Siamese Prototypical Contrastive Learning

Shentong Mo[1]
shentonm@andrew.cmu.edu
Zhun Sun*[2]
sun@vision.is.tohoku.ac.jp
Chao Li[3]
chao.li@riken.jp

[1] Carnegie Mellon University
Pittsburgh, PA 15213, United States
[2] Tohoku University
Sendai, Miyagi, Japan
[3] Center for Advanced Intelligence
Project (AIP), RIKEN
Tokyo, Japan

Abstract

Contrastive Self-supervised Learning (CSL) is a practical solution that learns meaningful visual representations from massive data in an unsupervised approach. The ordinary CSL embeds the features extracted from neural networks onto specific topological structures. During the training progress, the contrastive loss draws the different views of the same input together while pushing the embeddings from different inputs apart. One of the drawbacks of CSL is that the loss term requires a large number of negative samples to provide better mutual information bound ideally. However, increasing the number of negative samples by larger running batch size also enhances the effects of false negatives: semantically similar samples are pushed apart from the anchor, hence downgrading downstream performance. In this paper, we tackle this problem by introducing a simple but effective contrastive learning framework. The key insight is to employ siamese-style metric loss to match intra-prototype features, while increasing the distance between inter-prototype features. We conduct extensive experiments on various benchmarks where the results demonstrate the effectiveness of our method on improving the quality of visual representations. Specifically, our unsupervised pre-trained ResNet-50 with a linear probe, out-performs the fully-supervised trained version on the ImageNet-1K dataset.

1 Introduction

Massive studies have recently shown that Contrastive Self-supervised Learning (CSL) is a powerful tool for unsupervised pre-training models and semi-supervised training methods [4, 5, 6, 7, 10, 16, 24]. The core idea of CSL is to utilize the views of samples to construct a discrimination pretext task. This task learns the capability of the deep neural network extracting meaningful feature representations, which can be further used by tons of downstream tasks, such as image classification, object detection, and instance segmentation.

This idea of CSL has appeared in former literature a long time ago [3, 12, 13, 15, 32]. The term "view" in CSL generally refers to augmented samples, and the goal of CSL is to close the distance between different views from the same sample, *a.k.a.* the anchor, while keeping views from different samples away from the anchor. This goal is commonly achieved in recent studies by employing the InfoNCE [28] loss and its variants. The ideal outcome of a contrastive pre-trained model should distribute the samples in the embedding space uniformly by maximizing the mutual information between the anchor and its views.

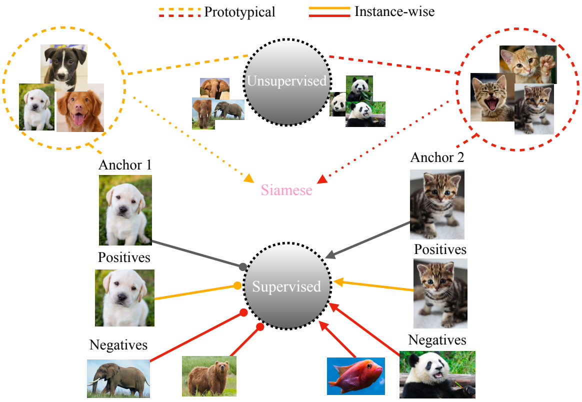


Figure 1: Illustration of our Siamese Prototypical Contrastive Learning. The embedded features are grouped into separate prototypes, and a Siamese-style metric loss is employed to match intra-prototype while pushing apart inter-prototype features. Instance-wise contrastive learning is conducted inside each separate prototype. And the views of the anchors are regularized to the corresponding prototypes.

However, theoretically, it is demonstrated that the loss term requires a large number of negative samples to provide better mutual information bound [30, 35], which leads to large batch sizes. Although the growth of computation resources nowadays can support training with batch sizes of dozens of thousands, the large mini-batch often results in new trouble: The chance of semantically similar samples being computed as negative ones is raised. To push apart the *false negative* pairs in the embedding space does not help to build a semantic understanding of the samples, and does actually damage the performance of the pre-trained model when the downstream tasks heavily rely on the semantic information [1, 35].

One straightforward approach to address it is to employ the label information of the samples explicitly, *i.e.* supervised contrastive learning [20]. The samples are regularized to the corresponding semantic clusters with an ordinary cross-entropy loss. The obvious shortcoming of this approach is the prerequisite of the downstream tasks knowledge. Another approach is to use "prototypes" to tackle the issue in a task-agnostic manner. The term “prototype” refers to the clusters of the samples in the embedding space. The clusters can be either pre-defined or computed during training mini-batches. The samples from the same prototype contain toughly similar semantic information. Thus, the learned embedded features provide better discriminability for the downstream classification tasks.

