Progressive Cluster Purification
for Unsupervised Feature Learning

Yifei Zhang[1,2*], Chang Liu[3*], Yu Zhou[1†], Wei Wang[1,2], Weiping Wang[1], and Qixiang Ye[3†]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[3]University of Chinese Academy of Sciences, Beijing, China

{zhangyifei0115, zhouyu, wangweiping}@iie.ac.cn

{liuchang615, wangwei1608}@mails.ucas.ac.cn, qxye@ucas.ac.cn

Abstract—**In unsupervised feature learning, sample specificity based methods ignore the inter-class information, which deteriorates the discriminative capability of representation models. Clustering based methods are error-prone to explore the complete class boundary information due to the inevitable class inconsistent samples in each cluster. In this work, we propose a novel clustering based method, which, by iteratively excluding class inconsistent samples during progressive cluster formation, alleviates the impact of noise samples in a simple-yet-effective manner. Our approach, referred to as Progressive Cluster Purification (PCP), implements progressive clustering by gradually reducing the number of clusters during training, while the sizes of clusters continuously expand consistently with the growth of model representation capability. With a well-designed cluster purification mechanism, it further purifies clusters by filtering noise samples which facilitate the subsequent feature learning by utilizing the refined clusters as pseudo-labels. Experiments on commonly used benchmarks demonstrate that the proposed PCP improves baseline method with significant margins. Our code will be available at https://github.com/zhangyifei0115/PCP.**

I. INTRODUCTION

Representation learning has achieved unprecedented success in various computer vision tasks. This can be attributed to the availability of large-scale datasets with precise label annotations and convolutional neural networks (CNNs) capable of absorbing the annotation information [27], [31], [35]. Nevertheless, precisely annotating samples on large-scale datasets is laborious, expensive, or even impractical.

As an alternative method, unsupervised feature learning (UFL), which is usually based on data completeness, sample distribution and instance similarity, has attracted increased attention. The completeness-based methods typically leverage the intrinsic structure of the input data by forcing models to predict the hidden part in specific well-designed pretext tasks, such as context predicting [5], colorization [18], jigsaw puzzles [20], counting [21], split-brain [30] and rotations prediction [9]. The distribution-based methods could use Auto-Encoders [26], Variational Auto-Encoders [16] and Generative Adversarial Nets [10] to approximate the distribution of raw data. Both methods focus on intra-class distributions but unfortunately ignore inter-class information of samples that the discriminative capacity of extracted features is not fully explored.

As one of the most representative similarity-based UFL method, DeepCluster (DC) [2] iteratively assigns samples to clusters according to their similarity in the feature space and then uses the assignment as pseudo category labels to train a feature extractor. However, DC is puzzled by a significant number of class inconsistent instances (noise samples) within clusters, which could seriously deteriorate the representation learning performance. To avoid the impact of clustering noise, instance recognition (IR) [28] treats each sample as an independent class and achieves impressive performance on commonly used benchmarks. Nevertheless, it totally misses the intrinsic inter-sample class information, which limits its discriminative capability of learned features. To improve discriminative capability, the anchor neighbourhood discovery (AND) [14] method adopts a divide-and-conquer strategy to find local neighbours under the constraint of cluster consistency. However, the sample noise problem remains not systematically solved, which makes the learned features less representative.

In this study, we propose a simple-yet-effective UFL method, termed Progressive Cluster Purification (PCP), to alleviate the impact of sample noise and estimate reliable sample labels. PCP roots in simple k-means clustering while introducing a Progressive Clustering (PC) strategy and a Cluster Purification (CP) module to refine the clustering assignments. With iterative training, reliable class consistent samples converge together so that clear and purified pseudo labels are well estimated while discriminative features are learned in a progressive self-guided manner. As shown in Fig. 1, IR, DC and our PCP take three different strategy to learn discrimimative features in unsupervised manner, where PCP shows its superiority in more stable and efficient training procedure and richer feature representation capability.

PC is implemented by reducing the number of clusters from the number of samples towards the number of categories during the multiple iterations of clustering, so that the size of each cluster progressively increases. In this way, intra-cluster variance increases consistently with the growth of model representation capability so that the class inconsistent samples could be controlled at a reasonable level, which drives the model to improve the discriminability of extracted features.
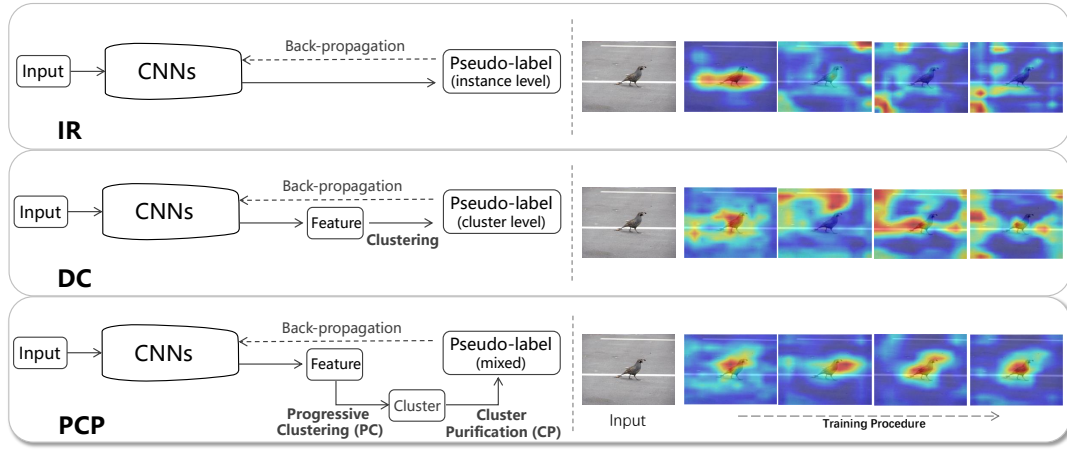
Fig. 1: Comparison of unsupervised feature learning frameworks including IR [28], DC [2] and our PCP (left) and visualization of corresponding features extracted by them (right). The features learned by IR and DC incorporate more background noise while features by our PCP are more stable and consistent with the foreground information.

