Learning Representation for Clustering Via Prototype Scattering and Positive Sampling

Zhizhong Huang, *Graduate Student Member, IEEE*, Jie Chen,
Junping Zhang, *Senior Member, IEEE*, and Hongming Shan, *Senior Member, IEEE*

Abstract—Existing deep clustering methods rely on either contrastive or non-contrastive representation learning for downstream clustering task. Contrastive-based methods thanks to negative pairs learn uniform representations for clustering, in which negative pairs, however, may inevitably lead to the class collision issue and consequently compromise the clustering performance. Non-contrastive-based methods, on the other hand, avoid class collision issue, but the resulting non-uniform representations may cause the collapse of clustering. To enjoy the strengths of both worlds, this paper presents a novel end-to-end deep clustering method with prototype scattering and positive sampling, termed ProPos. Specifically, we first maximize the distance between prototypical representations, named prototype scattering loss, which improves the uniformity of representations. Second, we align one augmented view of instance with the sampled neighbors of another view—assumed to be truly positive pair in the embedding space—to improve the within-cluster compactness, termed positive sampling alignment. The strengths of ProPos are avoidable class collision issue, uniform representations, well-separated clusters, and within-cluster compactness. By optimizing ProPos in an end-to-end expectation-maximization framework, extensive experimental results demonstrate that ProPos achieves competing performance on moderate-scale clustering benchmark datasets and establishes new state-of-the-art performance on large-scale datasets. Source code is available at https://github.com/Hzzone/ProPos.

Index Terms—Contrastive learning, deep clustering, representation learning, self-supervised learning, unsupervised learning

✦

1 INTRODUCTION

DEEP clustering is gaining considerable attention as it aims to learn the representation of images and perform clustering in an end-to-end fashion. The main thrust to advance deep clustering is the self-supervised representation learning, including contrastive learning [1], [2] and non-contrastive learning [3], [4].

Remarkably, existing deep clustering methods heavily rely on contrastive representation learning, referred to as

---

• *Zhizhong Huang, Jie Chen, and Junping Zhang are with the Shanghai Key Lab of Intelligent Information Processing and the School of Computer Science, Fudan University, Shanghai 200433, China. E-mail: {zzhuang19, chenj19, jpzhang}@fudan.edu.cn.*
• *Hongming Shan is with the Institute of Science and Technology for Brain-inspired Intelligence, MOE Frontiers Center for Brain Science, 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. E-mail: hmshan@fudan.edu.cn.*

contrastive-based methods [5], [6], [7], [8], [9], [10], [11]. Specifically, they are usually built upon MoCo [1] or SimCLR [2], requiring specially designed losses [5], [7], [8], [9], [10] or an extra pre-training stage for more discriminative representations [6], [11]. Although achieving promising clustering results, contrastive-based methods usually require a large number of negative examples to learn uniform representations in an embedding space where all instances are well-separated. The involved negative pairs may *inevitably* lead to the class collision issue that different instances from the same semantic class are regarded as negative pairs and are *wrongly* pushed away, which hampers the downstream clustering. An alternative perspective on this issue is to separate the typical contrastive loss into two terms [12]: 1) *alignment* term to improve the closeness of positive pairs, and 2) *uniformity* term to encourage instances to be uniformly distributed on a unit hypersphere by pushing away the negative pairs. Apparently, the uniformity term could introduce class collision issue [13] as the constructed negative pairs may not be *truly* negative.

Different from contrastive learning, non-contrastive learning *only* involves the alignment term using the representations of one augmented view to predict another. The non-contrastive learning can avoid the class collision issue as there are no negative pairs. Lacking the uniformity term in contrastive loss, it is not guaranteed to learn uniform representations [12], [14], which may cause the collapse of downstream clustering—most samples are assigned to few clusters. This phenomenon would even worsen when learning in conjunction with extra clustering losses introduced by current state-of-the-art deep clustering methods [7], [8]; see Section 5.4 and supplemental Fig. 1, available online.

To enjoy the strengths of both worlds, we propose a novel end-to-end deep clustering method, ProPos, with two novel techniques: prototype scattering loss and positive sampling alignment. First, considering that different prototypes/clusters are *truly* negative pairs, we propose to perform contrastive learning over prototypical representations, in which two augmented views of the same prototypes are positive pairs and different prototypes are negative pairs. This yields the proposed prototype scattering loss or PSL, which maximizes the between-cluster distance so as to learn uniform representations towards well-separated clusters. Second, to improve the within-cluster compactness, we further propose to align one augmented view of the instance with the randomly sampled neighbors of another view that are assumed to be *truly* positive pairs in the embedding space, which we refer to as positive sampling alignment or PSA. Compared to conventional alignment between two augmented views, the proposed PSA takes into account the neighboring samples in the embedding space, improving the within-cluster compactness. Moreover, we optimize ProPos in an expectation-maximization (EM) framework, in which we iteratively perform E-step as estimating the instance pseudo-labels via spherical k-means and M-step as minimizing the proposed losses.

The contributions are summarized as follows:

- We propose a novel end-to-end deep clustering method, termed ProPos, which enjoys the advantages of contrastive and non-contrastive representation learning: avoidable class collision issue, uniform representations for improved clustering stability, well-separated clusters, and improved within-cluster compactness.
- We propose a novel prototype scattering loss or PSL, which can align one augmented view of prototypes with another view and maximize the between-cluster scattering on the unit hypersphere, hence maximizing the inter-cluster distance for uniform representations.
- We propose a positive sampling alignment or PSA to extend instance alignment by taking into account neighboring positive examples in the embedding space, which can improve the within-cluster compactness.
- By optimizing ProPos in an EM framework, extensive experimental results on several benchmark datasets demonstrate that ProPos outperforms the existing state-of-the-art methods by a significant margin, especially for large-scale datasets.

The remainder of this paper is organized as follows. A brief review on the related work of self-supervised learning and deep clustering is given in Section 2, followed by the contrastive and non-contrastive representation learning in Section 3. We present the proposed PSL and PSA as well as our ProPos in Section 4. Experimental results are reported and analyzed in Section 5, where Section 5.2 further justifies the motivation. Finally, Section 6 discusses the relations to previous works, followed by a concluding summary in Section 7.

2 RELATED WORK

This section briefly surveys the development of self-supervised learning and deep clustering.

2.1 Self-Supervised Learning

Previous self-supervised learning (SSL) methods for representation learning attempt to capture the data distribution using generative models [15], [16] or learn the representations through some special designed pretext tasks [17], [18], [19], [20]. In recent years, contrastive learning methods [1], [2], [21] have shown promising results for both representation learning and downstream tasks. Contrastive representation learning requires a large number of negative examples to achieve instance-wise discrimination in an embedding space where all instances are well-separated. The constructed negative pairs usually require a large batch size [2], memory queue [1], or memory bank [21], which not only bring extra computational cost but also give rise to class collision issue [22] that the semantic similar instances are pushed away since they could be regarded as negative pairs. For example, MoCo [1] uses a memory queue to store the consistent representations output by a moving-averaged encoder. However, the class collision issue remains unavoidable. Some attempts have been made to address this issue [23], [24], [25].

On the contrary, the recent studies of SSL demonstrate that the negative examples are not necessary, termed non-contrastive methods [2], [3], [26]. Recently, mask image modeling (MIM) such as MAE [27] arises a new trend for self-supervised learning that leverages ViT [28] to directly reconstruct mask images. However, MIM may not be ready for deep clustering yet as ViT needs to be trained on large datasets such as ImageNet-1k [29] and it does not learn discriminative representations for deep clustering.

In summary, SSL methods mainly focus on inducing transferable representations for the (supervised) downstream tasks instead of grouping the data into different semantic classes for deep clustering.

2.2 Deep Clustering

Deep clustering [30], [31], [32], [33], [34], [35], [36], [37], [38] has been significantly advanced by self-supervised representation learning. Most of deep clustering methods are based on contrastive learning by exploiting the discriminative representations, learned from contrastive learning, to assist the downstream clustering tasks [6], [11], [38] or simultaneously optimize representation learning and clustering [7], [9], [10], [39], [40]. For example, SCAN [6] yields the confident pseudo-labels by the pre-trained SimCLR model, and IDFD [9] proposes to perform both instance discrimination and feature decorrelation. Although GCC [41] and WCL [42] select the neighbor samples from a graph as pseudo-positive examples for contrastive loss, however, they still suffer from the class collision issue as these selected examples may not be *truly* positive. In a nutshell, all of them are built upon the contrastive learning framework, in which they require a large number of negative examples to maintain uniform representations, inevitably leading to class collision issue.

Although some attempts have been made to use non-contrastive learning, such as BYOL [3], to avoid class

collision issue [43], [44], they produce inferior results as they suffer from the collapse of downstream clustering due to the non-uniform representations.

Our work advances deep clustering via the two novel techniques to address current drawbacks. First, the proposed PSL maximizes the between-cluster distance, which leads to uniform representations, hence alleviating the collapse of downstream clustering. Second, the proposed positive sampling alignment improves within-cluster compactness. As a result, the proposed ProPos can enjoy the strengths of both worlds: avoidable class collision issue, uniform representation for improved clustering stability, well-separated clusters, and improved within-cluster compactness. In Section 6, we discuss the differences from existing methods including CC [8], GCC [41], WCL [42], PCL [7], and instance-reweighted contrastive loss [45].

3 PRELIMINARY

Here, we briefly introduce representative contrastive learning and non-contrastive learning methods.

3.1 Contrastive Learning

Contrastive learning methods perform instance-wise discrimination [21] using the InfoNCE loss [46]. Formally, assume that we have one instance x, its augmented version x^+ by using random data augmentation, and a set of M negative examples drawn from the dataset, $\{x_1^-, x_2^-, \ldots, x_M^-\}$. The contrastive learning aims to learn an embedding function $f(\cdot)$ that maps x onto a unit hypersphere. The corresponding InfoNCE loss for one instance is defined as follows:

$$-\log \frac{\exp\left(\frac{f(x)^T f(x^+)}{\tau}\right)}{\exp\left(\frac{f(x)^T f(x^+)}{\tau}\right) + \sum_{i=1}^{M} \exp\left(\frac{f(x)^T f(x_i^-)}{\tau}\right)} \quad (1)$$

$$\approx \underbrace{-\frac{f(x)^T f(x^+)}{\tau}}_{\text{instance alignment}} + \underbrace{\log \sum_{i=1}^{M} \exp\left(\frac{f(x)^T f(x_i^-)}{\tau}\right)}_{\text{instance uniformity}}, \quad (2)$$

where the representation $f(x)$ is ℓ_2 normalized on a unit hypersphere, and the temperature τ controls the concentration level of representations.

Intuitively, the InfoNCE loss aims to pull together the positive pair (x, x^+) from two different data augmentations of the same instance, and push x away from M negative examples of other instances. As discussed in [12], the InfoNCE loss in Eq. (1) can be approximately decoupled into two terms in Eq. (2): the first term refers to as *instance alignment*, and the second term *instance uniformity*. Despite the alignment term pulls together the positive pair, the key to avoiding representation collapse is the uniformity term, which makes the negative examples uniformly distributed on the unit hypersphere. Although alleviating the collapse of downstream clustering, the negative examples may inevitably lead to the class collision issue [13], hurting the representations for clustering.