In this work, we follow the instance-wise contrastive learning but also regularize the views of samples with supervised prototypes. In our initial experiments, we find that a straightforward addition of the prototypes’ cross-entropy loss decreases the performance. Moreover, it is not helpful to separate the cluster centers by minimizing their similarity. We conjecture that it is not feasible to distinguish the semantic false-negative samples in the original embedding space. To address this problem, we propose a Siamese-style metric loss [22] to maximize the inter-prototype distance while preserving the intra-prototype similarity.

From our experimental analysis, we empirically reveal that the similarity of the false-negative samples to their anchors decreases in the contrastive projection space. We present

[*] Corresponding author.

the overall framework as Siamese Prototypical Contrastive Learning, the main contributions of this paper can be summarized as follows:

- We propose a simple but effective contrastive learning framework, utilizing the Siamese-style metric loss to improve the semantic discriminability between prototypes.
- We show that our SPCL successfully mitigates the confusion brought by false negatives.
- Our SPCL outperforms its baseline by a large margin and achieves state-of-the-art performances on the ImageNet-1K dataset.

2 Related Works

2.1 Self-supervised Learning

In recent years, self-supervised learning methods have gained popularity in learning visual representations. DeepCluster [4] performs iterative clustering and unsupervised representation learning, which is further improved by DeeperCluster [5] to learn transferable representations from a large scale of images and online clustering [44]. In this work, we intend to employ clustering to boost the performance of contrastive self-supervised learning of visual representations. The objective functions used to train visual representations in many methods are either reconstruction-based loss functions or contrastive loss that measures the co-occurrence of multiple views [6, 7, 8, 10, 16, 34]. The effectiveness of self-supervised learning in visual representation can be attributed to two important aspects, namely pretext tasks and contrastive learning. For pretext tasks, there has been a wide range of pretext tasks explored to learn a good representation. These examples include colorization [41, 45], context autoencoders [29], spatial jigsaw puzzles [25, 27] and discriminate orientation [40]. For contrastive learning, that approaches [14] bring the representation of different views of the same image (positive pairs) closer, and push representations of views from different images (negative pairs) apart using instance-level classification with contrastive loss.

2.2 Contrastive Learning

Recently, researchers have been interested in exploring effective methods using contrastive self-supervised learning. The breakthrough approach is SimCLR [7], which is trained by an end-to-end structure, where the features of each instance are pulled away from those of all other instances in the training set. In-variances are encoded from low-level image transformations such as cropping, scaling, and color jittering. This sort of end-to-end structures [7, 8] always use large batch size to accumulate a large bunch of negative samples in a contrastive loss. Instead of using a large batch size, PIRL [25] applies a memory bank to store and update representations learned at a specified stage, which would be computationally expensive. To update the stored memory efficiently, MoCo [16] and MoCov2 [10] replace the memory bank with a memory encoder to queue new batch samples and to dequeue the oldest batch. A concurrent work [9] explores a simple Siamese network to maximize the similarity between two views of the anchor. DenseCL [37] and SCRL [31] apply contrastive learning on the pixel-level and spatial-level features to boost the performance of tasks related to images.

2.3 Prototypical Contrastive Learning

Compared to instance-wise contrastive methods, relatively few works consider using prototypical contrastive methods. More recent work, SwAV [6] focuses on using a "swapped" clustering-based mechanism to group similar representations, which enforce consistency between cluster assignments produced for different views of the anchor. PCL [24] uses prototypes as latent variables and performs iterative clustering in an expectation-maximization

Algorithm 1 SPCL main learning algorithm

```
Input: Data $X$, number of prototypes $N$, $f, g_c, g_m, g_p$, sets of augmentation $\mathcal{T}$.
for each epoch do
    Re-initialize $g_c$, $g_m$, $g_p$.
    Obtain clusters of prototypes $\boldsymbol{C} = \{\boldsymbol{c}^0, ..., \boldsymbol{c}^n\}$ using $K$-means clustering.
    for each step do
        Choose the prototype $p, q$.
        Sample $\boldsymbol{x}^p$, $\boldsymbol{x}^q$ from the corresponding clusters $\boldsymbol{c}^p$, $\boldsymbol{c}^q$.
        Obtain $\boldsymbol{h}^p$, $\boldsymbol{h}^q$ from the inputs and their augmented views.
        Compute the $L_{contra}, L_{metric}, L_{proto}$ using Eq. 2, 3, 4.
        Compute $L_{total}$ in Eq. 5, and update networks $f, g_c, g_m, g_p$ to minimize $L_{total}$
```