PC reduces the noise pollutants from the sources. Meanwhile, CP is applied on clustering results in each epoch, with the aim to explicitly obtain stable and reliable clusters. CP consists of two noise processing procedure including unreliable sample filtering (CP_r) and unstable sample filtering (CP_s). With unreliable sample filtering, instances far away from cluster centroids are identified as noise samples, each of which is temporarily regarded as a distinct class. Thereafter, in order to improve the robustness of CP, we further introduce unstable sample filtering, a voting strategy, which is utilized to reassign class consistent instances based on the prevenient filtering results. PCP avoids network focus on low-level feature for clustering by warm-up training strategy. With PCP, the obtained pseudo labels contain purified and clear class discriminative information that facilitates the successive feature learning procedure.

The contributions of this work are summarized as follows:

1) We propose the simple-yet-effective Progressive Cluster Purification (PCP) method, which implements progressive clustering (PC) by iteratively expanding cluster sizes and progressively enforcing feature discrimination power.
2) A novel cluster purification (CP) mechanism is designed to exclude class inconsistent instances, which further aggregates the discrimination power of features and speeds up the convergence of unsupervised learning.
3) With warm-up training strategy and well designed objective function, PCP improves the baseline method with significant margins on commonly used benchmarks.

II. RELATED WORKS

From a general perspective, unsupervised feature learning methods can be classified into three categories: completeness-based, distribution-based and similarity-based methods.

Completeness-based methods. This kind of methods usually follows self-supervision, which designs pretext tasks by hiding some information of examples to construct a prediction task. By training models to predict the hidden information, rich feature representation could be learned. Pretext tasks include context predicting [5], colorization [18], jigsaw puzzles [20], counting [21], split-brain [30] and rotations [9]. After that, FeatureDecoupling [8] proposes combining rotation and its unrelated part, VCP [19] proposes to fill the blank with video options and PRP [33] produces self-supervision signals about video playback rates. Although completeness-based methods can obtain proper feature representation for some specific tasks, it has a low performance bound as the learning procedure focuses on optimizing pretext tasks. In addition, it is unclear whether these pretext tasks are suitable for downstream tasks or not in new domains.

Distribution-based methods. The main idea of distribution-based methods is using encoders, *e.g.*, restricted Boltzmann machines [12] or auto-encoders [26], to reconstruct data distribution while learning feature representation. Deep feature encoders such as deep belief networks [12], deep Boltzmann machines [23], variational auto-encoders [16] and generative adversarial network [10] have been explored for feature learning. Despite the effectiveness, these methods unfortunately ignore inter-class information of samples thus the discriminative power of features is not fully explored.

Similarity-based methods. In the deep learning era, similarity-based feature learning can be implemented by associating clustering with CNNs [2], [4], [32]. As a representative method, DC [2] iteratively groups features and uses the assignments as pseudo labels to train feature representation. DC leverages the advantages of self-supervised learning and clustering to make up their disadvantages. To alleviate the impact of noise samples, Exemplar CNN [6] and IR [28] treat each sample as an independent class. Recent invariant information clustering [15] learns a neural network classifier from scratch with solely unlabelled data samples, and the objective is to maximize mutual information [1], [24] between the class assignments of each pair. AND [14] adopts a divide-and-conquer method to find local neighbours under the constraint of cluster consistency.
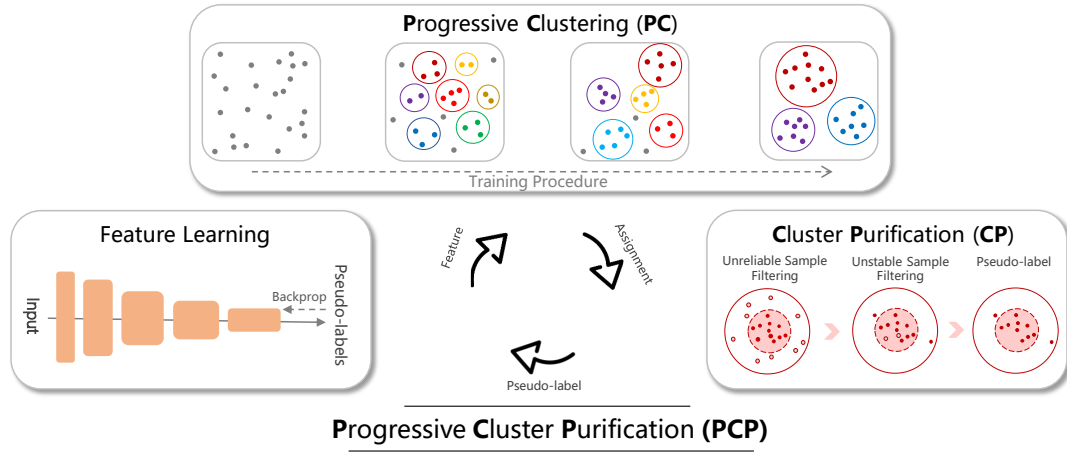
Fig. 2: Overview of our PCP for unsupervised feature learning. Progressive Clustering (PC) is implemented by gradually reducing the number of clusters while continuously expanding the size of clusters during training. Cluster Purification (CP) targets at improving reliability and stability of clustering by identifying unreliable instances and unstable instances. Feature Learning is carried out by using assignments as supervision.