3.2 Non-Contrastive Learning

Non-contrastive methods only optimize the alignment term in Eq. (2) to match the representations between

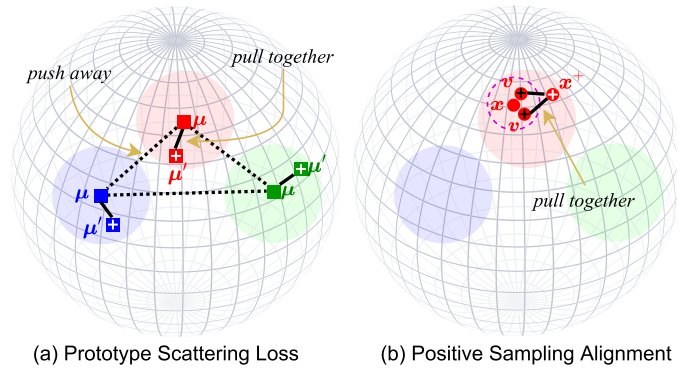(a) Prototype Scattering Loss (b) Positive Sampling Alignment

Fig. 1. Illustration of the proposed two key techniques in ProPos. (a) The proposed prototype scattering loss to encourage alignment and maximize the between-cluster distance. (b) The proposed positive sampling alignment to encourage the alignment between sampled neighbors of one view with another for better within-cluster compactness.

two augmented views [3], [4]. Specifically, they often leverage an online, a target, and a predictor network to bridge the gap between these two views with stop gradient operation to avoid representation collapse. In particular, if $\tau = 0.5$, the loss used in [3], [4] can be written as:

$$-2g(f(x))^T f'(x^+) = \|g(f(x)) - f'(x^+)\|_2^2 - 2, \quad (3)$$

where $g(\cdot)$, $f(\cdot)$, and $f'(\cdot)$ are the predictor, online, and target networks, respectively; $g(f(x))$ and $f'(x^+)$ are ℓ_2-normalized. Without using negative pairs, non-contrastive learning avoids the class collision issue. However, due to the lack of uniformity, they tend to produce non-uniform representations that usually result in the collapse of downstream clustering, making them unstable for deep clustering; see Section 5.2.

4 METHOD

The goal of deep clustering is to learn the representation of images and perform the clustering task simultaneously. Our ProPos advances deep clustering via two novel techniques: prototype scattering loss (PSL) and positive sampling alignment (PSA) detailed in Sections 4.1 and 4.2, respectively. We then present the overview of ProPos and its EM optimization process in Section 4.3.

4.1 Prototype Scattering Loss

A good clustering is supposed to have well-separated prototypes/clusters. Assuming that the dataset has K clusters, where K is a predefined number and assumed to be known, it naturally constructs a contrastive loss for these K prototypes as for one prototype, the remaining $K-1$ prototypes are definitely negative examples. Therefore, we propose a prototype scattering loss or PSL, which encourages the prototypical alignment between two augmented views and the prototypical uniformity, hence maximizing the inter-cluster distance.

Specifically, assume we obtain K prototypes from one view in the embedding space, $\{\mu_1, \mu_2, \ldots, \mu_K\}$, and another K prototypes from another view, $\{\mu_1', \mu_2', \ldots, \mu_K'\}$, our proposed PSL, illustrated in Fig. 1a, is defined as follows:

$$\mathcal{L}_{\mathrm{psl}} = \frac{1}{K}\sum_{k=1}^{K} -\log \frac{\exp\left(\frac{\boldsymbol{\mu}_k^T \boldsymbol{\mu}_k'}{\tau}\right)}{\exp\left(\frac{\boldsymbol{\mu}_k^T \boldsymbol{\mu}_k'}{\tau}\right) + \sum_{j=1, j\neq k}^{K} \exp\left(\frac{\boldsymbol{\mu}_k^T \boldsymbol{\mu}_j}{\tau}\right)} \tag{4}$$

$$\approx \underbrace{\frac{1}{K}\sum_{k=1}^{K} -\frac{\boldsymbol{\mu}_k^T \boldsymbol{\mu}_k'}{\tau}}_{\text{prototypical alignment}} + \underbrace{\frac{1}{K}\sum_{k=1}^{K} \log \sum_{j=1, j\neq k}^{K} \exp\left(\frac{\boldsymbol{\mu}_k^T \boldsymbol{\mu}_j}{\tau}\right)}_{\text{prototypical uniformity}}. \tag{5}$$

Here, the cluster centers $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_k'$ are computed within a mini-batch $\mathcal{B}$ as follows:

$$\boldsymbol{\mu}_k = \frac{\sum_{\boldsymbol{x}\in\mathcal{B}} p(k|\boldsymbol{x})f(\boldsymbol{x})}{\|\sum_{\boldsymbol{x}\in\mathcal{B}} p(k|\boldsymbol{x})f(\boldsymbol{x})\|_2}, \tag{6}$$

$$\boldsymbol{\mu}_k' = \frac{\sum_{\boldsymbol{x}\in\mathcal{B}} p(k|\boldsymbol{x})f'(\boldsymbol{x})}{\|\sum_{\boldsymbol{x}\in\mathcal{B}} p(k|\boldsymbol{x})f'(\boldsymbol{x})\|_2}, \tag{7}$$

where $p(k|\boldsymbol{x})$ is the cluster assignment posterior probability. When $K > |\mathcal{B}|$, it is obvious that the mini-batch cannot cover all clusters. To this end, we zero out the losses and logits of empty clusters for each iteration. During training, it is critical to estimate accurate $p(k|\boldsymbol{x})$ to optimize the proposed PSL, so we adopt an EM framework that alternately uses a k-means clustering for every epoch at the E-step and then minimizes Eq. (4) at the M-step, which will be detailed later.

Intuitively, our PSL for prototypes is similar to conventional contrastive loss in Eq. (1) for instances. The key difference is that PSL will not cause the class collision issue as the prototypes are definitely negative examples for each other, which is more suitable for deep clustering. However, the cluster centers may not be as accurate as expected during the early epochs of training. For an accurate initialization, following [7], PSL will be involved in training after the finish of the learning rate warmup.

Similarly, PSL in Eq. (4) can be approximately divided into Eq. (5): *prototypical alignment* and *prototypical uniformity*. On one hand, the prototypical alignment is to align the prototypes between two views, which can stabilize the update of the prototypes. On the other hand, the prototypical uniformity is to encourage the prototypes to be uniformly distributed on a unit hypersphere, which can maximize the inter-cluster distance. We note that there are two cluster-level losses related to ours: ProtoNCE [7] to improve cluster compactness and CC [8] to contrast cluster assignments; we discuss the major differences in Section 6.

Uniform Representation and Well-Separated Clusters. In the context of non-contrastive learning for deep clustering, the lack of uniformity term in Eq. (3) fails to produce uniform representations, which may cause severe collapse of downstream clustering. PSL overcomes this drawback by maximizing the inter-clusters distance between prototypical representations, which yields uniform representations and well-separated clusters.

4.2 Positive Sampling Alignment

Following previous discussion, the negative examples are essential for contrastive-based deep clustering to learn uniform representations, at the cost of inevitable class collision issue [22] that harms the within-cluster compactness. On the other hand, non-contrastive learning can avoid the class

collision issue by *only* optimizing the instance alignment. However, the conventional instance alignment in Eq. (3) only encourages the representation of one augmented view to be close to another view. In the context of clustering, such compactness is instance-wise and neutral for deep clustering, since the semantic class information cannot be captured at only instance level.

To improve within-cluster compactness for conventional instance alignment while avoiding class collision issue, we propose to optimize the *opposite* of the uniformity instead, i.e. the compactness within clusters. We aim to encourage the neighboring examples around one augmented view—sampled from the embedding space and assumed to be *truly* positive pairs—to be aligned with another view, as shown in Fig. 1b. Our motivation is that although we cannot guarantee the negative pairs constructed from the dataset are *truly negative*, we can certainly assume that the neighboring samples around one view in the embedding space are *truly positive* with respect to another view and belong to the same class. Therefore, we propose a positive sampling alignment to extend the instance alignment in Eq. (3) by taking into account the neighboring samples towards improved within-cluster compactness.

The key step of PSA is to sample the neighboring examples $\boldsymbol{v}$. A natural way is modeling the representation of one augmented view of an instance as a continuous distribution in the embedding space. We thus introduce a Gaussian distribution thanks to its simplicity, which can be formulated as follows:

$$\boldsymbol{v} \sim \mathcal{N}\big(f(\boldsymbol{x}), \sigma^2 \boldsymbol{I}\big), \tag{8}$$

where $\boldsymbol{I}$ represents the identity matrix and σ is a positive hyperparameter controlling how many samples around one view can be treated as positive pairs with another view. However, the sampled examples from Eq. (8) cannot allow the error to be backpropagated through the network to update the network parameters. By leveraging the reparametrization trick [47], the positive sampling can be implemented as follows:

$$\boldsymbol{v} = f(\boldsymbol{x}) + \sigma\boldsymbol{\epsilon}, \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}). \tag{9}$$

We then extend the instance alignment in Eq. (3) by taking into account the neighboring samples. With only sampling one example from the Gaussian distribution, the positive sampling alignment (PSA) can be formally defined as:

$$\begin{aligned}\mathcal{L}_{\mathrm{psa}} &= \|g(\boldsymbol{v}) - f'(\boldsymbol{x}^+)\|_2^2 \\ &= \|g(f(\boldsymbol{x}) + \sigma\boldsymbol{\epsilon}) - f'(\boldsymbol{x}^+)\|_2^2.\end{aligned} \tag{10}$$

Here, when $\sigma = 0$, PSA reduces to Eq. (3).

