Work

Contrastive Learning

Contrastive learning extracts semantic features which improve performances of downstream tasks. Contrastive learning methods include SimCLR (Chen et al. 2020), MOCO (He et al. 2020) and BYOL (Grill et al. 2020). As a paradigm for unsupervised learning, contrastive learning has achieved state-of-the-art performance in representation learning (Li et al. 2021, 2022). Contrastive learning first defines positive and negative sample pairs, and maximizes the similarity of positive samples while minimizing the similarity of negative samples. SimCLR (Chen et al. 2020) first conducts two augmentations on data in a mini-batch, and considers results of the same image and augmentation after contrastive prediction task as positive samples. The results of the other samples in mini-batch after two kinds of augmentations and contrastive prediction task are considered as negative samples. Performance of contrastive learning can be improved by data enhancement, such as predicting patch context (Doersch, Gupta, and Efros 2015; Mundhenk, Ho, and Chen 2018), coloring images (Zhang, Isola, and Efros 2016; Larsson, Maire, and Shakhnarovich 2017), using adversarial training (Donahue, Krähenbühl, and Darrell 2016; Donahue and Simonyan 2019), predicting noise (Bojanowski and Joulin 2017), predicting rotations (Gidaris, Singh, and Komodakis 2018), performing instance differentiation (Wu et al. 2018; Tian, Krishnan, and Isola 2020; Misra and Maaten 2020).

Deep Clustering

Traditional clustering algorithms, such as K-means (Hartigan and Wong 1979), Gaussian Mixture Model (Reynolds et al. 2009), DBSCAN (Schubert et al. 2017) and Hierarchical Clustering (Johnson 1967), rely on handcrafted features heavily and perform poorly on high-dimensional data. Unlikely, deep clustering uses the powerful representation learning capability of deep networks for clustering high-dimensional data. Initially, several methods that combine deep networks with traditional clustering algorithms like K-means and Spectral Clustering (Ng, Jordan, and Weiss 2001) have been proposed. For example, Deep embedded clustering (DEC) (Xie, Girshick, and Farhadi 2016) uses a self-encoder to project samples into a low-dimensional space, where K-means is adopted to obtain cluster assignments. JULE (Yang, Parikh, and Batra 2016) utilizes CNN to extract supervised information from high-confidence images, and implements clustering assignment via K-means to iteratively improve clustering. SpectralNet (Shaham et al. 2018) learns a mapping that embeds data points into the eigenspace of their associated graph Laplacian matrix and subsequently clusters them. For training the network, SpectralNet incorporates a constrained stochastic optimization, which can scale to large datasets. However, these methods are susceptible to random initialization of networks, resulting in extracting only low-level features. These low-level features have a direct impact on performance and may lead to errors accumulated during iteration. Recently, many studies adopt two-stage clustering strategies (Van Gansbeke et al. 2020; Dang et al. 2021; Niu, Shan, and Wang 2022), where the first stage (i.e., self-labeling) uses pseudo-labels generated by initial clustering of representation learning, and the second stage utilizes labeled data to further improve the clustering performance. All these approaches assumes that near-neighboring instances have the same semantics in the embedded space discovered by representation learning. However, features in

the embedding space are not perfect and similar instances do not always have the same semantics, especially when samples are located near the boundaries of different clusters. This will degrade the clustering performance.

Method

In this section, we introduce RPSC whose framework is in Fig. 2. We optimize an unsupervised representation learning model and freeze the pre-trained CNN backbone to extract features for the following two stages: RPSC-Self and RPSC-Semi. RPSC-Self stage, which has two branches, learns projection heads in an unsupervised setting. The former branch takes original images as input to perform the self-labeling using the CNN backbone and projection head, and the latter branch takes the strongly transformed image as input to predict the cluster labels and supervise it with pseudo-labels generated by previous self-labeling process. RPSC-self utilizes only the second branch to compute the loss function to train the projection head. With results from self-labeling process, RPSC-Semi first determines reliable pseudo-labels based on local consistency, followed by performing semi-supervised learning using pseudo-labels and unlabeled data, and finally predicts the clustering labels of all the images using the trained projection head and CNN backbone of the frozen pre-trained model. The right most part in this framework illustrates a toy example. First, three cluster centers are determined, which are three asterisk marks. Pseudo-labels and confidences are assigned to samples near the cluster centers. Ellipses in different colors denote different pseudo-labels. For the table in Fig. 2, the first row denotes the indices of samples, the second row pseudo-labels for RPSC-self stage, and the third row the confidences of pseudo-labels.