Similarity-based methods, *e.g.* DC, have set strong baselines for UFL. However, the significant performance gap between UFL and supervised feature learning indicates that it is a challenging task when representation models and sample clusters require to be estimated simultaneously. Existing methods remain challenged by the significant number of noise samples during clustering, which deteriorate the performance of representation models.

III. PROGRESSIVE CLUSTER PURIFICATION

In unsupervised feature learning, clustering based method is susceptible to noisy supervision caused by inevitable class inconsistent samples. However, its superiority in reasoning class boundaries, which is so called class conceptualization, should not be neglected. Inspired by this, we propose the Progressive Cluster Purification approach targeting at alleviating the propagation of the noisy supervision during feature learning. As shown in Fig. 2, PCP consists of three components including progressive clustering, cluster purification and feature learning, which cooperate with each other in a circular way. PC assigns samples to distinct clusters in the feature space constructed during feature learning and the assignments are purified by CP to generate subsequent supervision for feature learning.

Given an imagery data set $X = \{x_1, x_2, x_3, \cdots, x_n\}$ without any class label, pseudo-labels produced by PCP are used as supervision to derive CNN with parameters θ. The learned model maps an input image x_i to a feature space $\mathcal{V}(t)$ at the t-th epoch, as $v_i(t) = f_{\theta_t}(x_i)$. Details about the main components of the proposed PCP are illustrated in the following paragraph.

A. Progressive Clustering

IR [28] evidences that neural networks, to some extent, can learn class discriminative feature representation with solely instance-level supervision which regards individual instances as distinct classes. Inspired by this, we would go one more step to infer that deep models can extract the underlying class information under different grain-level supervision from instance-wise to class-wise. More importantly, in early phase of feature learning, the network has not converged that clustering with few centroids would cooperate significant class inconsistent samples within each cluster. With contaminated pseudo labels, inferior class conceptualization will further hinder the model representation capability growth.

To alleviate this situation, we propose progressive clustering to improve unsupervised class conceptualization by shrinking the number of clusters from the number of samples towards the number of true categories, Fig. 2. Consistently with the growth of our model's representation capability, the clusters gradually expand while less noise sample pollutants occur. As the total of samples N is tremendous, we implement a linear declining strategy on its logarithm to decide the number of clusters at the t-th epoch denoted as N_t, which is formulated as

$$lg(N_t) = (1 - \frac{t}{T})lg(N), \tag{1}$$

where T denotes the total of training epochs (200 as default). As shown in Eq.(1), N_t declines fast in early epochs and goes slower in later epochs. When $t = 0$, N_t equals to N. As the true category number is usually unknown prior knowledge, when $t \geq t_0$, we set N_t fixed as N_{t_0} that the cluster number N_t begins with N and stops at N_{t_0} (hyperparameter). Table. III show that our PCP is not as sensitive to the prior knowledge of the cluster number as DC [2] does.

B. Cluster Purification

With PC, the feature space becomes compact and more class-consistent, which facilitates class conceptualization. To further make the cluster more reliable and stable to optimize the feature learning under supervision of clear pseudo-labels with less noise, we design a Cluster Purification (CP) mechanism

8478

which consists of unreliable sample filtering (CP_r) and unstable sample filtering (CP_s).

Unreliable Sample Filtering (CP$_r$). Based on the observation that samples near cluster centroids share higher apparent similarity, thus they are more likely to belong to the same class. We discard the samples far away from the centroids and temporarily regard each one as a distinct class in the subsequent learning procedure. As shown in the lower right of Fig. 2, the pink area within the dashed circle represents the reliable district and hollow points indicate samples to be excluded while solid points to remain. We denote the unreliable sample filtering ratio in each cluster as γ that with higher γ, few samples remain in a cluster. When γ is close to 1, PCP degenerates to IR. That is to say, we build a bridge between sample specificity based methods and clustering based methods through γ that essentially makes our PCP absorb the advantages of clear pseudo supervision with more inter-class information. In the feature space $\mathcal{V}(t)$, we denote the set of reliable class consistent samples of c-th cluster, $\mathcal{S}_c(t)$, as $\mathcal{S}_c^r(t)$ and the corresponding set of noise samples to be discarded as $\mathcal{N}_c^r(t)$. It is obvious that $\mathcal{S}_c(t) = \mathcal{S}_c^r(t) \cup \mathcal{N}_c^r(t)$. Thereafter, these preliminary purification results of CP_r are fed the successive purification procedure.

Unstable Sample Filtering (CP$_s$). Easily distinguished samples are likely to be consistently assigned to the same cluster at different iterations of clustering. However, perplexing samples go the opposite that clustering assignments of which are inconsistent and unstable. Inspired by this, we propose a voting function $V(v_i(t), v_j(t))$ which utilizes the previous clustering results to quantitatively estimate the class consistency of samples $v_i(t)$ and $v_j(t)$ in the same cluster from the feature space $\mathcal{V}(t)$. For cluster $S_c(t)$, we denote the sample closest to the centroid as $v_{i_c}(t)$. Thereafter, the voting score of sample $v_i(t)$ from cluster $S_c(t)$, according to the progressive clustering result in the past n epochs from the t-th epoch, can be calculated as

$$V(v_i(t), v_{i_c}(t)) = \sum_{k=0}^{n} \alpha^k \cdot \delta(C(v_i(t-k)), C(v_{i_c}(t-k))), \quad (2)$$

where $\alpha \in (0,1)$ denotes the decay rate, $C(v_i(t))$ denotes the pseudo label which sample $v_i(t)$ belongs to, and $\delta(x,y) = 1$ when $x = y$, otherwise -1.