Avoidable Class Collision Issue. In the context of non-contrastive learning without negative examples, our PSA can guarantee that the positive examples around one instance, sampled in the embedding space, are from the same cluster and form *truly positive* pairs. Therefore, optimizing PSA loss would not cause class collision issue that exists in contrastive-based deep clustering methods. We discuss the difference from [41], [42] that sample positive examples from the dataset in Section 6.
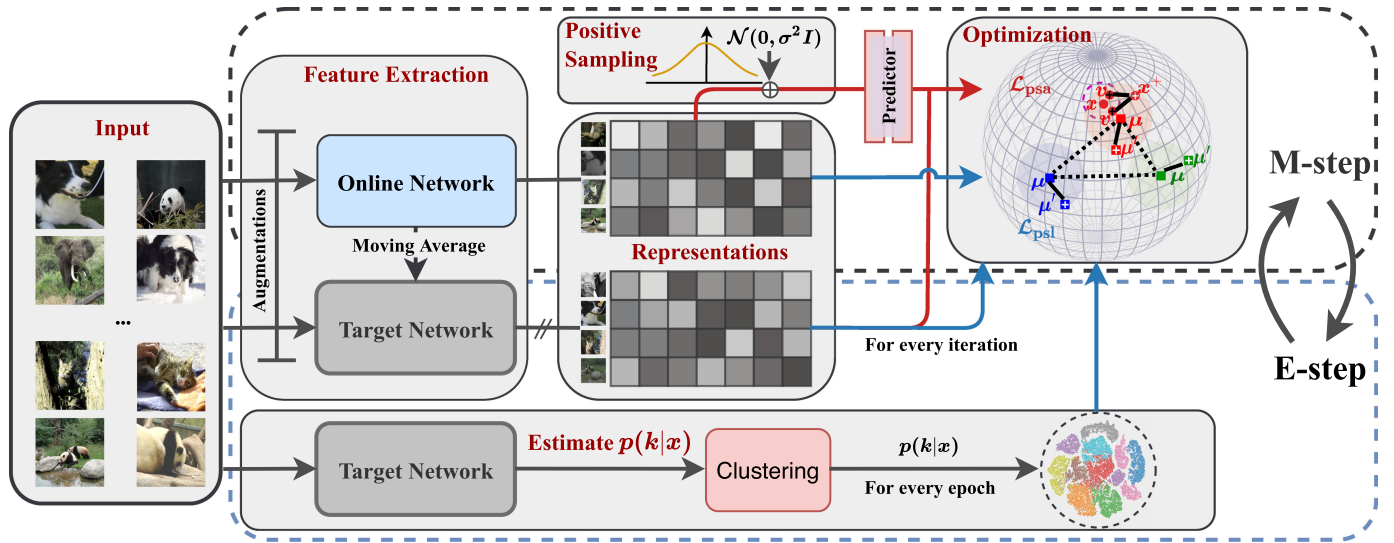
Fig. 2. The overall framework of the proposed ProPos in an EM framework.

Improved Within-Cluster Compactness. Unlike conventional instance alignment for non-contrastive learning, our PSA in Eq. (10) encourages neighboring examples around one augmented view—either different augmented examples of the same instance or same/different augmented examples of different instances within the same cluster—to be positive pairs with another view. This helps to improve within-cluster compactness.

4.3 Overview of ProPos and Its Optimization

We present the overview of our ProPos in Fig. 2 and optimize it in an EM framework to facilitate the understanding of the training procedure.

4.3.1 Overview

We build ProPos upon non-contrastive learning framework similar to BYOL, which is comprised of three networks: an online, a target, and a predictor. During training, the parameters of target network are momentum updated (*a.k.a* moving averaged) from the ones of online network, following

$$\boldsymbol{\theta}_{\text{target}} = m\boldsymbol{\theta}_{\text{target}} + (1 - m)\boldsymbol{\theta}_{\text{online}}, \tag{11}$$

where $m \in [0, 1)$ is the coefficient, and $\boldsymbol{\theta}_*$ denotes the parameters. Two different random data augmentations from the same inputs are fed into the network to optimize the proposed losses in an EM framework. Specifically, k-means clustering is performed in the E-step at the beginning of each epoch to obtain the $p(k|\boldsymbol{x})$, which is fixed in the latter training to optimize the proposed PSL. The $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}'_k$ are computed from online and target networks using Eqs. (6) and (7), respectively. Furthermore, PSA is applied to the representations from online network, which are then passed through predictor network for alignment.

4.3.2 EM Framework

The optimization of ProPos is done in an EM framework, where E-step and M-step are detailed as follows; supplemental Section 1, available online, presents detailed derivations.

E-step: This step aims to estimate $p(k|\boldsymbol{x})$ for the proposed PSL. We perform spherical k-means algorithm on the features extracted from the target network since the target network performs more stable and yields more consistent clusters, similar to BYOL and MoCo. Although we need an additional k-means clustering to obtain the cluster pseudo-labels $p(k|\boldsymbol{x})$ for every r epochs, we found that even with a larger $r > 1$, our method can still produce consistent performance improvement over the baseline methods. Therefore, our method will not introduce much computation cost and is robust to the cluster pseudo-labels; see detailed results in Sections 5.5 and 5.6. Finally, with $p(k|\boldsymbol{x})$, we build the prototypical representations within a mini-batch without additional memory.

Algorithm 1. Training Algorithm

Input: Dataset $\mathcal{D} = \{\boldsymbol{x}\}$;
    Functions $f(\cdot)$ and $f'(\cdot)$
Output: Clustering results $\{p(k|\boldsymbol{x})\}$.
repeat
    E-step: update $\{p(k|\boldsymbol{x})\}$ for each sample in $\mathcal{D}$ using k-means clustering
    M-step: **repeat**
        Randomly sample a mini-batch $\mathcal{B}$ from $\mathcal{D}$
        for *each $\boldsymbol{x}_i$ in $\mathcal{B}$* **do**
            Randomly augment $\boldsymbol{x}$ and $\boldsymbol{x}^+$
            Compute cluster centers using Eqs. (6) and (7)
            $\mathcal{L}_{\text{psl}} \leftarrow$ Eq. (4)
            $\mathcal{L}_{\text{psa}} \leftarrow$ Eq. (10)
        end
        $\mathcal{L} \leftarrow$ Eq. (12)
        Update f' with momentum moving average
        Update f with SGD optimizer
    until *an epoch finished;*
until *reaching max epochs;*

M-step: Combining PSL in Eq. (4) and the PSA in Eq. (10) yields our objective function for M-step as follows:

$$\mathcal{L} = \mathcal{L}_{\text{psa}} + \lambda_{\text{psl}}\mathcal{L}_{\text{psl}}, \tag{12}$$

TABLE 1
Summary of the Datasets

Dataset	Split	# Samples	# Classes	Image Size
CIFAR-10	Train+Test	60,000	10	32×32
CIFAR-20	Train+Test	60,000	20	32×32
STL-10	Train+Test	13,000	10	96×96
ImageNet-10	Train	13,000	10	96×96
ImageNet-Dogs	Train	19,500	15	96×96
Tiny-ImageNet	Train	100,000	200	224×224
ImageNet-1k	Train	1,281,167	1,000	224×224

where λ_{psl} controls the balance between two loss components. Therefore, there are only two hyper-parameters in the loss function, including: σ in $\mathcal{L}_{psa}$ and the loss weight λ_{psl}; see the effects of hyper-parameters in Section 5.5.

The training procedure of the proposed ProPos is presented in Algorithm 4.3.2.

Relations Between PSL and PSA. A good clustering model should have well-separated clusters and within-cluster compactness. On the one hand, PSL encourages well-separated clusters by maximizing inter-cluster distance, which, however, cannot improve within-cluster compactness. On the other hand, PSA can pull together the sampled neighboring examples around one augmented view and another view, which can further improve within-cluster compactness. By combing these two losses in Eq. (12), we expect ProPos can improve deep clustering towards well-separated clusters and within-cluster compactness.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets

We conducted experiments on seven benchmark datasets, including **CIFAR-10** [48], **CIFAR-20** [48], **STL-10** [49], **ImageNet-10** [31], **ImageNet-Dogs** [31], **Tiny-ImageNet** [50], and **ImageNet-1k** [29], which are summarized in Table 1. We note that CIFAR-20 contains 20 superclasses of CIFAR-100. STL-10 includes extra unlabeled images. ImageNet-10, ImageNet-Dogs, and Tiny-ImageNet are the widely-used subsets of ImageNet-1k [29], containing 10, 15, 200 classes, respectively. This paper follows the experimental settings widely used in deep clustering work [9], [10], [31], [35], [36], including the image size, backbone, and train-test split. For image size, we use 32×32 for CIFAR-10 and CIFAR-20, 96×96 for STL-10, ImageNet-10, and ImageNet-Dogs, and 224×224 for Tiny-ImageNet and ImageNet-1k. For train-test split, we use the whole datasets including training and testing set for CIFAR-10 and CIFAR-20 while both labeled and unlabeled data are employed for STL-10.

5.1.2 Backbones

We use ResNet-34 [51] as the backbone for fair comparisons to report the main results on moderate-scale benchmark datasets. We use ResNet-18 [51] on Tiny-ImageNet and ResNet-50 [51] on ImageNet-1k, following the literature. Unless noted otherwise, we use ResNet-18 for the rest of the experiments. Since the image sizes of CIFAR-10 and CIFAR-

20 are relatively small, following [2], we replace the first convolution layer of kernel size 7×7 and stride 2 with a convolution layer of kernel size 3×3 and stride 1, and remove the first max-pooling layer for all experiments on CIFAR-10 and CIFAR-20.

5.1.3 Implementation Details

We train all methods with 1,000 epochs, strictly following the literature [9], [10], and adopt the stochastic gradient descent (SGD) optimizer and the cosine decay learning rate schedule with 50 epochs for learning rate warmup. The base learning rate for MoCo v2 [52], BYOL [3], and ProPos were 0.05, scaled linearly with the batch size (LearningRate = 0.05×BatchSize/256). Note that the learning rates for predictor networks of BYOL and ProPos are $10\times$ as the learning rate of feature extractor. It is relatively important to achieve satisfactory performance, as discussed in [3], [4].

For other hyperparameters of ProPos, the temperature τ, λ_{psl} for prototypical scattering loss, and σ for positive sampling were set as 0.5, 0.1, and 0.001, respectively. The mini-batch size was 512 for MoCo and 256 for the remaining methods. ProPos was trained on 4 NVIDIA V100 GPUs.

Regarding CC [8] and PCL [7], we tried our best to reproduce their results for fair comparisons. For CC, we used their official code. For PCL, under the fair conditions of MoCo, we set the loss weight of ProtoNCE to 0.01 and the number of clusters to $\{250, 500, 1000\}$ following the suggestions of authors, which we found can achieve the best results. We integrated CC and PCL into BYOL by adding their losses without changing other settings.

5.1.4 Configurations of SSL Frameworks

We adopt the same data augmentations as SimCLR [2], including ResizedCrop, ColorJitter, Grayscale, and HorizontalFlip. We have removed GaussianBlur since we only used a small image size for all datasets. We also strictly follow the settings of BYOL [3]. Specifically, despite the standard ResNet backbones, the projection and predictor networks have the architectures of FC-BN-ReLU-FC, where the projection dimension and hidden size were 256 and 4,096 for both two networks, respectively. For fair comparisons, we have also set the projection dimension of MoCo v2 as 256. We have used symmetric loss for all methods, i.e., swapping two data augmentations to compute twice loss. For the momentum hyperparameter $m \in [0, 1)$, we set it to 0.996 for both BYOL and ProPos same as [3] and 0.99 for MoCo v2. For MoCo v2, the queue size, temperature for InfoNCE loss, weight decay were 4,096, 1.0, and 1.0×10^{-4}, respectively. We have not employed SyncBN in ProPos. We note that SyncBN would introduce much additional computation cost. Instead, we adopt the shufflingBN in MoCo to avoid the trivial solution of non-contrastive learning.