RPSC-Self Stage

Before RPSC-Self stage, we have pre-trained an unsupervised representation learning model and frozen its backbone. Given the backbone parameters θ_B and image data set $\mathcal{X} = \{x_i\}_{i=1}^N$, the goal of RPSC-Self is to obtain robust pseudo-labeling for images by learning the projection head. Unlike previous pseudo-labeling schemes, we obtain robust semantic pseudo-labeling based on the degree of semantic certainty in the RPSC-Self stage. This can solve the problem that previous methods obtain ambiguous semantic pseudo-labeling. After generating robust semantic pseudo-labels, we design a Soft Contrastive Loss (SCL), which supervises the projection head with robust semantic pseudo-labels.

In RPSC-Self stage, each training process consists of self-labeling and supervised training. Specifically, in the self-labeling phase we utilize the trained CNN backbone to compute features of a batch original samples, i.e., $F = \Phi_B\left(\mathcal{X}_b; [\theta_B]\right)_{M\times D}$, where $\mathcal{X}_b$ denotes a batch original samples, θ_B and Φ_B are parameters and mapping function of the trained CNN backbone, *M* denotes the batch size in the training phase of RPSC-Self and *D* denotes the embedding space dimension. We utilize the projection head $ph(\cdot)$ to compute the semantic prediction probability for instances in each batch, i.e., $Prob = \Phi_{ph}\left(F; [\theta_{ph}]\right)_{M\times C}$, where θ_{ph} and Φ_{ph} are parameters and mapping function of the projection head, and *C* denotes the number of cluster centers. Here we adopt the projection head of BYOL (Grill et al. 2020).

In each batch, we select top-γ confident prediction from each cluster as pseudo-labels. A prediction prob will be selected as a pseudo-label if it meets the following condition:

$$\begin{aligned} n &= \gamma \times M/C, \\ \mathrm{Prob}_k^* &= sort(\{\mathrm{Prob}[:,k] \mid k \in [1,C]\})[n], \\ \mathrm{id}_c &= \{i|\mathrm{Prob}[i,c] \geq \mathrm{Prob}_c^*, i \in [1,\cdots,M]\}, \end{aligned} \tag{1}$$

where γ is the confidence ratio which is fixed at 0.5, M/C denotes the balanced allocation of M samples to C clusters, Prob_k^* is the *n*-th maximum confidence on cluster $k \in [1, C]$, and id_c denotes the index of the highest confidence samples that are partitioned into the c-th cluster.

With the prediction results of the projection head, one can select the top highest-confidence samples for each cluster and obtain the cluster centers. We assign corresponding semantic pseudo-labels to nearest neighboring samples for each cluster center c, denoted by $\mathcal{N}_c$, and assign semantic confidence to samples based on their distance from the cluster center, which can be formally described as

$$\Gamma_c = \frac{1}{|\mathrm{id}_c|} \sum_{i \in id_c} F[i,:], \tag{2}$$

$$\mathrm{d}_{ic} = \frac{E(F[i,;],\Gamma_c)}{max(E(F[j,:],\Gamma_c)), x_j \in \mathcal{N}_c}, \tag{3}$$

where Γ_c is based on the average in the embedding space of samples indexed with high confidence in the c-th cluster (*i.e.,* id_c). $E(\cdot,\cdot)$ denotes Euclidean distance. The denominator in Eq. (3) indicates the furthest distance from Γ_c among samples assigned semantic pseudo-label c. The numerator in Eq. (3) indicates the distance from the *i*-th sample to Γ_c. d_{ic} reflects the confidence of the *i*-th sample assignment semantic pseudo-label c.

In the supervised training phase, we use semantic pseudo-labeling and semantic confidence to update parameters of the projection head. Specifically, we first perform a strong data augmentation transformation on images in each batch of size *N*, which have semantic pseudo-label. The semantic prediction confidence is then computed using the CNN backbone and projection head. Finally, the Soft Contrastive Loss (SCL) is computed using semantic prediction confidence and semantic pseudo-label of image. The basic idea behind this loss function is to encourage the model to make credible predictions based on both semantic confidence and semantic pseudo-label. For samples with high semantic confidence, the model is encouraged to believe in semantic pseudo-labels, while samples with low semantic confidence are less driven. For each augmented sample x_i with pseudo-label pred_c, Soft Contrastive Loss (SCL) is defined as

$$L_{scl}(i) = -\log \frac{\exp\left(s\left(z_i, pred_c * d_{ic}\right)/\tau\right)}{\sum_{k=1}^{N} \mathbb{I}_{[pred_k \neq pred_i]} \exp(s(z_i, z_k)/\tau)}, \tag{4}$$

where z_i is the feature of x_i extracted by the backbone and projection head, $s(\cdot,\cdot)$ denotes the cosine similarity between two vectors, $pred_c \in \mathbb{R}^C$ denotes a one-hot vector whose