As shown in the lower right of Fig. 2, we set different thresholds in $\mathcal{S}^r(t) = \cup_c \mathcal{S}_c^r(t)$ and $\mathcal{N}^r(t) = \cup_c \mathcal{N}_c^r(t)$ to discard samples in $\mathcal{S}^r(t)$ with low voting scores (hollow points) and pull back samples in $\mathcal{N}^r(t)$ with high voting scores (solid points). Then sets $\mathcal{S}_c^r(t)$ and $\mathcal{N}_c^r(t)$ are refined as $\mathcal{S}_c^s(t)$ and $\mathcal{N}_c^s(t)$ respectively. $\mathcal{S}^s(t) = \cup_c \mathcal{S}_c^s(t)$ and $\mathcal{N}^s(t) = \cup_c \mathcal{N}_c^s(t)$ are regarded as mixed instance-level and cluster-level pseudo labels for feature learning.

C. Feature Learning

IR [28] involves a non-parametric softmax classifier aiming to minimize the negative log-likelihood objective function on the training set where each sample is regarded as a distinct

---

Algorithm 1 Progressive Cluster Purification.

Input: An imagery dataset X without labels;
Output: CNN model f_θ with parameters θ ;

1: Preset embedding feature dimension D, training epochs T, cluster number N_{t_0} for stopping declining;
2: **for** epoch $t = 1$ **to** T **do**
3:     Get the number of clusters $N_t = max(N_t, N_{t_0})$ during the process of **PC**, Eq.(1);
4:     Obtain D-dimensional feature space $\mathcal{V}(t)$ by CNN model, $v(t) = f_{\theta_t}(x)$;
5:     Implement k-means clustering algorithm to get $\cup_c \mathcal{S}_c(t)$ with N_t clusters;
6:     **for** each cluster $c = 1$ **to** N_t **do**
7:         Split $\mathcal{S}_c(t)$ into class consistent set $\mathcal{S}_c^r(t)$ and noise set $\mathcal{N}_c^r(t)$ by **CP**$_r$;
8:         Update class consistent set as $\mathcal{S}_c^s(t)$ and noise set as $\mathcal{N}_c^s(t)$ by **CP**$_s$, Eq.(2);
9:         Calculate objective loss L_{pcp}^t (Eq.(6)) according to the union of set, $\cup_c \mathcal{S}_c^s(t)$ and $\cup_c \mathcal{N}_c^s(t)$;
10:     Feature learning by gradient back-propagation and updating model weights;
11: **return** f_θ.

---

class. The probability of an input x being recognized as i-th example is

$$P(i|v) = \frac{exp(v_i^T v/\tau)}{\sum_{j=1}^{n} exp(v_j^T v/\tau)}, \quad (3)$$

where $v = f_\theta(x)$, n is the size of whole dataset and τ is a temperature parameter used for controlling the distribution concentration degree [13]. Following IR, we develop our instance-wise supervision loss and cluster-wise supervision loss at t-th epoch as

$$L_{instance}^t = - \sum_{i \in \mathcal{N}^s(t)} log(P(i|v_i(t))), \quad (4)$$

and

$$L_{cluster}^t = -\sum_{c=1}^{N_t} \sum_{i,j \in \mathcal{S}_c^s(t)} log(P(i|v_j(t))). \quad (5)$$

With Eq.(4) and Eq.(5), we can formally propose our loss function as

$$L_{pcp}^t = L_{instance}^t + L_{cluster}^t. \quad (6)$$

So far, the learning procedure of PCP can be summarized as Algorithm 1.

Warm-up. In early training epochs, the cluster number dramatically decreases, which challenges the stability and convergence of feature learning. In addition, early clustering may lead to network focus on low-level features. To alleviate this situation, a warm-up training strategy is implemented by adding a branch sharing the same architecture and weights with the original PCP branch to assist its early learning. Following
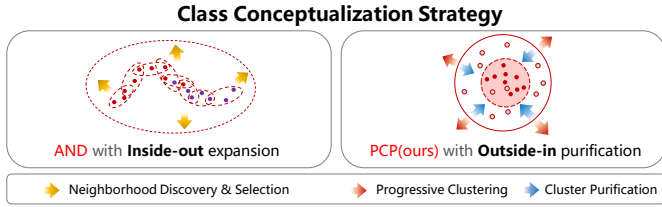
Class Conceptualization Strategy

AND with **Inside-out** expansion | PCP(ours) with **Outside-in** purification

Neighborhood Discovery & Selection | Progressive Clustering | Cluster Purification

Fig. 3: Comparison of class conceptualization strategies used by AND [14] and PCP.

TABLE I: Effects of the components in our approach with kNN classification accuracy.

DC [2]	PC	CP_r	CP_s	Acc
√	-	-	-	73.6
√	√	-	-	76.9
√	√	√	-	78.9
√	√	√	√	81.6

TABLE II: Evaluation of CP_r and CP_s under different filtering ratio.