5.2 Empirical Justification

We provide an empirical justification on how the proposed ProPos improves representation learning for deep clustering from the aspects of PSL and PSA.
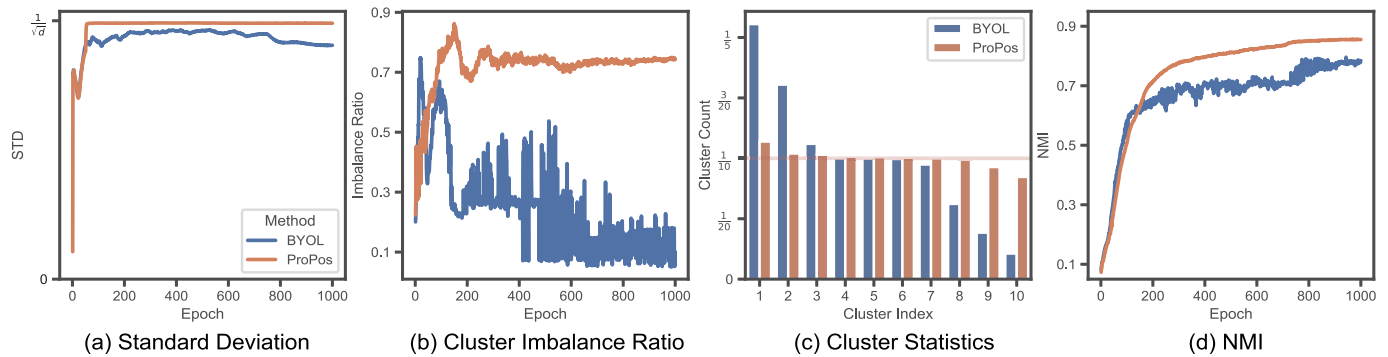
Fig. 3. Detailed comparison between BYOL [3] and ProPos on CIFAR-10 in terms of (a) standard deviation (STD) of ℓ_2-normalized features to evaluate the uniformity, (b) cluster imbalance ratio computed by $\min(\{N_k\}_{k=1}^{K})/\max(\{N_k\}_{k=1}^{K})$ to show how balanced the clusters are, (c) cluster statistics, or the sorted number of samples in each cluster for the model at 1000-th epoch, and (d) normalized mutual information (NMI) between the clustering results and ground-truth labels.

5.2.1 The Role of PSL

The non-uniform representations produced by non-contrastive representation learning would lead to the collapse of downstream clustering where most samples are assigned to few clusters. We emphasize that it is desirable to avoid the trivial solution for deep clustering. For example, CC [8] and etc [41], [53] have usually employed an entropy term in loss function to regularize the model equally assigning the images into different clusters. This paper mainly investigates this phenomenon of non-contrastive representation learning for deep clustering, and the theoretical analysis can be found in [14], [54].

Our ProPos can encourage cluster uniformity for representations via PSL. Here, we use the representative non-contrastive learning method, Bootstrap Your Own Latent (BYOL), as the representation learning for deep clustering. We performed spherical k-means on the learned representations for the clustering task with 10 different initializations. Following [4], we use the standard deviation (STD) of ℓ_2-normalized representations to measure the uniformity. Ideally, if the ℓ_2-normalized representations are uniformly distributed on a unit hypersphere, we have $\text{STD}[\boldsymbol{z}'] \approx 1/\sqrt{d}$, where $\boldsymbol{z}'$ and d are the ℓ_2-normalized version and the dimension of the feature representation $\boldsymbol{z}$. To justify the effectiveness of ProPos, we conducted the following experiments in terms of the uniformity of the representations, the collapse of clustering, and the clustering performance.

First, we visualize the uniformity of representations in Fig. 3a. Taking a look at STDs during the training stage, ProPos produces higher STDs, while BYOL performs unstable with the STDs gradually decreasing. Note that a higher standard deviation close to $1/\sqrt{d}$ indicates more uniform representations. Our ProPos yields more uniform representations than BYOL. Most importantly, the uniformity of ProPos is rather stable during training.

Second, we further visualize the cluster imbalance to measure the potential collapse during clustering. More specifically, we compute the cluster imbalance ratio between the cluster with least samples and the cluster with most samples, $\min(\{N_k\}_{k=1}^{K})/\max(\{N_k\}_{k=1}^{K})$, where N_k is the number of samples in k-th cluster. A higher value indicates more balanced clusters. In addition, we also show the cluster statistics or the sorted number of samples in each cluster for

the model at 1000-th epoch. The results of cluster imbalance during training and the cluster statistics at the final epoch are shown in Figs. 3b and 3c. Fig. 3b shows that the k-means clustering process of ProPos produces more balanced clusters with a higher cluster imbalance ratio. On the contrary, the clusters of BYOL are highly imbalanced, which is consistent with decreasing STDs. Moreover, Fig. 3c shows that the cluster statistics for ProPos are approximately and equally assigned to different clusters, compared to almost long-tailed assignments of BYOL. The more balanced clusters validate that ProPos can alleviate the collapse of k-means clustering over BYOL.

Finally, Fig. 3d shows the clustering performance comparison between the proposed ProPos and BYOL, which is measured by the normalized mutual information (NMI) between clustering results and ground-truth labels. We can see that ProPos produces higher and more stable NMIs than BYOL. In line with the analysis above, we conclude that directly applying BYOL to deep clustering, although avoiding class collision issue, suffers from the collapse of k-means clustering due to the non-uniform representations. In contrast, ProPos with PSL yields more uniform representations and well-clustered samples.

5.2.2 The Role of PSA

PSA assumes that the sampled neighbors are truly positive examples with respect to another view, i.e., they belong to the same semantic classes. To validate this assumption, we investigate the behavior of PSA during training by checking whether the semantic classes of input examples have been changed.

First, we perform k-NN classification to predict the classes of both inputs and their sampled neighbors from testing set, and then use the proportion of sampled neighbors that have preserved original classes as the preservation rate. We run the experiments 10 times with different σ for PSA. As shown in Fig. 4a, even at the early training stage, the sampled neighbors well preserve original classes for $\sigma < 0.005$. The preservation rate drops as expected for $\sigma = 0.01$ since PSA may sample the instances from another cluster for large σ.

Second, we showcase some randomly-selected input images whose nearest neighbors have been changed during the sampling procedure, Specifically, Fig. 4b visualizes the input images, their nearest neighbors, and the sampled
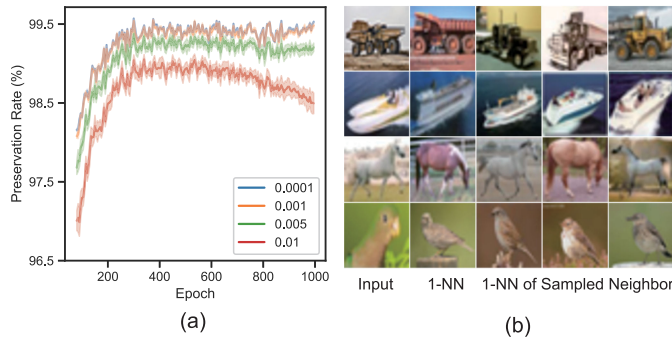
Fig. 4. Visualization of sampled neighbors for ProPos on CIFAR-10: (a) The odds of sampled neighbors that have preserved their original semantic classes under different σ during training; and (b) The input images, the 1-nearest-neighbors (1-NN) of input images, and three sample neighbors at 100th epoch.

neighbors. We note that we replaced the images of sampled neighbors with their nearest neighbors since it is too hard to reconstruct the image from embedding space. Obviously, the sampled neighbors belong to the same class as the input images even at the early stage of training (i.e. 100th epoch).

In summary, the results suggest that the sampled neighbors could be truly positive examples both quantitatively and qualitatively so that PSA can improve the within-cluster compactness.

5.3 Main Results

In this section, we evaluate ProPos with previous state-of-the-art clustering methods on various benchmark datasets. We divide these methods into 5 types, listed in Table 2 from top to bottom: i) methods without using contrastive learning (IIC to PICA); ii) multi-stage methods requiring step-by-step pretraining or finetuning (SCAN and NMM); iii) methods directly outputting the cluster assignments (CC to TCC); iv) methods learning general

representations (MoCo to BYOL); and v) methods improving representation learning for clustering (IDFD to ProPos). We strictly follow the experimental settings of previous works [9], [10] for fair comparisons. We reproduced PCL [7], SimSiam [2], and BYOL [3] under the same conditions, and directly use the their learned representations for k-means clustering.

In terms of qualitative results of clustering, we visualize the learned representations by t-SNE [57] for four different training epochs throughout the training process in supplemental Fig. 2, available online, and the outlier points produced by the model at 1000-th epoch on CIFAR-10 in supplemental Fig. 3, available online. For fair comparisons, supplemental Table 1, available online, presents the results excluding the testing set for CIFAR-10/20 and using a larger image size for ImageNet-10/Dogs.

5.3.1 Results on Moderate-Scale Datasets

The comparisons on five moderate-scale datasets are reported in Table 2. The results of MoCo are referred from [10]. For fair comparison, we excluded SPICE [11] in Table 2 since it requires multiple pre-training stages. ProPos achieves significant performance improvement on all benchmark datasets, demonstrating the superiority of ProPos for deep clustering to capture the semantic class information.

On the ImageNet-10, our ProPos achieves competitive performance as compared to IDFD [9] since this dataset is relatively small with only 13 k images, which cannot arise discriminative differences for current state-of-the-art methods. On the ImageNet-Dogs, a fine-grained dataset containing different species of dogs from the ImageNet dataset, there are almost 20% improvements over previous state-of-the-art work. The contrastive-based methods cannot handle this kind of dataset due to severe class collision issue that pushes away the instances from the same class. Meanwhile, IDFD can deal with this problem to some degree thanks to the