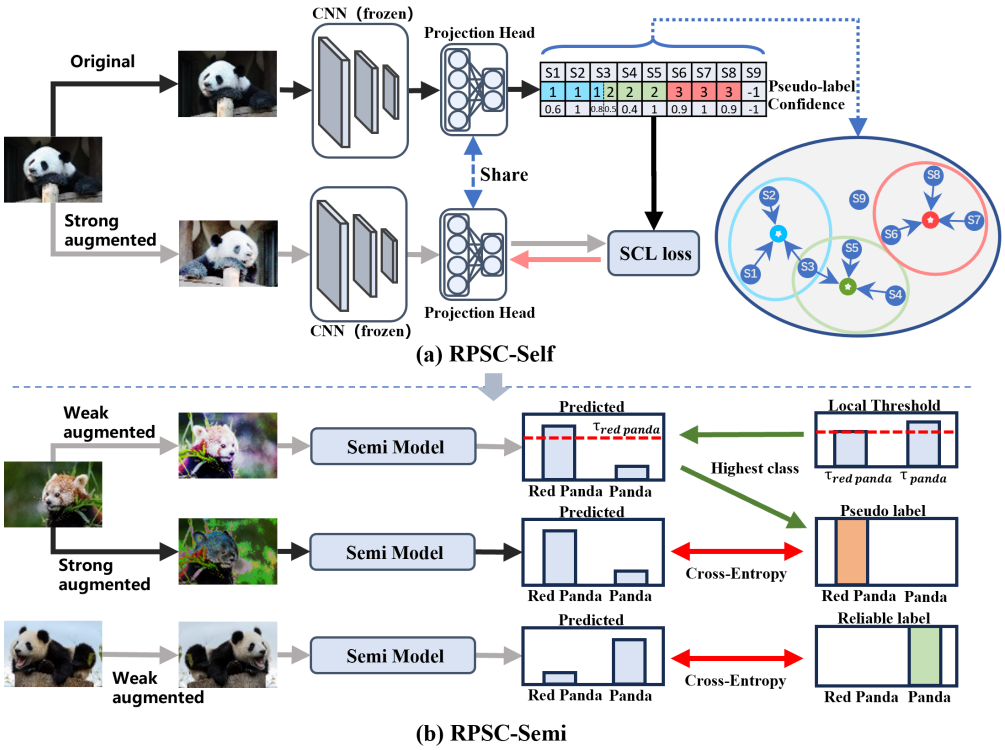


Figure 2: The detailed framework of the proposed RPSC, which is composed of two stages: RPSC-Self and RPSC-Semi.

c-th dimension is 1, $\mathbb{I}$ is the indicator, and τ is the temperature parameter to control the softness. We traverses all augmented V samples to compute the Soft Contrastive Loss by minimizing the following objective:

$$\mathcal{L}_{SCL} = \frac{1}{V}\sum_{i=1}^{V} L_{scl}(i). \tag{5}$$

In contrastive learning, pairs of different augmented samples were considered as negative samples because of missing labeled information. However, when we have pseudo-labels, intra-class samples should not be pushed apart. Therefore, we use semantic pseudo-labels to remove intra-class samples (i.e., the denominator in Eq. (4)). We consider that the predicted probability of the cluster center should approximate the one-hot form of that cluster. Thus in the numerator of Eq. (4), we expect the predictions to approximate the corresponding semantic pseudo-labels. The purpose of semantic confidence is to find robust semantic pseudo-labels while preventing the model from learning wrong semantic pseudo-labels simultaneously.

In the RPSC-Self phase, we optimize the projection head merely. Therefore, the computational burden is significantly reduced. We train independent multiple heads simultaneously to mitigate the instability of the initialized clustering, and choose the best projection head with the minimum loss value of $\mathcal{L}_{SCL}$ over the whole data set. During testing, input images are categorized into different clusters using the trained model with the chosen best projection head.

RPSC-Semi Stage

The goal of RPSC-Semi is to filter out robust semantic pseudo-labels as supervised information and further improve the clustering performance using a semi-supervised learning paradigm. Specifically, we compute cluster centers Γ for all the images using Eq. (1) and Eq. (2), and attach the semantic label of the nearest cluster in the embedded space to each image. For each sample, we select its nearest N_e samples in embedded space based on cosine similarity. Semantic labels of these samples are denoted by L_{N_e}. We use local consistency to filter out reliable semantic pseudo-labels as below:

$$\begin{aligned} l_i &= \arg\max_j (s(F[i,:], \Gamma_j)), j \in [1, \cdots, C], \\ \alpha_i &= \frac{1}{N_e}\sum_{l_j \in L_{N_e}} \mathbb{I}(l_j = l_i). \end{aligned} \tag{6}$$

The semantic pseudo-labeling of a sample is considered reliable if α_i is greater than a predefined threshold τ_t. After obtaining some images with reliable semantic pseudo-labels, we use a semi-supervised learning framework to retrain the model, as shown in Fig. 2. The core of RPSC-Semi is Self-Adaptive Thresholding which automatically define and adjusts confidence thresholds for each category by using predictions during iterations. The loss of RPSC-Semi stage is composed of supervised loss and unsupervised loss. Supervised loss of data with semantic pseudo-labels is defined by:

$$\mathcal{L}_s = \frac{1}{B}\sum_{b=1}^{B} \mathcal{H}(l_b, \Phi_\eta(\omega(x_b))), \tag{7}$$

where B denotes the batch size of the labeled image in the RPSC-Semi stage, $\mathcal{H}(\cdot,\cdot)$ cross-entropy loss, $\omega(\cdot)$ weak augmentation (i.e., random crop and flip), and Φ_η is parameterized by a neural network with weights η.