(EM)-based framework using the proposed ProtoNCE loss. In this work, we employ clustering as an unsupervised learning method to generate prototypes as a “pseudo label”. Furthermore, we propose the siamese-style metric loss to match the intra-class features and increase the distance between inter-class features. Our “metric” loss is similar to the supervised contrastive loss in SupCon [20]. However, we do not use the ground-truth directly as supervised information and instead leverage the clustering to generate “pseudo label”.

3 Method

We first illustrate the main framework of our proposed Siamese-style Prototypical Contrastive Learning (SPCL) and then provide technical details in the following subsections.

3.1 Overview and Notation

Overview. Our goal is to learn meaningful visual representations in an unsupervised and task-agnostic style. To achieve this, we present a simple but effective variant of contrastive self-supervised learning. The fundamental idea of this method is to group the samples in the embedding space to produce prototypes; then, we refine the prototypes by maximizing/minimizing the inter-/intra-cluster distance between samples. Finally, we regularize the views of samples using the prototypical information with cross-entropy.

Our SPCL contains four major modules: the start-epoch clustering, the ordinary contrastive loss, the Siamese-style metric loss, and the prototypical cross-entropy loss, which we will introduce below. Fig. 1 and Algo. 1 provide a rough illustration of our method.

Notation. We use italic letters to denote scalars, *e.g.*, $n, N \in \mathbb{R}$, and use boldface letters to denote vectors/matrices/tensors, *e.g.*, $\boldsymbol{h}, \boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{R}^{h \times w \times c}$. We denote *sets* by calligraphic letters, *e.g.* $\mathcal{X}, \mathcal{T}$ stands for the set of samples and transformations, correspondingly. Sometimes italic letters are also used to represent functions or procedures, we employ the form $f(\cdot)$ to exclude ambiguity.

3.2 Clustering and Feature Extracting

Clustering. The first step of our SPCL is to group the embedded features into separate clusters in an unsupervised manner. This is achieved by the simple $K-$means algorithm, where the number of prototypes K is a preset hyper-parameter. Concretely, at the beginning of every epoch, we split the dataset $\mathcal{X}$ into mini-batches. For each mini-batch, we draw an augmentation $t \sim \mathcal{T}$ and extract the features of the augmented samples. We concatenate all the embedded features and conduct the clustering algorithm to obtain the set of prototypes $\mathcal{C} = \boldsymbol{c}^1, \ldots, \boldsymbol{c}^K$, where $\boldsymbol{c}^k$ is the vector of index recording samples of the cluster k.

Feature Extracting. During the feature extracting phase, we first draw an anchor prototype p and sample a mini-batch $\mathcal{X}^p = \{\boldsymbol{x}_n^p\}_{n=1}^N$ from $\boldsymbol{c}^p$. Then, we draw samples that do not belong to prototype p, *i.e.* from $\mathcal{C} \setminus \{\boldsymbol{c}^p\}$. and denote this mini-batch with $\mathcal{X}^q = \{\boldsymbol{x}_n^q\}_{n=1}^N$, where q stands for any prototypes that are not p. For each sample inn $\mathcal{X}^p$ and $\mathcal{X}^q$, we acquire its *two* augmented views and the feature representations in the embedding space. The feature representations are denoted by $\mathcal{H}^p = \{\boldsymbol{h}_{n,1}^p, \boldsymbol{h}_{n,2}^p\}_{n=1}^N, \mathcal{H}^q = \{\boldsymbol{h}_{n,1}^q, \boldsymbol{h}_{n,2}^q\}_{n=1}^N$, with all their elements $\boldsymbol{h} \in \mathbb{R}^{d_e}$, where d_e is the dimension of the embedding space.