TABLE 2
Clustering Results (%) of Various Methods on Five Benchmark Datasets

Method	CIFAR-10			CIFAR-20			STL-10			ImageNet-10			ImageNet-Dogs		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
IIC [36]	51.3	61.7	41.1	-	25.7	-	43.1	49.9	29.5	-	-	-	-	-	-
DCCM [35]	49.6	62.3	40.8	28.5	32.7	17.3	37.6	48.2	26.2	60.8	71.0	55.5	32.1	38.3	18.2
PICA [55]	56.1	64.5	46.7	29.6	32.2	15.9	-	-	-	78.2	85.0	73.3	33.6	32.4	17.9
SCAN [6]	79.7	88.3	77.2	48.6	50.7	33.3	69.8	80.9	64.6	-	-	-	-	-	-
NMM [56]	74.8	84.3	70.9	48.4	47.7	31.6	69.4	80.8	65.0	-	-	-	-	-	-
CC [8][1]	70.5	79.0	63.7	43.1	42.9	26.6	76.4	85.0	72.6	85.9	89.3	82.2	44.5	42.9	27.4
MiCE [10]	73.7	83.5	69.8	43.6	44.0	28.0	63.5	75.2	57.5	-	-	-	42.3	43.9	28.6
GCC [41]	76.4	85.6	72.8	47.2	47.2	30.5	68.4	78.8	63.1	84.2	90.1	82.2	49.0	52.6	36.2
TCL [38][1]	81.9	88.7	78.0	52.9	53.1	35.7	79.9	86.8	75.7	87.5	89.5	83.7	62.3	64.4	51.6
TCC [39]	79.0	90.6	73.3	47.9	49.1	31.2	73.2	81.4	68.9	84.8	89.7	82.5	55.4	59.5	41.7
MoCo [1]	66.9	77.6	60.8	39.0	39.7	24.2	61.5	72.8	52.4	-	-	-	34.7	33.8	19.7
SimSiam [2]	78.6	85.6	73.6	52.2	48.5	32.7	65.9	71.6	57.2	83.1	92.1	83.3	58.3	67.4	50.1
BYOL [3]	81.7	89.4	79.0	55.9	56.9	39.3	71.3	82.5	65.7	86.6	93.9	87.2	63.5	69.4	54.8
IDFD [9]	71.1	81.5	66.3	42.6	42.5	26.4	64.3	75.6	57.5	89.8	95.4	90.1	54.6	59.1	41.3
PCL [7]	80.2	87.4	76.6	52.8	52.6	36.3	71.8	41.0	67.0	84.1	90.7	82.2	44.0	41.2	29.9
ProPos (**ours**)	**88.6**	**94.3**	**88.4**	**60.6**	**61.4**	**45.1**	75.8	86.7	73.7	89.6	**95.6**	90.6	69.2	74.5	62.7

[1] CC and TCL use a large image size of 224×224 for all datasets.
The best and second best results are shown in bold and underline, respectively. The works most related to our method are IDFD and PCL that improve the representations for clustering.

TABLE 3
Clustering Results (%) on Tiny-ImageNet

Method	Tiny-ImageNet		
	NMI	ACC	ARI
DCCM [35]	22.4	10.8	3.8
PICA [55]	27.7	9.8	4.0
CC [8]	34.0	14.0	7.1
GCC [41]	34.7	13.8	7.5
MoCo [1]	34.2	16.0	8.0
PCL [7]	35.0	15.9	8.7
SimSiam [2]	35.1	20.3	9.4
BYOL [3]	36.5	19.9	10.0
ProPos (**ours**)	**40.5**	**25.6**	**14.3**

We trained ProPos using ResNet-18.

feature decorrelation along with the instance discrimination. Furthermore, BYOL and SimSiam can achieve significant improvements, which suggests a great potential for non-contrastive representation learning for deep clustering *without* suffering class collision issue. However, our ProPos has introduced further substantial improvements for deep clustering by addressing existing issues. Although CC [8] and TCL [38] achieve the better NMIs on STL-10, we highlight that they use a large image size of 224×224 for all datasets, which is not fair to ours.

5.3.2 Results on Large-Scale Datasets

To validate the effectiveness of our method on large-scale datasets with large number of classes, we evaluate it on Tiny-ImageNet and ImageNet-1k, which contains 200 and 1000 classes, respectively. The results are reported in Tables 3 and 4. We note that we exclude the methods that their authors did not report results on the corresponding datasets from Tables 3 and 4. For ImageNet-1k, we strictly follow the settings in [7] and employed Adjusted Mutual Information (AMI) to evaluate the performance, the results in Table 4 are referred from [7], and we trained a ResNet-50 for 200 epochs same as [7]. The results show the strong generalization ability of ProPos on complex datasets with a large number of clusters.

To further show the effectiveness of ProPos for downstream classification task, we conducted the linear evaluation on ImageNet-1k dataset, and provide the comparisons with the recent state-of-the-art methods in Table 5. Following the same settings in [4], we have trained the linear classifier for 90 epochs with a batch size of 4,096, an initial learning rate of 1.6, cosine learning rate decay, and the SGD optimizer of momentum 0.9 and weight decay 0. Under fair conditions, ProPos outperforms other competitors by a clear margin.

5.4 Ablation Study

Here, we perform detailed ablation studies with both quantitative and qualitative comparisons to provide more insights into why ProPos performs well for deep clustering.

5.4.1 Quantitative Ablation Study

The quantitative results are shown in Table 6.

Ablation Study of PSL and PSA. ProPos w/o PSA improves the baseline results by a large margin while ProPos w/o PSL achieves marginal improvements, which indicates that PSL

TABLE 4
Clustering Results (%) on ImageNet-1k
Using ResNet-50

Method	AMI
DeepCluster [20]	28.1
MoCo [1]	28.5
PCL [7]	41.0
ProPos (**ours**)	**52.5**

is the key to boosting the clustering performance. However, PSA plays an important role in two parts. First, simply using the PSA can stable and further improve the performance than the baseline BYOL (only instance alignment loss), especially when the number of semantic classes increases for CIFAR-20. Second, PSA can make ProPos better for clustering when PSL is used in conjunction with the PSA to pull together the neighbor examples. This is because the well-seperated clusters by PSL can further ensure that PSA samples the positive neighbors that are in the same semantic classes than the one without PSL. On the other hand, PSL only considers inter-cluster distance, and cannot benefit within-cluster compactness. Therefore, the combination of the positive sampling and PSL achieves the best clustering results, where PSL aligns the cluster centers between two augmented views and maximizes the inter-cluster distance, and PSA improves the within-cluster compactness.

To further explore the effect of PSL, following Eq. (5) we split PSL into alignment and uniformity terms denoted as PSL-alignment and PSL-uniformity in Table 6. It is clear that the performance gain from the alignment term is marginal while the gain from the uniformity term is significant. For only alignment term, we compute the loss after predictor

TABLE 5
Linear Evaluation on ImageNet-1k Dataset

Method	Backbone	Pre-training		ACC
		Batch size	Epochs	
Jigsaw [18]	AlexNet	256	-	34.6
Rotation [58]	AlexNet	128	100	38.7
DeepCluster [20]	AlexNet	256	500	41.0
InstDisc [21]	ResNet-50	256	200	54.0
LocalAgg [59]	ResNet-50	128	200	60.2
CMC [60]	ResNet-50	-	200	66.2
SimCLR [2]	ResNet-50	256	200	64.3
MoCo [1]	ResNet-50	256	200	60.6
MoCo v2 [52]	ResNet-50	256	200	67.5
PCL [7]	ResNet-50	256	200	67.6
IFND [61]	ResNet-50	256	200	69.7
BYOL [62]	ResNet-50	4096	200	70.6
SimSiam [4]	ResNet-50	256	200	70.0
ProPos (**ours**)	ResNet-50	256	200	**72.2**
CPC [46]	ResNet-101	512	-	48.7
SeLa [63]	ResNet-50	1024	400	61.5
PIRL [64]	ResNet-50	1024	800	63.6
SimCLR [2]	ResNet-50	4096	1000	69.3
BYOL [62]	ResNet-50	4096	1000	74.3
SwAV [26]	ResNet-50	4096	800	75.3

We Report the Top-1 Classification Accuracy (%) by Training a Linear Classifier; the Results are Adopted From Corresponding Papers. The Upper Group Above CPC Uses More Fair Conditions, e.g., Backbone and Training Epoch.

TABLE 6
Ablation studies (NMI/ACC/ARI) for Different Self-Supervised Learning Frameworks, Positive Sampling Alignment (PSA), and Prototype Scattering Loss (PSL) for ProPos

Method	CIFAR-10			CIFAR-20		
	NMI	ACC	ARI	NMI	ACC	ARI
CC [8]	66.1±0.3	74.6±0.3	58.3±0.4	46.4±0.3	45.0±0.1	29.5±0.2
CC + PSL	74.3±0.4	83.4±0.5	69.6±1.0	48.3±0.2	49.1±0.2	32.2±0.4
PCL [7]	77.6±0.1	85.5±0.1	73.4±0.0	50.0±0.3	48.6±0.7	32.7±0.4
BYOL [3]	79.4±1.7	87.8±1.7	76.6±2.8	55.5±0.6	53.9±1.6	37.6±0.9
BYOL + CC	76.6±3.1	86.3±2.7	73.8±4.7	51.0±2.0	48.9±3.0	33.3±2.9
BYOL + PCL	74.4±2.3	85.3±0.9	71.4±1.4	49.7±0.7	46.9±0.7	27.8±1.5
ProPos (**ours**)	85.1±0.5	91.6±0.4	83.5±0.7	**58.2**±0.3	**57.8**±0.2	**42.3**±0.3
ProPos w/o PSL	79.4±0.9	87.9±0.5	76.4±1.1	57.0±0.0	55.0±0.6	39.8±1.1
ProPos w/o PSA	83.4±1.2	90.3±0.9	81.1±1.7	56.6±0.4	55.1±0.5	40.7±1.0
ProPos w/o PSL-uniformity	79.6±0.7	87.8±1.5	76.5±2.1	56.7±0.3	56.6±1.4	39.7±1.1
ProPos w/o PSL-alignment	**85.3**±0.2	**92.1**±0.1	**84.4**±0.3	57.2±0.3	57.3±0.6	41.7±0.5

The best and second best results are shown in bold and underline, respectively.

network instead of feature extractor, otherwise, representation collapse will turn out. This indicates that uniformity is more important than alignment which can scatter the prototypes to encourage the uniform representations to address the issues in Section 5.2. However, the alignment term is essential to stabilize the training process, as demonstrated in the results for CIFAR-20 with more clusters.

Ablation Study of Self-Supervised Learning Framework. To alleviate the bias of self-supervised learning framework, we conduct experiments in two folds. First, we integrate CC [8] and PCL [7] into BYOL; the results in Table 6 show that both of them compromise clustering performance and become unstable. This is because CC contrasts the cluster probability not helpful for representation learning, and PCL would collapse without negative examples; see supplemental Fig. 1, available online, for detailed analysis. We emphasize that BYOL has addressed the class collision issue. Instead, this paper proposes the ProPos with PSL to scatter the prototypes by maximizing inter-cluster distance and positive sampling to improve within-cluster compactness. Second, we replace the cluster head of CC with our PSL on the representations while keeping other hyper-parameters unchanged. Although class collision issue remains, the significant improvements over CC on both datasets suggest that 1) PSL over representation prototypes is better than the

one over cluster probabilities, and 2) PSL can be generalized to other self-supervised learning frameworks.

5.4.2 Qualitative Ablation Study

Fig. 5 visualizes the distribution of representations learned from BYOL, ProPos w/o PSL, ProPos w/o PSA, and our ProPos. ProPos w/o PSA leverages PSL to discriminate different clusters by maximizing the inter-cluster distance, and produces more uniform representations. Although ProPos w/o PSL with only PSA achieves marginal improvements and does not produce a significant difference than BYOL, the positive sampling can further improve the within-cluster compactness with only PSL via the sampling-based instance alignment loss to pull together the neighbor samples.