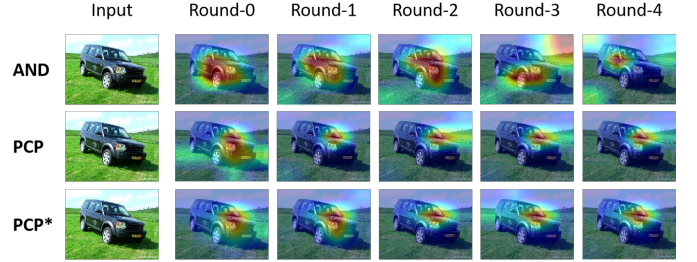| γ	0	0.3	0.5	0.7	0.9	0.95	0.99
CP_r	76.9	77.7	78.9	79.8	81.4	81.6	81.7
CP_r+CP_s	80.3	81.5	81.6	81.3	81.1	80.9	80.6

Fig. 4: Visualization of features extracted by AND and PCP, which clearly shows that the features learned by PCP steadily focus on objects. * denotes with warm up.

IS [29], the branch is implemented in a sample specific learning manner aiming at learning random data augmentation invariant and instance spread-out features. With warm-up for training a certain epochs, the additional IS branch is discarded for further improvement of the PCP branch. Figures in Table III and Table IV clearly show the effect of the warm-up training strategy.

D. Discussion

To show the advantages of PCP over existing UFL methods, we compare them from three aspects including reliability, stability, and efficiency.

Reliability. With PC and CP, PCP explores the underlying class information while mitigating the impact of clustering error. As shown in Fig. 3, PCP uses outside-in filtering strategy for class conceptualization, which generates reliable class consistent cluster by discarding noise samples. In contrast, AND [14] takes an inside-out expanding strategy to form class concept by merging neighborhoods. However, if the neighborhood size is large, it likely absorbs noise samples and drifts to be confused for feature learning; if the neighborhood size is small, the efficiency of model learning will be reduced, Table IV.

Stability. With a simple-yet-effect clustering correction mechanism, PCP gets rid of dependence of approximating the number of classes of the training set from which other clustering based methods suffer a lot It can be found that the learning procedure of PCP converges more stably, Fig. 1 and Fig. 4.

Efficiency. With the warm-up strategy, PCP extracts discriminative features in early training epochs, which not only improves the reliability and stability of feature learning but also results in fast convergence of training effect. Meanwhile, with quick decrease of cluster number, PCP is forced to efficiently perform class conceptualization. In contrast, AND sets the smallest neighbourhood size to avoid class drift, which limits its learning efficiency.

IV. EXPERIMENTS

Datasets. The learned feature representation is validated on image classification and object detection. For image classification, we use CIFAR [17] which contains 50,000/10,000 train/test images, and CIFAR10/CIFAR100 has 10/100 object classes with 6,000/600 images per class, all images with size 32 $\times$ 32. And IN-100 dataset which is selected from ImageNet [34]

with 100 classes. For fine-grained classification, we use CUB-200-2011 [3] dataset which consists of 5994/ 5994 train/test images with 200 bird species. For object detection, we use PASCAL VOC 2007 [7] which contains 20 classes, 5011 images in trainval set and 4952 images in test set.

Experimental Setting. Following the setting of [28], [29], we initialize the learning rate as 0.03 with the decreasing strategy that scaled-down by 0.1 at 120-th epoch and 0.01 at 160-th epoch. Then we set momentum to 0.9, weight decay to 0.0005, batch size to 128, temperature parameter τ to 0.1, and embedding feature space dimension to 128 for network training. For fair comparison, we train PCP models with 200 epochs (a round) as default, and choose $k = 200$ while implementing kNN evaluation. For the reliability of voting function for cluster purification, we implement cluster purification after 100 epochs training and take 15 epochs clustering results for voting function with two thresholds 0 and 3.0. For IS branch warm-up, the weight of loss in IS t decrease from 0.8 to 0.5 during epoch 0 to epoch 180 continuously, and the weight of PCP with $1 - t$. For multiple rounds training, we apply warm-up in the first round, the rest of rounds without warm-up and with the same decreasing strategy of learning rate. One-off AND is implemented with 2 rounds for the first round as warm-up. Unless stated otherwise, the reported data is implemented on CIFAR10 with ResNet18, and kNN performance adopted top-1 accuracy.

Evaluation Metrics. We use linear classification and weighted kNN to evaluate features from different layers [28]. With linear classification, we implement classifier by designing
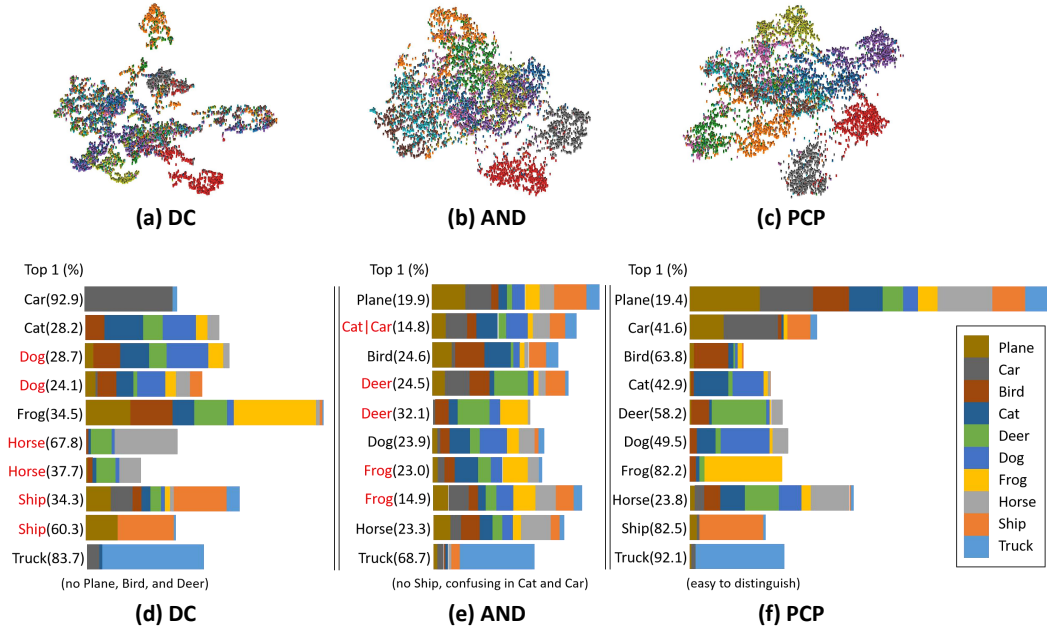
Fig. 5: Visualization of 2-dimensional t-SNE distributions of the feature space (a-c) and its class statistics under k-means (k=10) clustering results (d-f) by DC , AND and PCP. (Best viewed in color)