3.3 Contrastive loss

We employ the ordinary **NT-Xent** (the normalized temperature-scaled cross entropy loss) [28, 33, 39] as our contrastive loss. Following SimCLR [7], we use a projection head $g_c(\cdot)$ that maps the extracted representations $\boldsymbol{h}$ to the space where contrastive loss is applied. That is, $\boldsymbol{z} = g_c(\boldsymbol{h})$. We use $s(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{v} / ||\boldsymbol{u}|| ||\boldsymbol{v}||$ to denote the dot product between ℓ_2 normalized $\boldsymbol{u}$ and $\boldsymbol{v}$ (*i.e.* cosine similarity). Then the contrastive loss of a sample $\boldsymbol{z}$ is defined as

$$L_{contra,\boldsymbol{z}} = -\log \frac{\exp(s(\boldsymbol{z}, \boldsymbol{z}')/\tau)}{\sum\limits_{\substack{\bar{\boldsymbol{z}} \in \{\mathcal{Z}^p \cup \mathcal{Z}^q\}, \\ \bar{\boldsymbol{z}} \neq \boldsymbol{z}, \boldsymbol{z}'}} \exp(s(\boldsymbol{z}, \bar{\boldsymbol{z}})/\tau))}, \tag{1}$$

where $\mathcal{Z}$ is the set of representations in the projection space, $\boldsymbol{z}'$ is the representation of the other view of the sample, τ denotes a temperature parameter. The final contrastive loss is computed as the sum-reduce across all samples:

$$L_{contra} = \sum_{\boldsymbol{z} \in \{\mathcal{Z}^p \cup \mathcal{Z}^q\}} L_{contra,\boldsymbol{z}} \tag{2}$$

3.4 Siamese-style metric loss

In our initial experiments, we observe that the prototypical cross-entropy loss employment does not improve the performance of the downstream tasks. One reason is that the clusters of prototypes are not well separated; the samples that are far away from the cluster centers might be *false positive* and are aggressively pulled towards the anchor sample. As a consequence, the semantic information of the prototypes becomes more obscure. To tackle this problem, we introduce a Siamese-style metric loss, which matches intra-class features and increases the distance between inter-class features. Concretely, for each feature representation of the anchor prototype p, we draw two feature embeddings from p and q as positive and negative pairs, denoted as $\{\boldsymbol{h}^p, \hat{\boldsymbol{h}}^p\}$ and $\{\boldsymbol{h}^p, \hat{\boldsymbol{h}}^q\}$. It is worth mentioning that, here, we still slightly abuse the notation q, which stands for *a prototype that is not p*. With a projection head $g_m(\cdot)$ that predicts the distance between the pairs, the Siamese-style metric loss can be written as

$$L_{metric} = \sum_{\boldsymbol{h}^p \in \mathcal{H}^p} \left(\mathtt{CE}(g_m(|\boldsymbol{h}^p - \hat{\boldsymbol{h}}^p||, 1) + \mathtt{CE}(g_m(|\boldsymbol{h}^p - \hat{\boldsymbol{h}}^q||, 0) \right) \tag{3}$$

3.5 Prototypical cross-entropy loss

Motivated by SupCon [20], we introduce the prototypical cross-entropy loss in our SPCL to guide the views of one sample to its corresponding prototype. This loss term is helpful when the similarity between views is small, *i.e.* $s(\boldsymbol{z}, \boldsymbol{z}') \approx 0$, and the contrastive loss term only produces gradients with trivial magnitude. We use the notation $\mathtt{CE}(\cdot, \boldsymbol{y})$ to denote the sparse cross-entropy loss with a softmax attached, where $\boldsymbol{y}$ standards for the one-hot encoding label of the sample. Then the prototypical cross-entropy loss is computed as

$$L_{proto} = \sum_{\boldsymbol{h}\in\{\mathcal{H}^p\cup\mathcal{H}^q\}} \mathtt{CE}(g_p(\boldsymbol{h}), \boldsymbol{y}) \tag{4}$$

where $g_p(\cdot)$ is a linear projection head that maps the representations to the label.

Similar ideas have been explored in proxy learning works [21, 26]. However, our SPCL is distinct from them in terms of both the general motivation and the detailed implementation. 1) Generally, they take the advantage of the proxy based loss for learning fine-grained semantic relations between data points to boost the speed of convergence. However, we utilize the Siamese-style metric loss to mitigate the semantic confusion of prototypes between false positives and the anchor. 2) In details, they implement the Log-Sum-Exp as the max function to pull the anchor sample and most dissimiliar positive samples together and to push its most similar negative samples apart. In this work, we directly apply a prototypical cross entropy loss between the softmax output and the proxy prototype to guide the positive sample towards its prototypes with most similiar semantic relations. Our prototypical cross-entropy loss is much simpler than the Proxy-Anchor Loss, the variant of Proxy-NCA [26].