5.4.3 Effect of Predefined Number of Clusters

In the above experiments, the number of clusters is predefined as the number of ground-truth classes, which cannot be identified in the practical scenarios. To this end, we conduct experiments on CIFAR-10 and CIFAR-20 with different number of clusters, i.e., $K \in \{5, 10, 20, 30, 40, 50\}$. We reported NMIs following [20] in Fig. 6. We note that the predefined K of ProPos during the training of ProPos is the same as the K in k-means clustering process for evaluation.

a) BYOL
(NMI=78.4%)

b) ProPos w/o PSL
(NMI=78.8%)

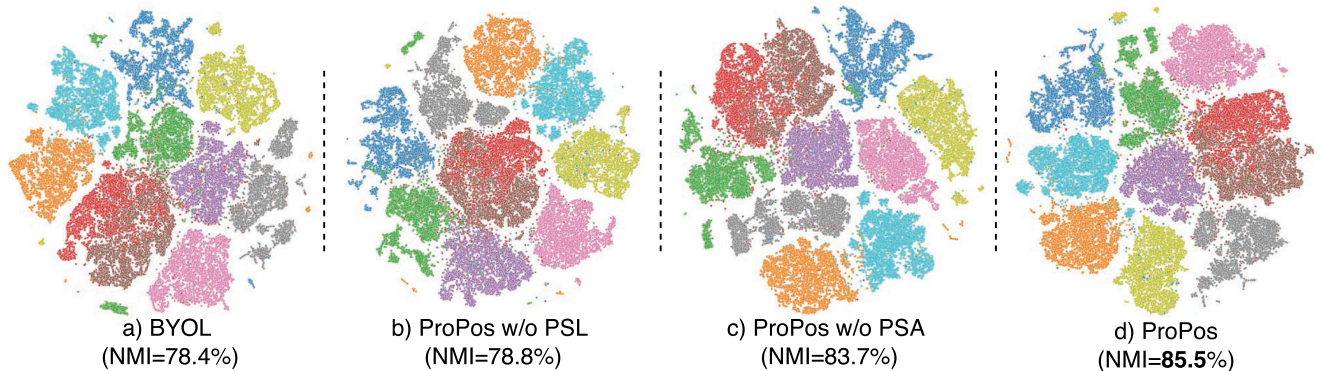c) ProPos w/o PSA
(NMI=83.7%)

d) ProPos
(NMI=**85.5**%)

Fig. 5. Visualization of feature representations learned by different representation learning frameworks and our proposed ProPos on CIFAR-10 with t-SNE. Zoom in for better view.
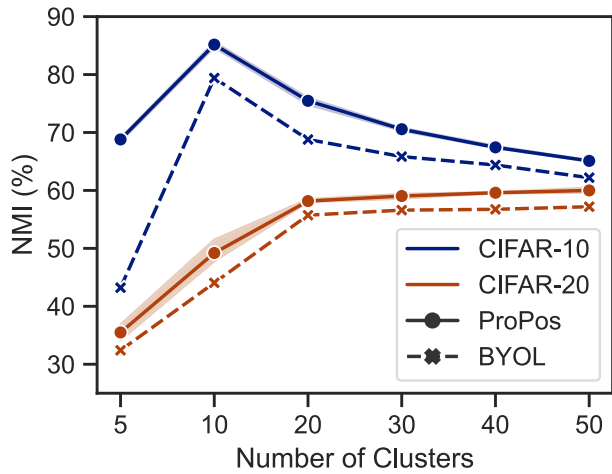
Fig. 6. The effect of the predefined number of clusters K on CIFAR-10/20 datasets.

To further investigate the influences of K, we also reported the results of vanilla BYOL during k-means clustering.

The results in Fig. 6 demonstrate that BYOL and ProPos have the same behavior on these two datasets. For over-clustering cases (K is larger than the true number of classes), the trends on these two datasets are opposite. Specifically, over-clustering leads to a performance drop for CIFAR-10, but it leads to an increase for CIFAR-20. However, our ProPos can still produce large improvements over BYOL with the same predefined number of clusters. The opposite results are due to the significant difference between these two datasets. Although having the same number of samples, CIFAR-10 has 10 distinct classes while CIFAR-20 has, in fact, 100 classes but uses 20 super-classes instead. Same trends are also consistently reported in [20]. In other cases, if the representations are well aligned within the same semantic clusters, the

over-clustering would try to destroy the structures of the clusters and push the semantically similar examples away, which certainly compromises the clustering performance. For under-clustering cases, the clustering performance has been significantly harmed for both two datasets and two methods. In supplemental Fig. 4, available online, we have visualized the learned representations for under-clustering, which shows that the examples from the same semantic classes can be still clustered together.

5.5 Hyperparameter Analysis

To investigate the effect of different hyperparameters, we conduct extensive experiments on the CIFAR-10/20 datasets. For the projection dimension, backbone, and data augmentation of SSL, we adopted the BYOL as the baseline method. The results are reported in Fig. 7.

5.5.1 Frequency of Performing k-Means Clustering

ProPos performs k-means clustering for every r epochs. Here, we study how different r influences the clustering performance. The results in Fig. 7a demonstrate ProPos is robust to large r and the cluster pseudo-labels, which means it is not necessary to perform clustering for every epoch so that the computation cost can be significantly reduced. In summary, we suggest that r can be set to $[1, 8]$ by considering the datasets and computation resources.

5.5.2 The Hyperparameter σ in PSA

The hyperparameter σ in PSA controls the degree of positive sampling. Taking a look at $\sigma \sim [0, 10^{-3}]$ in Fig. 7b, although introducing the positive sampling into BYOL causes a slight drop on CIFAR-10, the clustering performance becomes more stable as evidenced by the standard deviation. This is because the neighbors of one sample are
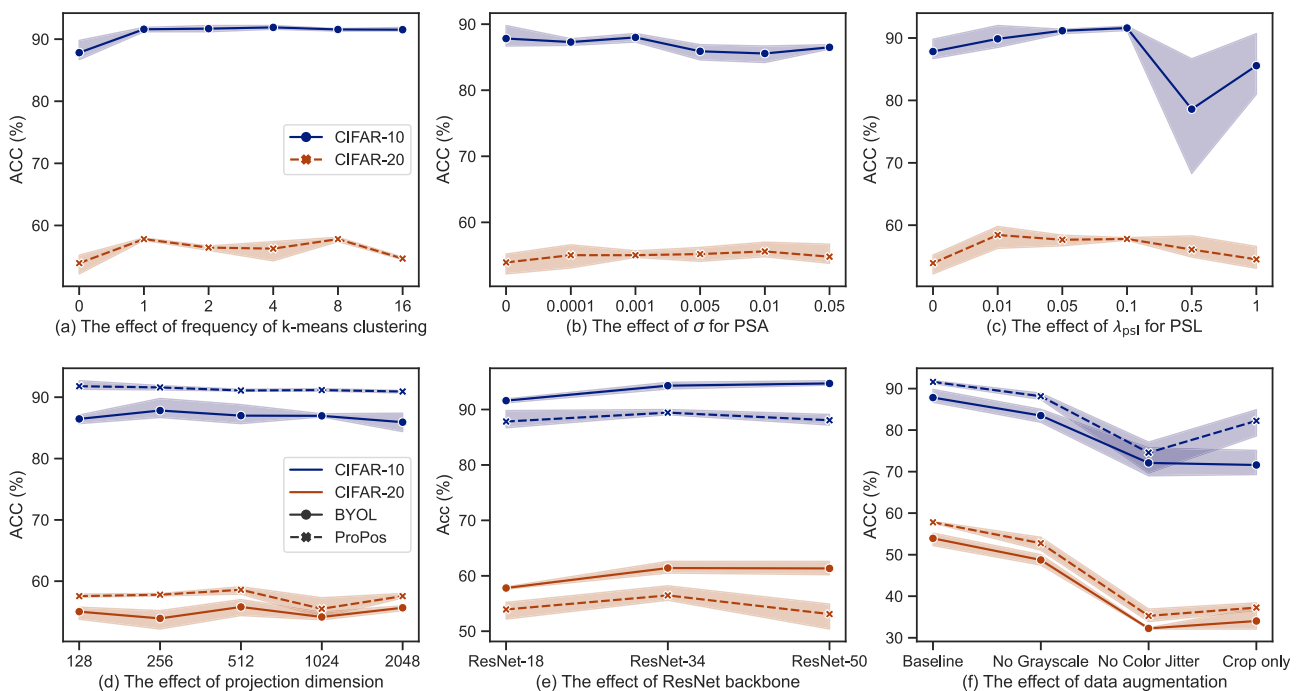

Fig. 7. Effects of different hyperparameters in ProPos.

TABLE 7
The Training time (hours) in the Settings of 1,000 Epochs, 4 V100 GPUs, and ResNet-50

Method	CIFAR-10	CIFAR-20	STL-10	ImageNet-10	ImageNet-dogs	Tiny-ImageNet
BYOL [3]	9.0	9.0	14.7	1.7	2.6	13.0
ProPos $(r = 1)$	10.9(+1.9)	11.0(+2.0)	15.8(+1.1)	2.7(+1.0)	3.7(+1.1)	15.7(+2.7)

regarded as positive examples. Besides, the performance for CIFAR-20 has increased over baseline with the standard deviation reduced. These results indicate that positive sampling can improve the stability of performance. However, when σ is too large, the performance becomes unstable and drops a lot. It is not surprising since during positive sampling with large σ, the instances from other clusters could be sampled and regarded as positive examples. Therefore, we suggest setting σ to a small value, saying $(0, 10^{-3}]$.

5.5.3 The Hyperparameter λ_{psl} for PSL

The hyperparameter λ_{psl} controls the importance of PSL. The results in Fig. 7c suggest that ProPos is robust to different choices on CIFAR-20. However, the higher λ_{psl} leads to instability on CIFAR-10. The possible reason is that CIFAR-20 is more diverse and has more semantic classes than CIFAR-10 (100 versus 10). Therefore, we suggest that λ_{psl} can be set to $[0.01, 0.1]$, which has demonstrated superior performance on both two datasets.

5.5.4 Projection Dimension

The projection dimension describes the embedding space of SSL. The results in Fig. 7d shows that ProPos achieves consistent and significant performance improvement over baseline regardless of different projection dimension.

5.5.5 ResNet Backbone

Fig. 7e shows that with the deeper ResNet networks, ProPos achieves significant improvements with small standard deviations for clustering, demonstrating its superior stability and performance against the baseline method.

5.5.6 Data Augmentation

Data augmentation is important for self-supervised learning. Fig. 7f shows that the performance drops for both BYOL and ProPos when removing some data augmentations. However, the results suggest that ProPos still performs more stable and is robust to data augmentations.

5.6 Computational Cost

The main additional computational cost of ProPos is the k-means clustering procedure. We have implemented the k-means algorithm with k-means++ [65] initialization using PyTorch to utilize the GPU and accelerate the clustering process. We performed k-means clustering with 10 different initialization in cosine distance. Table 7 summarizes the training time of ProPos and BYOL on different datasets. ProPos does not introduce much additional computational cost. Besides, as suggested in the results of Fig. 7a, ProPos is robust to the different r, so there is no need to perform k-means for every epoch. Therefore, the training time can be further reduced for ProPos. The computational cost of the PSL is also small since we build the cluster centers within mini-batch, saying that ProPos does not need additional memory to store the cluster centers. Consequently, considering the promising performance improvements, the additional computational cost is relatively affordable.