Unsupervised loss of data without pseudo-labels is given by:

$$\mathcal{L}_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B}\mathbb{I}(\max(q_b) > \tau_t(q_b)) \cdot \mathcal{H}(\widehat{q}_b, \Phi_\eta(\Omega(x_b))), \quad (8)$$

where μB is the batch size of unlabeled image in the RPSC-Semi stage and Ω strong augmentation, $q_b = \Phi_\eta(\omega(u_b))$ the prediction of a weakly-transformed image, $\hat{q}_b$ is the hard "one-hot" label converted from q_b, and $\tau_t(q_b)$ determines a threshold from FreeMatch (Wang et al. 2022).

The overall objective for RPSC-semi Stage is defined by:

$$\mathcal{L} = \mathcal{L}_s + L_u, \quad (9)$$

where the first term encourages the model to learn the clustering semantics based on results of RPSC-Self, and the second term encourages the model to make consistency predictions for samples under different data augmentations.

Experiments

In this section, we investigate effectiveness of the proposed RPSC by conducting comparative experiments on six public image data sets: STL-10, CIFAR-10, CIFAR-100-20, ImageNet-10, ImageNet-Dog, and Tiny-ImageNet. Table 1 summarizes the information about the above six data sets.

Dataset	Classes	Training	Testing	Image Size
STL10	10	5000	8000	96×96
Cifar10	10	50000	10000	32×32
Cifar100-20	20	50000	10000	32×32
ImageNet-10	10	13000	N/A	224×224
ImageNet-Dog	15	19500	N/A	224×224
Tiny-ImageNet	200	100000	10000	200

Table 1: The Description about Six Image Data Sets.

Implementation Details and Evaluation Metrics

Frame Settings For a fair comparison, we use the same backbone as SPICE (Niu, Shan, and Wang 2022) for RPSC-Self. For all the data sets, our backbone is first pre-trained using Moco-v2 (He et al. 2020). The projection head of RPSC-Self is the same as that in BYOL (Grill et al. 2020). Specifically, the projection head is sequentially composed of a linear layer with batch normalization, a ReLU layer, and a linear layer. For strong augmentation in the RPSC-Self stage, we employ the same strategy in SCAN (Van Gansbeke et al. 2020). The images are strongly augmented by combining Cutout (DeVries and Taylor 2017) and four randomly selected transformations from RandAugment (Cubuk et al. 2020). In the RPSC-Semi stage, we adopt the same framework and data augmentation strategy as FreeMatch (Wang et al. 2022).

Parametric Setting We set M to 1,000 for STL-10, CIFAR-10, and ImageNet10 that contain 10 clusters, 1,500 for ImageNet-Dog with 15 clusters, 2,000 for CIFAR-100-20 with 20 clusters, and 5,000 for Tiny-ImageNet with 200 clusters, which we used the same settings as SPICE (Niu, Shan, and Wang 2022). We set the number of projection heads to 10 and set the confidence ratio γ to 0.5 based on experience and temperature parameters $\tau = 0.5$. $N_e = 100$ and $\tau_t = 0.95$ will be set when selecting reliable semantic pseudo-labels in the RPSC-Semi. Other parameters settings in RPSC-Semi are the same as FreeMatch.

Evaluation Metrics The performance of clustering is evaluated by normalized mutual information (NMI) (McDaid, Greene, and Hurley 2011), accuracy (ACC), and adjusted rand index (ARI) (Hubert and Arabie 1985). Higher values of these metrics indicate better performances.