3.6 Overall Loss

The overall loss is simply computed as the weighted summation of the three losses:

$$L_{total} = \alpha L_{contra} + \beta L_{metric} + \gamma L_{proto} \tag{5}$$

We carry out a series of ablation studies in Section 4.3.2 to explore the effects of each loss.

4 Experimental Results

4.1 Dataset and Configurations

We employ 3 benchmark datasets for evaluating the classification performance: CIFAR-10/100 [23] and ImageNet-1K [11]. For CIFAR-10/100 pre-training, we closely follow SimCLR [7] and use the same data augmentation. For the encoder network ($f(\cdot)$) we experiment with the commonly used encoder architectures, ResNet-50. As the optimizer, we use LARS [42] with learning rate of 1.0 and weight decay of 10^{-6}. We use linear warmup for the first 10 epochs, and decay the learning rate with the cosine decay schedule. We train at batch size 512 for 1000 epochs, using 512 prototypes for clustering.*

For ImageNet-1K, following [10, 37], we train it for 200 epochs, using LARS with an initial learning rate of 1.0 and weight decay of 10^{-6}. We train at a batch size of 1024, and prototypes of 2048 for clustering. For clustering, we adopt the faiss library [18] for efficient k-means clustering, where it takes 6.5 seconds per epoch during the pre-training. The clustering only takes less than $1,300$ seconds in total, which is fairly negligible compared to the whole training time (482 hours using 8 Tesla V100 GPUs).

4.2 Comparison with State-of-the-arts

In Table 1, we report the architecture, parameters, batch size, and top-1 accuracy of image classification on ImageNet-1K, where the MLP stands for the projection head. Since we employ g_c, g_m, g_p for different proposals in SPCL, we denote it with $MLP_{\times 3}$. It can be clearly seen that, our SPCL outperforms all the state-of-the-art methods CSL methods, except the SupCon which used the label information for end-to-end training. Especially, our method achieves $+0.53\%$ top-1 accuracy compared to the supervised model trained with the standard CE loss, shows better generalization in representation learning and semantic understanding. It is also worth mentioning that, we do not introduce a extreme large batch size setting

[*] Detailed configurations of these experiments are provided in the supplementary materials.

Table 1: Comparison results of linear classification on ImageNet-1K.

Method	Arch.	Param.(M)	Batch	Top-1(%)
Instance-wise Contrastive Methods:				
CPC [28]	ResNet-101	28	2048	48.70
MoCo [16]	ResNet-50	24	256	60.60
PIRL [25]	ResNet-50	24	1024	63.60
CMC [34]	ResNet-50+MLP$\{L, ab\}$	47	128	64.00
CPCv2 [17]	ResNet-170	303	512	65.90
AMDIM [2]	Custom-ResNet	192	1008	68.10
CMC [34]	ResNet-50 (2×)+MLP	192	128	68.40
MoCo [16]	ResNet-50 (4×)	375	256	68.60
	ResNet-50+MLP	28	4096	70.00
SimCLR [7]	ResNet-50 (2×)+MLP	98	4096	74.20
	ResNet-50 (4×)+MLP	379	4096	76.50
MoCHi [19]	ResNet-50+MLP	28	512	70.60
MoCov2 [10]	ResNet-50+MLP	28	256	71.10
SimSiam [9]	ResNet-50+MLP	28	256	71.30
SimCLRv2 [8]	ResNet-50+MLP	28	4096	71.70
BOYL [14]	ResNet-50+MLP	35	4096	74.30
Prototypical Contrastive Methods:				
PCL [24]	ResNet-50	24	256	61.50
PCLv2 [24]	ResNet-50+MLP	28	256	67.60
SwAV [6]	ResNet-50+MLP	28	4096	75.30
SPCL (ours)	ResNet-50+MLP$_{\times 3}$	36	1024	**77.68**
Supervised Methods:				
Supervised(CE)	ResNet-50	24	256	77.15
SupCon [20]	ResNet-50+MLP	28	6144	<u>78.70</u>

as done in SimCLRv2, BOYL, and SwAV, nor have no dictionary/queue mechanism for tracking negative samples.