5.7 Subsets of ImageNet

In addition, we also reported the clustering results on ImageNet subsets like SCAN [6] in Table 8. We strictly follow the settings in [6]: we have adopted the same 50, 100, and 200 classes from ImageNet, clustered on the training set, and tested on the validation set. We have used the same experimental settings as the other benchmarked datasets and trained ProPos with ResNet-50 for 300 epochs. We note that SCAN has used the pre-trained model of MoCo trained on the full ImageNet for 800 epochs. The results are directly referred from their published paper including k-means with pre-trained MoCo, SCAN after the clustering step, and SCAN after the self-labeling step. With much fewer training epochs and training data, ProPos still produces better performance by a clear margin, demonstrating the superiority of ProPos.

5.8 Long-Tail Dataset

We also conducted additional experiments in Table 9 to demonstrate the ability of ProPos handling the long-tailed datasets. We built the long-tailed version of CIFAR-10 and CIFAR-20, termed CIFAR-10-LT and CIFAR-20-LT using the codes of [66], which follows [67], [68]. Specifically, they

TABLE 8
Clustering Results (%) on the Subsets of ImageNet

ImageNet	50 Classes		100 Classes		200 Classes	
Method	NMI	ARI	NMI	ARI	NMI	ARI
k-means with pre-trained MoCo	77.5	57.9	76.1	50.8	75.5	43.2
SCAN [6] after clustering step	80.5	63.5	78.7	54.4	75.7	44.1
SCAN [6] after self-labeling step	82.2	66.1	80.8	57.6	77.2	47.0
ProPos (**ours**)	**82.8**	**69.1**	**83.5**	**63.5**	**80.6**	**53.8**

TABLE 9
Clustering Results (%) on Long-Tailed Datasets of Different Self-Supervised Learning Frameworks and Our Proposed ProPos

Method	CIFAR-10-LT			CIFAR-20-LT		
	NMI	ACC	ARI	NMI	ACC	ARI
MoCo v2 [1]	46.7±0.1	33.4±0.3	27.7±0.0	31.2±0.3	28.2±0.2	16.1±0.3
BYOL [3]	51.6±1.0	41.3±0.4	30.8±0.4	41.9±0.4	34.6±0.5	22.3±1.0
ProPos w/o PSL	53.1±0.7	42.7±0.4	31.6±0.8	43.4±0.8	35.1±0.6	24.0±0.1
ProPos (**ours**)	**55.3**±0.4	**43.9**±0.1	**36.3**±0.3	**44.6**±0.2	**39.0**±0.7	**27.3**±0.2

were built upon the training datasets under the control of data imbalance ratio $\min(\{N_k\}_{k=1}^{K})/\max(\{N_k\}_{k=1}^{K}) = 0.1$. Consequently, the samples in the long-tailed datasets are almost all in the minority classes (head), versus few samples in other classes (tail). MoCo v2 cannot handle this problem well due to the class collision issue, as a result, the samples in the head will be pushed away and the ones in the tail will be mixed together. BYOL and ProPos do not need negative examples so that they outperform MoCo v2 by a large margin. By introducing PSA and PSL, we can further boost the clustering performance of BYOL.

5.9 Boosting Performance With Memory Queue

To better represent the prototype of one class, the mini-batch should contain a sufficient number of samples. However, this would significantly increase the requirement of GPU memory when one dataset has a large number of clusters. Here, we highlight that although we use a mini-batch size of 256, we find that ProPos generalizes well on the datasets such as Tiny-ImageNet with 200 classes (about 1 sample per class in a mini-batch) and ImageNet-1k with 1,000 classes (about 0.25 sample per class in a mini-batch).

To compute class prototypes accurately with a small mini-batch size, we propose to employ a memory queue for updating prototypes. In the vanilla PSL, we use the representations within mini-batch $\mathcal{B}$ to estimate the prototypes in Eqs. (6) and (7). With a memory queue $\mathcal{Q}$ used in [1] to store the representations from the momentum-updated encoder, we can update the prototypes with samples in both mini-batch and memory queue, $\mathcal{B} \cup \mathcal{Q}$.
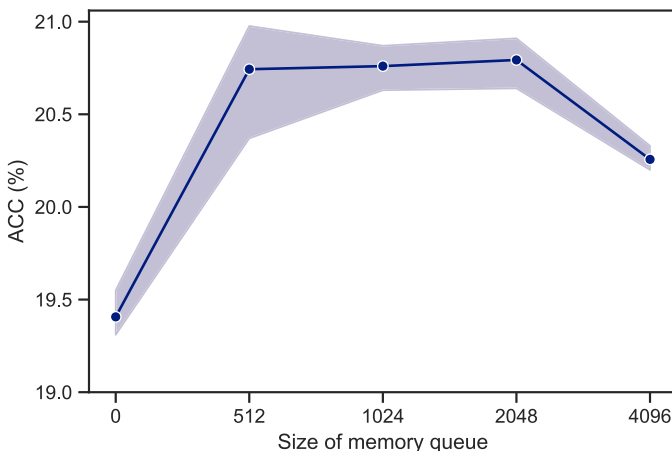

Fig. 8. The performance of ProPos using a memory queue with different sizes on Tiny-ImageNet. We repeated each run for 3 times and reported the mean and std values.

To show the effectiveness of the memory queue, we conduct the experiments on Tiny-ImageNet as it is challenging enough with 200 classes, and the corresponding results are shown in Fig. 8; we trained the model with the same settings as detailed in Section 5.1.3 except for 200 epochs. As shown in Fig. 8, the performance can be further improved with more samples used to compute the prototypes. However, the performance could drop when the size of memory queue becomes too large, saying 4,096 in Fig. 8. The reason is that when the memory queue is too large, the representations enqueued at early iterations may be far away from the true ones due to the longer update of the encoder [1]. Moreover, using the memory queue brings more computational costs and introduces an additional hyperparameter—the size of memory queue. Therefore, ProPos works well when the mini-batch size is small compared to the number of classes, and its performance can be further improved with a memory queue.

6 DISCUSSION

In this section, we discuss the differences between our method and previous works.

6.1 Relation to CC

Although both PSL and CC [8] are class-level contrastive loss, which perform contrastive learning at the cluster level, they have the following difference.

- The class-level contrastive loss in CC implements the contrastive loss on *the cluster probabilities* while ours on *the representation of cluster centers*. Implementing contrastive loss on the cluster probability in [8] would lose the semantic information of the learned representations, which is not helpful for representation learning. Specifically, given the $\boldsymbol{x} \in \mathcal{B}$, CC obtains the cluster assignments $\boldsymbol{P}_k = [p(k|\boldsymbol{x}^{(1)}), \ldots, p(k|\boldsymbol{x}^{(N)})]$ from one view and $\boldsymbol{P}_k'$ from another view, and then contrasts $\boldsymbol{P}_k$ and $\boldsymbol{P}_k'$ at the cluster level using the InfoNCE loss. In contrast, PSL implements the contrastive loss on the representation of the cluster centers within a mini-batch using the pseudo-labels from k-means clustering. As a result, PSL is able to sense the semantic information of the latent space and make the representations of clusters more discriminative and suitable for the clustering task.

- The class-level contrastive loss in CC does not encourage cluster uniformity while our PSL does. CC still needs the instance-wise contrastive loss to

encourage instance uniformity, which inevitably introduces the class collision issue.

In addition to these differences, experimentally, we also compare the PSL and CC [8] in the same BYOL framework and report the results in Table 6. The experimental results show that compared to BYOL, BYOL+CC drops the performance and makes the training unstable as CC fails to encourage uniform representations. Under the same conditions, ProPos achieves significant improvements over BYOL +CC.

6.2 Relation to GCC and WCL

To alliterate the class collision issue, GCC [41] and WCL [42] built a graph to label the neighbor samples as pseudo-positive examples. Then, they enforce the two data augmentations of one example to be close to its multiple pseudo-positive examples using a supervised contrastive loss. GCC adopted a moving-averaged memory bank for the graph-based pseudo-labeling while WCL built the graph within a mini-batch. GCC and WCL mainly focus on how to effectively select positive examples from mini-batch/memory bank to alleviate the class collision issue. Here, we divide the class collision issue into the following two cases:

- Negative class collision issue: negative examples may not be *truly negative*, which is the case contrastive learning faces.
- Positive class collision issue: positive examples may not be *truly positive*, which is a new case raised in GCC and WCL.

Consequently, they still suffer from the positive class collision issue as the selected pseudo-positive examples may not be truly positive. In addition to this, they also suffer from the negative class collision issue since they still need negative examples for instance-wise contrastive learning.

We summarize the difference between our PSA and theirs in the following four aspects.

- GCC and WCL select the examples that exist in the dataset (mini-batch/memory bank) while ours samples examples from the latent space that may not exist in the dataset.
- GCC and WCL select neighbor examples in a graph as pseudo-positive examples that may not be *truly positive* while ours samples examples around the instance in the embedding space that can be assumed to be *truly positive*.
- GCC and WCL still rely on instance-wise contrastive loss that could lead to class collision issue while ours can avoid class collision issue by using non-contrastive BYOL.
- GCC and WCL require additional computational cost for graph construction while ours is rather cheap in sampling one example in the embedding space.

6.3 Relation to PCL

Here, we summarize the difference between our ProPos and PCL [7] in terms of the losses and EM frameworks. First, we summarize the difference between our PSL and ProtoNCE loss used in PCL as follows.

- Our ProPos can avoid class collision issue while PCL cannot. ProPos is based on BYOL that does not require negative examples for representation learning while PCL is based on instance-wise contrastive loss that requires a number of negative examples for representation learning, inevitably leading to class collision issue.
- The proposed PSL in ProPos is conceptually different from the ProtoNCE in PCL. PSL is to maximize the inter-cluster distance to form a uniformly distributed space while ProtoNCE is to minimize the instance-to-cluster distance to improve the within-cluster compactness. The within-cluster compactness of ProPos is improved by the proposed positive sampling alignment.
- Pure PSL can work well for deep clustering while ProtoNCE requires another InfoNCE to form uniformly distributed space. This is a direct result of the different designs of the losses. PSL can maximize the inter-cluster distance to form a uniformly distributed space while ProtoNCE suffers from collapse without the help of another InfoNCE to form such a space.

Second, we summarize the difference between our ProPos and PCL in the EM framework. Formulating ProPos into an EM framework can offer more insights about ProPos and make it easy to understand. Although both in an EM framework, the M-step in PCL is significantly different from the one in our ProPos. More specifically, the M-step in PCL is to optimize the ProtoNCE, which is an *instance-to-prototypes contrastive loss* to improve the within-cluster compactness while the M-step in our ProPos is to optimize the proposed PSL, which is a *prototypes-to-prototypes contrastive loss* to maximize the inter-cluster distance for better clustering performance. In addition, ProPos also proposes a positive sampling alignment by sampling positive examples around each sample to improve within-cluster compactness.