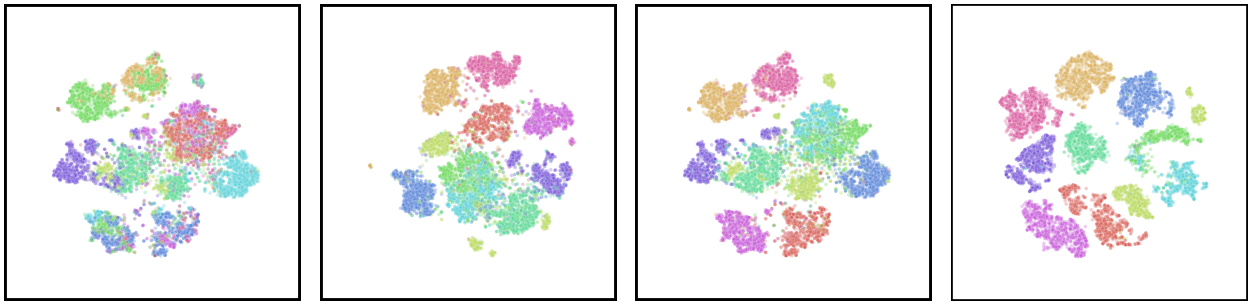
Comparisons with State of the Arts

We compare RPSC with competing methods which include K-means (Hartigan and Wong 1979) , SC (Zelnik-Manor and Perona 2004), AE (Bengio et al. 2006), DAE (Vincent et al. 2010), DCGAN (Radford, Metz, and Chintala 2015), DeCNN (Zeiler et al. 2010), VAE (Kingma and Welling 2013), JULE (Yang, Parikh, and Batra 2016), DEC (Xie, Girshick, and Farhadi 2016), DAC (Chang et al. 2017), DDC (Chang et al. 2019), IIC (Ji, Henriques, and Vedaldi 2019), PICA (Huang, Gong, and Zhu 2020), CC (Li et al. 2021), SCAN(Van Gansbeke et al. 2020), DeepCluE (Huang et al. 2022), SPICE (Niu, Shan, and Wang 2022) and DivClust (Metaxas, Tzimiropoulos, and Patras 2023). Table 2 illustrates the comparison results on six image data sets.

Since SPICE is also a two-stage algorithm, we compare it with our proposed RPSC in stages. In Table 2, the performance of RPSC-Self outperforms these state-of-the-art baselines on all the data sets except CIFAR-10. Especially on ImageNet-Dogs, NMI, ACC, and ARI of RPSC exceed the most advanced baselines by 5.4%, 9.4%, and 10.3% respectively. Moreover, RPSC-Self has improved, compared to the most advanced baselines on ImageNet-10, CIFAR-100, Tiny-ImageNet, and STL-10. In addition, RPSC-Self failed to surpass SCAN on CIFAR-10. The reason may be that RPSC-Self is different from the backbone used by SCAN on CIFAR-10, resulting in poor pre-training of features. After semi-supervised training, RPSC-Semi increased NMI, ACC and ARI by 1.0%, 1.4% and 2.4% on CIFAR-10, 0.4%, 1.8% and 1.7% on CIFAR-100, and 13.7%, 13.4% and 18.7% on ImageNet-Dogs. Although results of RPSC-Semi are significantly better than existing results, there will be problems with clustering on Tiny-ImageNet. This is because RPSC-Self has low performance on Tiny-ImageNet, resulting in the inability to generate reliable semantic pseudo-labels, which makes RPSC-Semi unable to perform semi-supervised training. Here is one of the problems we need to solve in the future. In a word, the above results indicate the effectiveness and robustness of our RPSC. This benefits from two factors: 1) RPSC considers the confidence of semantic pseudo-labels and the consistency of similar samples in the embedded space to improve precise network guidance;