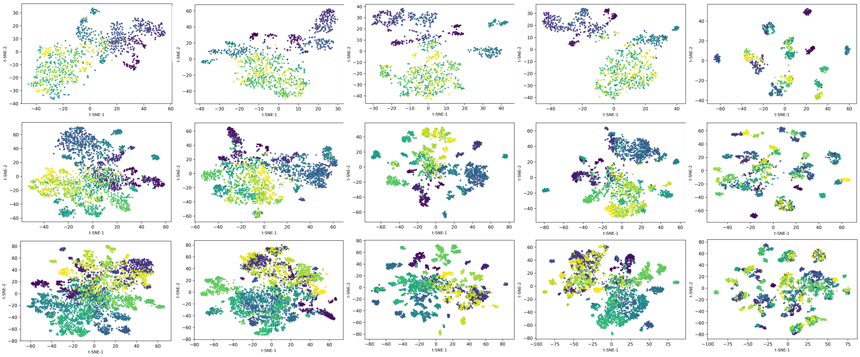


Figure 2: Visualizations of SimCLR, SwAV, PCL, DenseCL, and SPCL (**Left** to **Right**) pre-trained representations of $20, 50, 100$ (**Top** to **Bottom**) classes from ImageNet-1K.

Visualization of Feature Distributions. For comprehensive comparison with SimCLR [7], SwAV [6], PCL [24], and DenseCL [37], we evaluate the quality of representations pre-trained by those methods. Specifically, we implement the ResNet-50 pre-trained models on the $20, 50, 100$ classes randomly chosen from the ImageNet validation set. Then we project representations of 2048-dimension onto the 2-dimension space using t-SNE [36], as shown in Figure 2. We can observe that the SPCL pre-trained representations are distributed more uniformly on the space in terms of $20, 50, 100$ classes. This means that each cluster is clustered more inside, and different clusters are scattered more globally in the embedding space. Especially, the distance between each cluster is much larger than SwAV and PCL, although these two methods also utilize prototypes to perform cluster-level contrastive learning. In this case, larger distances between each separate prototype will benefit downstream tasks involved with classification.

We also compare our SPCL with non-contrastive learning approaches including BYOL [14], SimSiam [9], and BarlowTwins [43] in Figure 3. We can observe that non-contrastive learning approach shows more clustered distributions of representations pre-trained on ImageNet-1K, compared to the aforementioned contrastive learning methods. Besides, the SPCL pre-trained representations are still distributed more uniformly on the space in terms of 20,50,100 classes. That is, the intra-cluster distance is smaller and the inter-cluster distance is larger. In this way, our SPCL pre-trained representations are much suitable for classification tasks. See more experimental results in the supplementary material.

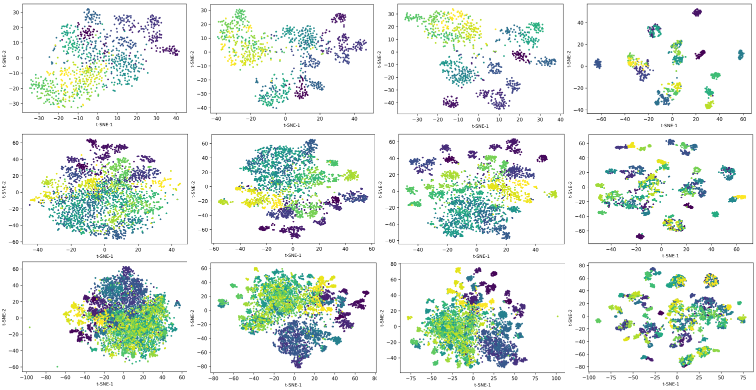


Figure 3: Visualizations of BYOL, SimSiam, BarlowTwins, and SPCL (**Left** to **Right**) pre-trained representations of $20, 50, 100$ (**Top** to **Bottom**) classes from ImageNet-1K.