a fully connected (FC) layer. We are aiming to optimize a FC layer by minimizing cross-entropy loss. With weighted kNN, for a test image $\hat{x}$, we calculate the cosine similarity $s_i = \cos(v_i, \hat{v})$ for each of sample x_i belonging to the train set, where $\hat{v} = f_\theta(\hat{x})$. The set of top-$k$ nearest neighbours denoted as $\mathcal{N}_k$ is then used to predict the class according to $\omega_c = \sum_{i\in\mathcal{N}_k} \alpha_i \cdot 1(c_i = c)$, where $\alpha_i = exp(s_i/\tau)$ and ω_c denotes the value of voting function that sample $\hat{x}$ belongs to class c. The result label of sample $\hat{x}$ is c^* when $\omega_{c^*} = \max(\omega_0, \omega_1, \ldots, \omega_{n-1})$, where n denotes the number of categories in test set.

A. Component Analysis and Discussion

For fair comparison with DC, we drop cluster number (N_c) down to 100 to evaluate the effect of PCP components. In Table I, DC method achieves accuracy 73.6% by taking cluster result as pseudo label, we can see that PC improves the accuracy by 3.3%, unreliable sample filtering of cluster purification (CP_r) gains 2% and unstable sample filtering of cluster purification (CP_s) further gains 2.7%.

Cluster Purification. In Table II, we explore the effect of CP on PCP in detail. As we can see that with the increase of the unreliable sample filtering ratio γ from 0 to 0.99, the classification accuracy of PCP with only CP_r consistently increases from 76.9% to 81.7%. This can be mainly attributed to that the supervision offered by clustering contains less noise. However, with higher γ, few samples remain in the cluster which limits the model to discovery inter-sample relationship. Further with CP_s, lower γ leads to more noise samples, higher γ leads to model close to IR, best performance appears in intermediate. PCP can not only keep the cluster size but also effectively correct the error assignments that rebounds the

TABLE III: Performance under different cluster numbers. * denotes with warm-up.

N_c	10k	5k	3k	1k	100	10	5
DC [2]	82.9	83.0	82.1	80.6	73.6	62.0	56.9
PCP	81.7	81.6	81.8	82.3	81.6	82.0	81.8
PCP*	84.1	84.4	84.3	84.7	83.9	83.1	83.2

TABLE IV: Comparison of AND and PCP (w or w/o warm-up) under different training rounds (kNN accuracy). * denotes with warm-up. N denotes neighborhood size.

Round	0	1	2	3	4
AND [14] (N = 1)	80.0	83.9	85.0	85.9	86.3
AND [14] (N = 5)	80.0	83.8	85.0	85.6	85.8
AND [14] (N = 10)	80.0	83.8	84.9	85.1	85.1
AND [14] (N = 20)	80.0	83.3	84.3	84.6	84.6
PCP (Ours)	82.3	84.8	85.3	86.0	86.0
PCP* (Ours)	**84.7**	**86.4**	**86.7**	**87.0**	**87.3**

accuracy when γ is low compared with only CP_r. This further evaluates the effect of CP with no doubt. In the following experiments, we set γ to 0.5 as default.

Progressive Clustering. As shown in Table III, the performance of DC drops dramatically from 82.9% to 56.9% with the cluster number (N_c) decreasing from 10,000 to 5 which is below the true class number of 10. In contrast, with the class conceptualization and the noise sample correction mechanism PC and CP, the performance of PCP is insensitive to N_c, which demonstrates the superior reliability and stability of PCP. With warm-up, PCP outperforms DC with a significant margin which is 4.1% higher when N_c is equal to 1k (set as default in the following experiments).

TABLE V: Comparison of classification accuracy (kNN) on CIFAR10. F-S denotes fully supervised. $*$ denotes the performance produced by our implementation. $+$ denotes 5 rounds training. The compared results of IR/IS/AND are directly transcribed from their references.

Model	Random	F-S*	DC* [2]	IR [28]	IS [29]
Acc	32.1	93.1	80.6	80.8	83.6

Model	Random	AND [14]	PCP	AND$^+$	**PCP$^+$**
Acc	32.1	84.2	84.7	86.3	**87.3**

TABLE VI: Evaluation on CIFAR10 and CIFAR100 with AlexNet by performing linear classifier on the features from $conv5$, and kNN from FC. F-S denotes fully supervised. $*$ denotes our rerunning. AND in our implementation has two rounds (one-off) while PCP with one round.