Dataset	CIFAR-10			CIFAR-100			STL-10			ImageNet-10			ImageNet-Dogs			Tiny-ImageNet		
Metrics	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
K-means	0.087	0.229	0.049	0.084	0.130	0.028	0.125	0.192	0.061	0.119	0.241	0.057	0.055	0.105	0.020	0.065	0.025	0.005
SC	0.103	0.247	0.085	0.090	0.136	0.022	0.098	0.159	0.048	0.151	0.274	0.076	0.038	0.111	0.013	0.063	0.022	0.004
AE	0.239	0.314	0.169	0.100	0.165	0.048	0.250	0.303	0.161	0.210	0.317	0.152	0.104	0.185	0.073	0.131	0.041	0.007
DAE	0.251	0.297	0.163	0.111	0.151	0.046	0.224	0.302	0.152	0.206	0.304	0.138	0.104	0.190	0.078	0.127	0.039	0.007
DCGAN	0.265	0.315	0.176	0.120	0.151	0.045	0.210	0.298	0.139	0.225	0.346	0.157	0.121	0.174	0.078	0.135	0.041	0.007
DeCNN	0.240	0.282	0.174	0.092	0.133	0.038	0.227	0.299	0.162	0.186	0.313	0.142	0.098	0.175	0.073	0.111	0.035	0.006
VAE	0.245	0.291	0.167	0.108	0.152	0.040	0.200	0.282	0.146	0.193	0.334	0.168	0.107	0.179	0.079	0.113	0.036	0.006
JULE	0.192	0.272	0.138	0.103	0.137	0.033	0.182	0.277	0.164	0.175	0.300	0.138	0.054	0.138	0.028	0.102	0.033	0.006
DEC	0.257	0.301	0.161	0.136	0.185	0.050	0.276	0.359	0.186	0.282	0.381	0.203	0.122	0.195	0.079	0.115	0.037	0.007
DAC	0.396	0.522	0.306	0.185	0.238	0.088	0.366	0.470	0.257	0.394	0.527	0.302	0.219	0.275	0.111	0.190	0.066	0.017
DDC	0.424	0.524	0.329	N/A	N/A	N/A	0.371	0.489	0.267	0.433	0.577	0.345	N/A	N/A	N/A	N/A	N/A	N/A
IIC	N/A	0.617	N/A	N/A	0.257	N/A	N/A	0.257	N/A	N/A	0.610	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PICA	0.591	0.696	0.512	0.310	0.337	0.171	0.611	0.713	0.531	0.802	0.870	0.761	0.352	0.352	0.201	0.277	0.098	0.040
CC	0.705	0.790	0.637	0.431	0.429	0.266	0.764	0.850	0.726	0.859	0.893	0.822	0.445	0.429	0.274	0.340	0.140	0.071
SCAN	**0.787**	**0.876**	**0.758**	0.468	0.459	0.301	0.680	0.767	0.616	0.859	0.893	0.822	N/A	N/A	N/A	N/A	N/A	N/A
DeepCluE	0.727	0.764	0.646	0.472	0.457	0.288	N/A	N/A	N/A	**0.882**	0.924	0.856	0.448	0.416	0.273	0.379	0.194	0.102
DivClust	0.724	0.819	0.681	0.422	0.414	0.260	N/A	N/A	N/A	0.879	0.918	0.851	0.458	0.448	0.296	N/A	N/A	N/A
SPICE-Self	0.734	0.838	0.705	0.448	0.468	0.294	0.817	0.908	0.812	0.840	0.921	0.836	0.498	0.546	0.362	0.449	0.305	0.161
RPSC-Self	0.754	0.857	0.731	**0.476**	**0.518**	**0.341**	**0.838**	**0.920**	**0.834**	0.830	**0.927**	**0.858**	**0.552**	**0.640**	**0.465**	**0.467**	**0.314**	**0.176**
SPICE	0.865	0.926	0.852	0.567	0.538	0.387	0.872	0.938	0.870	0.902	0.959	0.912	0.504	0.554	0.343	N/A	N/A	N/A
RPSC	**0.875**	**0.940**	**0.876**	**0.571**	**0.556**	**0.398**	**0.889**	**0.950**	**0.886**	**0.911**	**0.962**	**0.920**	**0.641**	**0.688**	**0.530**	N/A	N/A	N/A

Table 2: Clustering performance on six datasets. The best results are shown in boldface.



(a) 0 epoch (NMI = 0.212) (b) 25 epoch (NMI = 0.522) (c) 50 epoch (NMI = 0.689) (d) 100 epoch (NMI = 0.744)

Figure 3: The visualization of semantic features learnt by the proposed RPSC concerning epoches using t-SNE on data set CIFAR-10, where different colors indicate different cluster assignments.

and 2) RPSC adopts an adaptive threshold strategy to further improve network performance in the RPSC-Semi stage.

To illustrate the convergence, we provided the visualization of semantic features using t-SNE when the RPSC conducted network optimization on CIFAR-10 at four different epochs. The results are in Fig.3, where different colors indicate that projection head predicts different labels. It indicates that at the beginning, the cluster assignment is chaotic since the projection heads are randomly initialized. As the number of epochs increases, the allocation of clusters becomes more reasonable and the aggregation is more obvious.

Ablation Studies

We performed ablation studies to investigate contributions of data augmentation and contributions of terms in SCL.

1-Aug	2-Aug	NMI	ACC	ARI
No	No	0.752 ± 0.003	0.853 ± 0.002	0.726 ± 0.003
Yes	No	0.731 ± 0.021	0.839 ± 0.017	0.702 ± 0.041
Yes	Yes	0.748 ± 0.005	0.851 ± 0.006	0.721 ± 0.009
No	Yes	**0.754 ± 0.003**	**0.857 ± 0.003**	**0.731 ± 0.004**

Table 3: Effect of data augmentation.

Effect of data augmentation To verify the significance of data augmentation, we analyze the effect of different data enhancement strategies on RPSC-Self. In Table 3, 1-Aug and 2-Aug denote whether there is data augmentation in the first and second branches in Fig. 2(a). From this table, we observe that the model achieves the best performance when the first branch disabled data augmentation while the sec-

ond branch enabled data augmentation. Furthermore, when the first branch enabled data augmentation, the performance of the model is relatively poor because the goal of the first branch is to generate reliable semantic pseudo-labels and data augmentation will produce negative effects on the generation of pseudo-labels. When neither branch enabled data augmentation, the model performs poorly, which indicates that performance of RPSC is related to data augmentation.