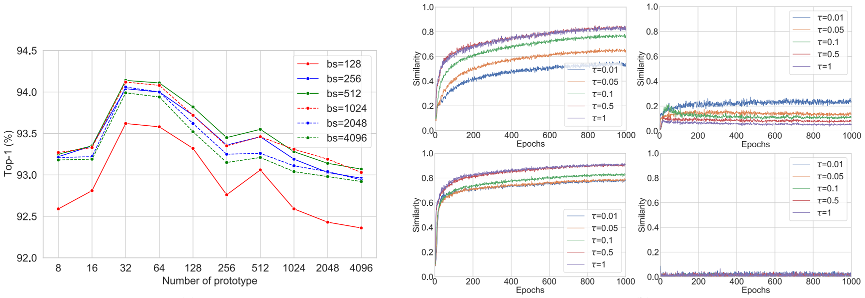
4.3 Ablation Study

4.3.1 Batch Size and Number of Prototypes

In this section, we study the effects of the *batch size* and *the number of prototypes*. Unless specified, we perform all ablation studies on the CIFAR-10 dataset. The batch size is one of the most important hyper-parameters in contrastive learning, former methods either unitize a genuine large batch size [8, 14] or mechanisms that could simulate large batch sizes [9, 10]. Here we also examine how the batch size affects the performance of our proposed method. The number of prototypes, on the other hand, is also crucial in the prototypical approaches [6, 24]. We modify the number of prototypes in a wide range to investigate the robustness of our proposed method. Furthermore, the number of prototypes is not strictly specified, and we may stack multiple prototypical cross-entropy losses using different numbers of prototypes as in [24]. We also conduct experiments employing this setting.

Specifically, we vary the number of prototypes from 8 to 4096 and the batch size from 128 to 4096 to pre-train our SPCL on CIFAR-10 for linear evaluation. The experimental results are shown in Figure 4 (a). From Figure 4 (a), we can observe that the top-1 accuracy of linear evaluation does not have much change along with the number of the batch size, when it is sufficient (larger than 128 in this case). We conjecture that the regularization from the prototypical loss helps eliminates the demand for the large number of negative samples.

On the other hand, the accuracy is more-or-less (although not significantly) sensitive to the choice of numbers of prototypes. Intuitively, as long as the the number of prototypes does not match the number of classes in the down-stream task, noise in the semantic structure will be introduced. In order to render this phoneme, we implement a symmetric prototypical cross-entropy loss, defined in [38], to overcome the noisy prototypes. We conduct



(a) (b)

Figure 4: (a). Comparison of performance of top-1 accuracy w.r.t. various batch sizes and numbers of prototypes. (b). TP (**Left**) and FN (**Right**) similarity distance w.r.t. epoch at SimCLR (**Top**) and SPCL (**Bottom**) pre-training stage. Zoom in for a better view.

experiments on CIFAR-100 using 512 prototypes, and the symmetric loss achieves better performance (74.04%, 93.35%) compared to asymmetric loss (73.97%, 93.24%) in terms of top-1 and top-5 accuracy.

4.3.2 Effects of Losses

In this section, we analyze three loss modules described in Section 3. Specifically, we compare the performance of top-1 accuracy, True Positive (TP), and False Negative (FN) distance by manipulating the weights of the loss modules (contrastive, metric, and prototype) proposed. The decision of whether a sample being positive or negative is based on the ground-truth of down-stream tasks. Concretely, we look at the ground-truth of an anchor sample, then mask all the negatively paired samples with the same label as the FN. The notation *TP distance* refers to the similarity between an anchor image (**A**) and positive views (**TP**), and *FN distance* is the similarity between A and FN, on the contrastive projection space, *i.e.*. after $g_c(\cdot)$. A larger TP distance means that positive views are similar to the anchor image, which will be beneficial for the positive sample terms in a contrastive learning loss. If FNs share semantic information with larger similarity to the anchor, the discriminability between positive and negative samples would be obscured, leading to pre-trained representations with low quality. On the other hand, FNs that share less similar semantic information with the anchor help the contrastive loss discriminate the difference between positive and negative samples during pre-training. With less FNs that share most similar semantic information with the anchor, pre-trained representations would be of high-quality and advantageous.

Table 2: Ablation study of each loss module (contrastive, metric, and prototype) on CIFAR-10 in terms of top-1 accuracy, True Positive (TP) and False Negative (FN) distance. CE and GT denotes the cross-entropy loss and the ground-truth, separately.

method	contrastive(α)	metric(β)	prototype(γ)	top-1 (%)	TP distance(↑)	FN distance(↓)
Supervised (CE)	✗	✗	✓(GT)	93.02 ± 0.05	—	—
SimCLR [7]	✓	✗	✗	92.01 ± 0.13	0.68 ± 0.03	0.19 ± 0.03
SPCL	✓(1.0)	✓(1.0)	✓(GT)	**94.42** ± 0.09	**0.93** ± 0.02	**0.03** ± 0.04
SPCL	✓(1.0)	✓(1.0)	✓(1.0)	<u>94.12</u> ± 0.11	<u>0.85</u> ± 0.04	<u>0.11</u> ± 0.05
SPCL	✓(1.0)	✓(1.0)	✗	92.43 ± 0.12	0.76 ± 0.04	0.15 ± 0.06
SPCL	✓(1.0)	✗	✓(1.0)	85.67 ± 0.15	0.58 ± 0.02	0.27 ± 0.03
SPCL	✗	✓(1.0)	✓(1.0)	89.95 ± 0.18	0.61 ± 0.03	0.25 ± 0.04
SPCL	✓(0.1)	✓(1.0)	✓(1.0)	93.20 ± 0.15	0.81 ± 0.05	0.13 ± 0.04
SPCL	✓(1.0)	✓(0.1)	✓(1.0)	92.69 ± 0.12	0.78 ± 0.04	0.14 ± 0.05
SPCL	✓(1.0)	✓(1.0)	✓(0.1)	93.68 ± 0.11	0.83 ± 0.03	0.12 ± 0.04