Classifier	Weighted kNN (FC)		Linear Classifier ($conv5$)	
Dataset	CIFAR10	CIFAR100	CIFAR10	CIFAR100
DC* [2]	70.3	27.4	77.1	44.0
IR* [28]	68.1	39.6	76.6	49.5
IS* [29]	76.4	46.3	78.7	51.2
AND* [14]	76.1	44.2	79.2	52.8
PCP (Ours)	**77.1**	**48.4**	**79.9**	**53.0**
F-S	91.9	69.7	91.8	71.0

Curriculum Learning. To enhance discriminative power of learned features, AND derives curriculum learning with five training rounds to select class-consistent neighbourhoods, PCP follows the same round number for training. In Table IV we can see that the convergence of PCP is faster and the performance is higher than those of AND, and PCP with warm-up further converges faster and performs better which show the superior efficiency of PCP. From the visualization in Fig. 4, that the features learned by PCP steadily focus on objects shows the superior stability of PCP.

Class Conceptualization. To show the effect of class conceptualization, we visualize the feature distribution of the whole test set of CIFAR10 by embedding the 128-dimensional feature space into a 2-dimensional space by t-SNE [25]. In Fig. 5 (a-c), both DC and PCP models are trained with $N_c = 10$ and we can see that the feature representations learned by DC and AND are less discriminative as many features with different classes mixed up. In contrast, the features extracted by PCP are better assigned to different clusters.

To quantitatively show the superiority of PCP, we cluster learned features to 10 clusters, then identify the class pseudo-label of each cluster by the category which the majority of features in the cluster belong to. As shown in Fig. 5 (d-f), PCP forms 10 clusters dominated by samples from 10 different classes. For DC, 3 categories ($Plane$, $Bird$ and $Deer$) are missed in that each of the 3 categories (Dog, $Horse$ and $Ship$) dominates two clusters respectively. AND misses the class of $Ship$, and shows same proportions in Cat and Car. We can easily distinguish each category in PCP while DC and AND look slightly confusing. So far we can argue that better class conceptualization is of essential importance to improve the performance of unsupervised clustering methods.

TABLE VII: Comparison with MoCo [11] on ImageNet subset IN-100 with AlexNet by performing linear classifier(LC) on the features from $conv5$, and kNN from FC.

Classifier	kNN (FC)	LC ($conv5$)
MoCo [11]	50.1	55.4
PCP(Ours)	50.7	56.9

TABLE VIII: Comparison of fine-grained classification performance. $*$ denotes our rerunning. PCP is implemented with one round.

Model	Random*	IR [28]	DC* [2]	AND [14]	**PCP (Ours)**
Acc	2.6	11.6	13.1	14.1	**16.9**

B. Image Classification

To further evaluate the proposed PCP approach, we conduct experiments on image classification and fine-grained classification to compare with baseline methods. The backbone is ResNet18 and the classification is evaluated by kNN. The results of IR/IS/AND are directly transcribed from the corresponding references. DC is reimplemented with 1000 clusters. As shown in Table V, PCP model achieves the best performance of 84.7% accuracy, and after 5 rounds the accuracy of 87.3% outperforms AND with 1%. Considering the fully-supervised accuracy is only 93.1%, the performance of PCP is extraordinary under unsupervised setting.

For generalization verification, we then compare image classification accuracy on CIFAR10 & CIFAR100 with a standard AlexNet backbone attached by a weighted kNN (FC layer) and a linear classifier ($conv5$ layer). The figures in Table VI clearly show that PCP outperforms other methods, and especially with a significant margin for CIFAR100 evaluated by kNN.

We further compare our method with MoCo [11] (rerun the code https://github.com/facebookresearch/moco, default batch size as 128, queue size as 6528, and 2 GeForce RTX 2080Ti GPUs) on ImageNet100 subset IN-100 under the same settings, the performance of PCP is higher than MoCo with 0.6% for kNN top-1 accuracy and 1.5% for linear classifier accuracy, Table VII.

Fine-grained Classification. PCP is further evaluated on the more challenging fine-grained classification task. We follow the settings with $N_c = 500$ and batch size as 64 on the CUB-200-2011 dataset. As shown in Table VIII, PCP significantly outperforms IR by 5.3%, DC by 3.8% and AND by 2.5%.

TABLE IX: Comparison of object detection performance. AND in our implementation has two rounds (one-off) while PCP with one round.

Model	Random	DC [2]	IR [28]	AND [14]	**PCP (Ours)**	F-S
mAP	0.5	27.8	30.6	36.9	**39.8**	46.1

C. Object Detection

Object detection experiments are conducted to evaluate the effectiveness of the features learned by PCP for down-stream tasks. We implement Faster R-CNN [22] with a ResNet18 backbone pre-trained on the CIFAR100 dataset and fine-tuned on the PASCAL VOC 2007 dataset.

For pretrained backbone, the first residual stage and all the batch normalization layers are frozen. Among the overall fine-tuning 30 epochs, the learning rate is first set to 0.001 and decreased 10 times after 25 epochs, and batch size is fixed as 8. Compared with the randomly initialized feature model, which is hard to converge, SOTA models significantly improve the mAP performance, Table IX. PCP achieves 39.8% mAP which outperforms AND by 2.9%. This shows the effectiveness of the features learned by PCP for down-stream tasks.