Effect of Terms in Soft Contrastive Loss To prove the effectiveness of each module of SCL, we run RPSC-Self with four variants of SCL on CIFAR-10. Specifically, the four SCL variants are: only use the molecular of SCL and do not consider confidence of pseudo-label d_{ic} (SCL-1), use only the molecular of SCL (SCL-2), use SCL without considering pseudo-label confidence (SCL-3), complete SCL. As shown in Table 4, the validity of the denominator of SCL is verified by comparing the performance of SCL over SCL-2, i.e., the denominator takes into account the exclusion of different pseudo-labels from the prediction. By comparing SCL-1 and SCL-2 or SCL-3 and SCL, it can be found that SCL-2 performs better than SCL-1, and SCL performs better than SCL-3, which verifies the validity of SCL in the numerator.

Loss	NMI	ACC	ARI
SCL-1	0.711	0.832	0.683
SCL-2	0.732	0.842	0.706
SCL-3	0.734	0.844	0.702
SCL	**0.754**	**0.857**	**0.731**

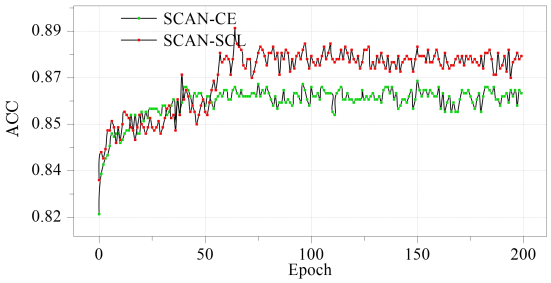
Table 4: Effect of SCL and its variants on CIFAR-10.

Method	NMI	ACC	ARI
SCAN-CE	**0.468**	0.459	0.301
SCAN-SCL	0.457	**0.472**	**0.313**
SPICE-CE	0.422	0.468	0.294
SPICE-SCL	**0.464**	**0.501**	**0.334**
RPSC-CE	0.445	0.486	0.308
RPSC-SCL	**0.476**	**0.518**	**0.341**

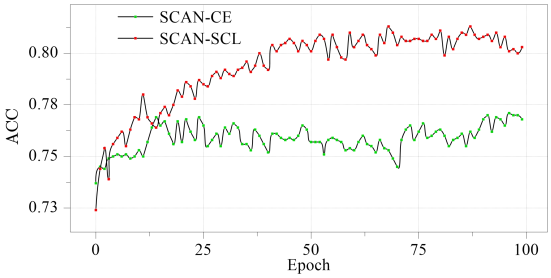
Table 5: RPSC, SPICE and SCAN with CE and SCL on CIFAR-100.

The Integration of Soft Contrastive Loss

Previous two-stage deep clustering methods perform a self-labeling phase in the first stage, and use pseudo labels and cross-entropy loss (CE) in the second stage. Unlike them, we design a soft contrastive loss (SCL) for utilizing pseudo-label. SCL can be easily integrated into previous two-stage deep clustering methods for delivering performance improvement. The reason behind the improvement owes to that when approaching a pseudo-label using SCL, the confidence of pseudo-label is considered, and the prediction of repelling different pseudo-labels is also considered. In Table 5, we compare SCAN, SPICE-Self and RPSC-Self when they adopt CE and SCL on CIFAR-100 respectively. The results show that all three methods using SCL achieve better performance than corresponding methods using CE on CIFAR-100. To intuitively compare CE and SCL, we report the clustering performances of SCAN on CIFAR-10 and STL-10 using CE and SCL in Fig. 4 respectively. From this figure, we observe that the clustering performance improves as the increase of epochs. When the training tends to converge, the performance of SCAN with SCL is better than that with CE on CIFAR-10 and STL-10. This verifies the effectiveness of SCL.



(a) SCAN: STL-10



(b) SCAN: CIFAR-10

Figure 4: Comparison of SCAN using cross-entropy loss and SCL on STL-10 (a) and CIFAR-10 (b) in terms of ACC.

Conclusion

In this paper, we propose a novel semantic clustering approach, referred to as robust Pseudo-labeling for Semantic Clustering (RPSC). RPSC considers the confidence of semantic pseudo-labels and the consistency of similar samples in the embedding space jointly. In the first stage (RPSC-self), self-supervised learning aims to mine robust semantic pseudo-labels with the help of our designed soft contrastive loss. These robust semantic pseudo-labels are high-confidence and used as supervised information for a semi-supervised learning paradigm to further improve clustering performance in the second stage (RPSC-Semi). Extensive experiments demonstrate that RPSC consistently outperforms state-of-the-art baseline methods on six public image data sets. In the future, we shall extend it to other applications such as multi-view clustering and transfer learning.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant 62106224, Grant U21A20478, Grant 62001482, Grant 62306052 and Grant U21B2029, and in part by the Major Key Project of PCL, China under Grant PCL2023AS7-1.