6.4 Relation to Instance-Reweighted Contrastive Loss

We provide a new perspective to understand the proposed cluster-wise PSL from instance-reweighted contrastive loss [45]. Here, we first focus on analyzing the alignment term of PSL. By substituting $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}'_k$ into the prototypical alignment term of PSL, we can rewrite the alignment term as:

$$\frac{1}{K}\sum_{k=1}^{K} -\frac{\boldsymbol{\mu}_k^T \boldsymbol{\mu}'_k}{\tau} \tag{13}$$

$$= \frac{1}{K}\sum_{k=1}^{K} -\frac{1}{\tau}\frac{\sum_{\boldsymbol{x}\in\mathcal{B}} p(k|\boldsymbol{x})f(\boldsymbol{x})^T}{c_k}\frac{\sum_{\boldsymbol{x}\in\mathcal{B}} p(k|\boldsymbol{x})f'(\boldsymbol{x})}{c'_k} \tag{14}$$

$$= \frac{1}{K}\sum_{k=1}^{K} -\frac{1}{\tau}\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} p(k|\boldsymbol{x}_i)p(k|\boldsymbol{x}_j)f(\boldsymbol{x}_i)^T f'(\boldsymbol{x}_j)}{c_k c_k'} \tag{15}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} -\underbrace{\frac{1}{K}\sum_{k=1}^{K}\frac{p(k|\boldsymbol{x}_i)p(k|\boldsymbol{x}_j)}{c_k c'_k}}_{\boldsymbol{w}_{ij}}\frac{f(\boldsymbol{x}_i)^T f'(\boldsymbol{x}_j)}{\tau}, \tag{16}$$

where $p(k|\boldsymbol{x}) \in \{0,1\}$, $c_k = \|\sum_{\boldsymbol{x}\in\mathcal{B}} p(k|\boldsymbol{x})f(\boldsymbol{x})^T\|_2$, $c'_k = \|\sum_{\boldsymbol{x}\in\mathcal{B}} p(k|\boldsymbol{x})f'(\boldsymbol{x})\|_2$, and $\boldsymbol{w} = \{\boldsymbol{w}_{ij}\}_{i,j=1}^{N} \in \mathbb{R}^{N\times N}$ denotes

the weights of each instance pair. Eq. (16) shows that the alignment term in PSL can be formulated as an instance-reweighted contrastive loss.

From Eq. (16), we have the following observation:

- When x_i and x_j belong to the same cluster, i.e. $p(k|x_i) = p(k|x_j) = 1$, we have $w_{ij} > 0$.
- When x_i and x_j belong to the different clusters, we have $w_{ij} = 0$.

As a result, the alignment term in PSL only contains the sample pairs belonging to the same clusters, which is similar to supervised contrastive loss [23]. Therefore, the alignment term of PSL is a generalized case of instance-reweighted contrastive loss that takes into account the pseudo-labels. Similarly, one can observe that the uniformity term in PSL is to maximize the distance between instances in different clusters ($j \neq k$).

Therefore, we can understand the proposed PSL from a perspective of instance-reweighted contrastive loss with cluster labels taken into account.

7 CONCLUSION

We introduced a novel deep clustering method ProPos, which enjoys the strengths of both contrastive- and non-contrastive-based methods. The proposed positive sampling alignment and prototype scattering loss can lead to within-cluster compactness and well-separated clusters. The results empirically showed that the proposed ProPos outperforms the state-of-the-art methods by a significant margin. Current state-of-the-art methods are mostly beneficial from the progress of self-supervised representation learning while the new trends such as MAE [27] have presented more superior performance on downstream tasks, which deserves studying as future work.

REFERENCES

[1] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[3] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, Art. no. 1786.

[4] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 750–15 758.

[5] X. Wang, Z. Liu, and S. X. Yu, "Unsupervised feature learning by cross-level instance-group discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12 586–12 595.

[6] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "SCAN: Learning to Classify Images Without Labels," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 268–285.

[7] J. Li, P. Zhou, C. Xiong, and S. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[8] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 8547–8555.

[9] Y. Tao, K. Takagi, and K. Nakata, "Clustering-friendly representation learning via instance discrimination and feature decorrelation," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[10] T. W. Tsai, C. Li, and J. Zhu, "MiCE: Mixture of contrastive experts for unsupervised image clustering," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[11] C. Niu, H. Shan, and G. Wang, "SPICE: Semantic pseudo-labeling for image clustering," 2021, *arXiv:2103.09382*.

[12] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9929–9939.

[13] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, "A theoretical analysis of contrastive unsupervised representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 9904–9923.

[14] C. Zhang, K. Zhang, C. Zhang, T. X. Pham, C. D. Yoo, and I. S. Kweon, "How does SimSiam avoid collapse without negative samples? A unified understanding with self-supervised contrastive learning," in *Proc. Int. Conf. Learn. Represent.*, 2022.

[15] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[16] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 946.

[17] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1422–1430.

[18] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.

[19] R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 649–666.

[20] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[21] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.

[22] N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar, "A theoretical analysis of contrastive unsupervised representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5628–5637.

[23] P. Khosla et al., "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 18661–18673.

[24] Q. Hu, X. Wang, W. Hu, and G.-J. Qi, "AdCo: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1074–1083.

[25] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, "Debiased contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 8765–8775.

[26] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9912–9924.

[27] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021, *arXiv:2111.06377*.

[28] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[30] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[31] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5879–5887.

[32] J. Chang, G. Meng, L. Wang, S. Xiang, and C. Pan, "Deep self-evolution clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 809–823, Apr. 2018.

[33] P. Haeusser, J. Plapp, V. Golkov, E. Aljalbout, and D. Cremers, "Associative deep clustering: Training a classification network with no labels," in *Proc. German Conf. Pattern Recognit.*, 2018, pp. 18–32.

[34] J. Chang, L. Wang, G. Meng, Q. Zhang, S. Xiang, and C. Pan, "Local-aggregation graph networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2874–2886, Nov. 2020.

[35] J. Wu et al., "Deep comprehensive correlation mining for image clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8150–8159.

[36] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9865–9874.

[37] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman, "AutoNovel: Automatically discovering and learning novel visual categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6767–6781, Oct. 2022.

[38] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng, "Twin contrastive learning for online clustering," *Int. J. Comput. Vis.*, vol. 130, pp. 2205–2221, 2022.

[39] Y. Shen, Z. Shen, M. Wang, J. Qin, P. Torr, and L. Shao, "You never cluster alone," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 27734–27746.

[40] Y. Guo et al., "HCSC: Hierarchical contrastive selective coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9696–9705.

[41] H. Zhong et al., "Graph contrastive clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9204–9213.

[42] M. Zheng et al., "Weakly supervised contrastive learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 10022–10031.

[43] J. Lee, H. Lee, I. Shin, J. Bae, I. S. Kweon, and J. Choo, "Learning representations by contrasting clusters while bootstrapping instances," 2020. [Online]. Available: https://openreview.net/forum?id=MRQJmsNPp8E

[44] J. R. Regatti, A. A. Deshmukh, E. Manavoglu, and U. Dogan, "Consensus clustering with unsupervised representation learning," in *Proc. IEEE Int. Conf. Neural Netw.*, 2021, pp. 1–9.

[45] J. Mitrovic, B. McWilliams, J. Walker, L. Buesing, and C. Blundell, "Representation learning via invariant causal mechanisms," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[46] A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[47] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014.

[48] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, Canada, 2009.

[49] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, 2011, pp. 215–223.

[50] Y. Le and X. Yang, "Tiny ImageNet visual recognition challenge," *CS 231N*, vol. 7, no. 7, 2015, Art. no. 3.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[52] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[53] C. Niu, J. Zhang, G. Wang, and J. Liang, "GATCluster: Self-supervised Gaussian-attention network for image clustering," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 735–751.

[54] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10268–10278.

[55] J. Huang, S. Gong, and X. Zhu, "Deep semantic clustering by partition confidence maximisation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8849–8858.

[56] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang, "Nearest neighbor matching for deep clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13688–13697.

[57] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[58] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[59] C. Zhuang, A. L. Zhai, and D. Yamins, "Local aggregation for unsupervised learning of visual embeddings," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6002–6012.

[60] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 776–794.

[61] T.-S. Chen, W.-C. Hung, H.-Y. Tseng, S.-Y. Chien, and M.-H. Yang, "Incremental false negative detection for contrastive learning," in *Proc. Int. Conf. Learn. Represent.*, 2022.

[62] A. Fetterman and J. Albrecht, "Understanding self-supervised and contrastive learning with bootstrap your own latent (BYOL)," 2020. [Online]. Available: https://untitled-ai.github.io/understanding-self-supervised-contrastive-learning.html

[63] Y. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[64] I. Misra and L. V. D. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6707–6717.

[65] D. Arthur and S. Vassilvitskii, "k-means: The advantages of careful seeding," in *Proc. 8th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2006, pp. 1027–1035.

[66] K. Tang, J. Huang, and H. Zhang, "Long-tailed classification by keeping the good and removing the bad momentum causal effect," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1513–1524.

[67] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9719–9728.

[68] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 140.

Zhizhong Huang (Graduate Student Member, IEEE) received the BS degree from the Sichuan University, Chengdu, China, in 2019. He is currently working toward the PhD degree with the School of Computer Science, Fudan University, Shanghai, China. His research interests include machine/deep learning, computer vision, face aging, and image translation.

Jie Chen received the BS degree from Xiamen University, Xiamen, China, in 2017. He is currently working toward the PhD degree with the School of Computer Science, Fudan University, Shanghai, China. His research interests include machine/deep learning, graph neural networks, and causality.

Junping Zhang (Senior Member, IEEE) received the BS degree in automation from Xiangtan University, China, in 1992, the MS degree in control theory and control engineering from Hunan University, Changsha, China, in 2000, and the PhD degree in intelligent systems and pattern recognition from the Institution of Automation, Chinese Academy of Sciences, in 2003. He has been a professor with the School of Computer Science, Fudan University, since 2006. His research interests include machine learning, image processing, biometric authentication, and intelligent transportation systems. He has been an associate editor of *IEEE Intelligent Systems* since 2009 and was an associate editor of *IEEE Transactions on Intelligent Transportation Systems* (2010-2018).
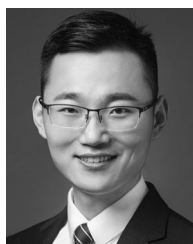
Hongming Shan (Senior Member, IEEE) received the PhD degree in machine learning from Fudan University, in 2017. From 2017 to 2020, he was a postdoctoral research associate and research scientist with Rensselaer Polytechnic Institute. He is currently an associate professor with the Institute of Science 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. His research interests include machine learning, medical imaging, and image processing. He was recognized with Youth Outstanding Paper Award at World Artificial Intelligence Conference 2021.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.