V. CONCLUSION

In this paper, we proposed Progressive Cluster Purification (PCP) method for unsupervised feature learning. To alleviate the impact of noise samples, we elaborately designed the Progressive Clustering (PC) strategy to gradually expand the cluster size consistently with the growth of the model representation capability and the Cluster Purification (CP) mechanism to reduce unreliable and unstable noise samples in each cluster to a significant extent. Resultantly, PCP is able to reliably, stably and efficiently learn discriminative and representative features. With warm-up training strategy, PCP avoids network focusing on low-level feature for early clustering. Extensive experiments on classification and detection benchmarks demonstrated that the proposed PCP approach improved the classical clustering method and provided a fresh insight into the unsupervised learning problem.

ACKNOWLEDGEMENT

REFERENCES

[1] P. Bachman, R. Hjelm and W. buchwalter, "Learning Representations by Maximizing Mutual Information Across Views", in *NeurIPS*, 2019. II

[2] M. Caron, P. Bojanowski, A. Joulin and M. Douze, "Deep Clustering for Unsupervised Learning of Visual Features", in *ECCV*, 2018. I, 1, II, III-A, I, III, V, VI, VIII, IX

[3] W. Catherine, B. Steve, W. Peter, P. Pietro and B. Serge, "The Caltech-UCSD Birds-200-2011 Dataset", in *Computation & Neural Systems Technical Report*, 2011. IV

[4] A. Coates and A. Ng, "Learning Feature Representations with K-Means", in *Neural Networks*, 2012. II

[5] C. Doersch, A. Gupta and A. Efros, "Unsupervised Visual Representation Learning by Context Prediction", in *ICCV*, 2015. I, II

[6] A. Dosovitskiy, P. Fischer, J. Springenberg, M. Riedmiller and T. Brox, "Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks", in *TPAMI*, 2016. II

[7] M. Everingham, L. Van, C. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge", in *IJCV*, 2010. IV

[8] Z. Feng, C. Xu and D. Tao, "Self-Supervised Representation Learning by Rotation Feature Decoupling", in *CVPR*, 2019. II

[9] S. Gidaris, P. Singh and K. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations", in *ICLR*, 2018. I, II

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets", in *NeurIPS*, 2014. I, II

[11] K. He, H. Fan, Y. Wu, S. Xie and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning", in *CVPR*, 2020. VII, IV-B

[12] G. Hinton, S. Osindero and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets", in *Neural Computation*, 2006. II

[13] G. Hinton, O. Vinyals and J. Dean, "Distilling the Knowledge in a Neural Network", in *arXiv*, 2015. III-C

[14] J. Huang, Q. Dong, S. Gong and X. Zhu, "Unsupervised Deep Learning by Neighbourhood Discovery", in *ICML*, 2019. I, II, 3, III-D, IV, V, VI, VIII, IX

[15] X. Ji, J. Henriques and A. Vedaldi, "Invariant Information Clustering for Unsupervised Image Classification and Segmentation", in *ICCV*, 2019. II

[16] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes", in *arXiv*, 2013. I, II

[17] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images", 2009. IV

[18] G. Larsson, M. Maire and G. Shakhnarovich, "Learning Representations for Automatic Colorization", in *ECCV*, 2016. I, II

[19] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye and W. Wang, "Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning", in *AAAI*, 2020. II

[20] M. Noroozi and P. Favaro, "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles", in *ECCV*, 2016. I, II

[21] M. Noroozi, H. Pirsiavash and P. Favaro, "Representation Learning by Learning to Count", in *ICCV*, 2017. I, II

[22] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", in *NeurIPS*, 2015. IV-C

[23] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines", in *Artificial intelligence and statistics*, 2009. II

[24] Y. Tian, D. Krishnan and P. Isola, "Contrastive Multiview Coding", in *arXiv*, 2019. II

[25] L. Van and G. Hinton, "Visualizing Data using t-SNE", in *JMLR*, 2008. IV-A

[26] P. Vincent, H. Larochelle, Y. Bengio and P. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders", in *ICML*, 2008. I, II

[27] F. Wan, P. Wei, Z. Han, J. Jiao and Q. Ye, "Min-Entropy Latent Model for Weakly Supervised Object Detection", in *TPAMI*, 2019. I

[28] Z. Wu, Y. Xiong, S. Yu and D. Lin, "Unsupervised Feature Learning via Non-Parametric Instance Discrimination", in *CVPR*, 2018. I, 1, II, III-A, III-C, IV, IV, V, VI, VIII, IX

[29] M. Ye, X. Zhang, P. Yuen and S. Chang, "Unsupervised Embedding Learning via Invariant and Spreading Instance Feature", in *CVPR*, 2019. III-C, IV, V, VI

[30] R. Zhang, P. Isola and A. Efros, "Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction", in *CVPR*, 2017. I, II

[31] X. Zhang, F. Wan, C. Liu, R. Ji and Q. Ye, "FreeAnchor: Learning to Match Anchors for Visual Object Detection", in *NeurIPS*, 2019. I

[32] C. Zhuang, A. Zhai and D. Yamins, "Local Aggregation for Unsupervised Learning of Visual Embeddings", in *ICCV*, 2019. II

[33] Y. Yao, C. Liu, D. Luo, Y. Zhou and Q. Ye, "Video Playback Rate Perception for Self-supervised Spatio-Temporal Representation Learning", in *CVPR*, 2020. II

[34] J. Deng, W. Dong, R. Socher, L. Li, K. Li and F. Li, "ImageNet: A large-scale hierarchical image database", in *CVPR*, 2009. IV

[35] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou and W. Wang, "SEED: Semantics Enhanced Encoder-Decoder Framework for Scene Text Recognition", in *CVPR*, 2020. I