References

Bengio, Y.; Lamblin, P.; Popovici, D.; and Larochelle, H. 2006. Greedy layer-wise training of deep networks. *NeurIPS*, 19.

Bojanowski, P.; and Joulin, A. 2017. Unsupervised learning by predicting noise. In *ICML*, 517–526. PMLR.

Chang, J.; Guo, Y.; Wang, L.; Meng, G.; Xiang, S.; and Pan, C. 2019. Deep discriminative clustering analysis. *arXiv preprint arXiv:1905.01681*.

Chang, J.; Wang, L.; Meng, G.; Xiang, S.; and Pan, C. 2017. Deep adaptive image clustering. In *Proceedings of ICCV*, 5879–5887.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of CVPR workshops*, 702–703.

Dang, Z.; Deng, C.; Yang, X.; Wei, K.; and Huang, H. 2021. Nearest neighbor matching for deep clustering. In *Proceedings of CVPR*, 13693–13702.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of ICCV*, 1422–1430.

Donahue, J.; Krähenbühl, P.; and Darrell, T. 2016. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*.

Donahue, J.; and Simonyan, K. 2019. Large scale adversarial representation learning. *NeurIPS*, 32.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33: 21271–21284.

Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017. Improved deep embedded clustering with local structure preservation. In *IJCAI*, volume 17, 1753–1759.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1): 100–108.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of CVPR*, 9729–9738.

Huang, D.; Chen, D.-H.; Chen, X.; Wang, C.-D.; and Lai, J.-H. 2022. Deepclue: Enhanced image clustering via multilayer ensembles in deep neural networks. *arXiv preprint arXiv:2206.00359*.

Huang, J.; Gong, S.; and Zhu, X. 2020. Deep semantic clustering by partition confidence maximisation. In *Proceedings of CVPR*, 8849–8858.

Hubert, L.; and Arabie, P. 1985. Comparing partitions. *Journal of classification*, 2: 193–218.

Ji, X.; Henriques, J. F.; and Vedaldi, A. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of ICCV*, 9865–9874.

Johnson, S. C. 1967. Hierarchical clustering schemes. *Psychometrika*, 32(3): 241–254.

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

Larsson, G.; Maire, M.; and Shakhnarovich, G. 2017. Colorization as a proxy task for visual understanding. In *Proceedings of CVPR*, 6874–6883.

Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J. T.; and Peng, X. 2021. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 8547–8555.

Li, Y.; Yang, M.; Peng, D.; Li, T.; Huang, J.; and Peng, X. 2022. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130(9): 2205–2221.

McDaid, A. F.; Greene, D.; and Hurley, N. 2011. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*.

Metaxas, I. M.; Tzimiropoulos, G.; and Patras, I. 2023. DivClust: Controlling Diversity in Deep Clustering. In *Proceedings of CVPR*, 3418–3428.

Misra, I.; and Maaten, L. v. d. 2020. Self-supervised learning of pretext-invariant representations. In *Proceedings of CVPR*, 6707–6717.

Mundhenk, T. N.; Ho, D.; and Chen, B. Y. 2018. Improvements to context based self-supervised learning. In *Proceedings of CVPR*, 9339–9348.

Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. *NeurIPS*, 14.

Niu, C.; Shan, H.; and Wang, G. 2022. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31: 7264–7278.

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

Reynolds, D. A.; et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663).

Schubert, E.; Sander, J.; Ester, M.; Kriegel, H. P.; and Xu, X. 2017. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3): 1–21.

Shaham, U.; Stanton, K.; Li, H.; Nadler, B.; Basri, R.; and Kluger, Y. 2018. Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive multiview coding. In *ECCV*, 776–794. Springer.

Van Gansbeke, W.; Vandenhende, S.; Georgoulis, S.; Proesmans, M.; and Van Gool, L. 2020. Scan: Learning to classify images without labels. In *ECCV*, 268–285. Springer.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.-A.; and Bottou, L. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12).

Wang, Y.; Chen, H.; Heng, Q.; Hou, W.; Fan, Y.; Wu, Z.; Wang, J.; Savvides, M.; Shinozaki, T.; Raj, B.; et al. 2022. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of CVPR*, 3733–3742.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*, 478–487. PMLR.

Yang, B.; Fu, X.; Sidiropoulos, N. D.; and Hong, M. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, 3861–3870. PMLR.

Yang, J.; Parikh, D.; and Batra, D. 2016. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of CVPR*, 5147–5156.

Zeiler, M. D.; Krishnan, D.; Taylor, G. W.; and Fergus, R. 2010. Deconvolutional networks. In *Proceedings of CVPR*, 2528–2535. IEEE.

Zelnik-Manor, L.; and Perona, P. 2004. Self-tuning spectral clustering. *NeurIPS*, 17.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *ECCV*, 649–666. Springer.