The results are reported in Table 2. We either lower the importance of a loss or completely eliminate it. As can be seen, in the contrastive scene, using only the prototypical loss will not improve the performance over the baseline, while a small metric loss will significantly boost the performance in terms of both accuracy and the FN distance. This shows that the necessity of refining the quality of prototypes using the metric loss. On the other hand, removing the regularization from the prototypical loss also harms the performance. The pure clustering setting (that without the contrastive loss) also has degraded performance. Note that the FN distance in our proposed methods is much smaller than that in SimCLR, allows the model to build a better understanding of semantic information. Moreover, when employing the ground-truth label as the prototypes, the strong regularization effects eliminate the influence caused by FN samples, achieving the best top-1 accuracy.

In addition, we perform extensive experiments to analyze the effect of TP and FN distance on contrastive self-supervised learning. Specifically, we pre-train SimCLR and our proposed SPCL separately on the CIFAR-10 dataset by running 1000 epochs for 10 times. We calculate the TP/FN distance through the pre-training process by using different temperatures, as shown in Figure 4 (b). As can be seen in SimCLR, the average FN distance become larger with the increase of running epochs. We can observe that those two distances are sensitive to the choice of the temperature. As the temperature τ increases, it gives a larger penalty to the samples closed to the anchor, which leads to the increase of the TP distance and the decrease of the FN distance. In the first 30 epochs, the mean FN distance arise to the maximum value, which influences the quality of the subsequently learned representations.

Moreover, we report the TP and FN distance of our SPCL in two figures on the bottom row of Figure 4 (b). It can be seen that the TP and FN distance become more robust to the choice of the temperature in the contrastive loss. Moreover, the TP distances in our SPCL are much higher than those in SimCLR, which means that our positives share strongly similar semantic information with the anchor. The FN distances in our SPCL are lower than those in SimCLR. In this case, the FN shares less similar semantic information with the anchor such that our SPCL mitigates the semantic confusion brought by FN. In a nutshell, the increase of TP distances and the decrease of FN distances in our SPCL bring the true positives together and push semantically less similar false negatives apart in the pre-trained representations, boosting the performance of downstream tasks.

5 Conclusion

In this paper, we realize the possibility of *perform supervised training in the unsupervised configuration*. We propose SPCL, a simple but effective Siamese-style prototypical contrastive self-supervised learning framework. Specifically, we group the embedded features into separate clusters by a prototypical cross-entropy loss and employ the Siamese-style metric loss to match intra-prototype features. It is supposed to mitigate the semantically-similar confusion brought by false negatives in the contrastive loss term. We conduct extensive experiments on various benchmarks where the results demonstrate our method's effectiveness in improving the quality of visual representations for image classification.

Acknowledgement

The work is partially supported by JSPS KAKENHI (Grant No. 20K19875), the National Natural Science Foundation of China (Grant No. 62006045).

References

[1] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

[2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[3] Suzanna Becker and Geoffrey E. Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163, 1992.

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of European Conference on Computer Vision*, 2018.

[5] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2959–2968, 2019.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of International Conference on Machine Learning*, 2020.

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, 2021.

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia. Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[12] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[13] Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742, 2006.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020.

[17] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[18] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[19] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[21] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2020.

[22] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *Proceedings of International Conference on Machine Learning (ICML) Deep Learning Workshop*, 2015.

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[24] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[25] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6707–6717, 2020.

[26] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 360–368, 2017.

[27] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[29] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[30] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 5171–5180, 2019.

[31] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially consistent representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1144–1153, 2021.

[32] Virginia R. De Sa. Learning classification with unlabeled data. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, page 112–119, 1993.

[33] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[34] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[35] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[37] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3024–3033, 2021.

[38] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 322–330, 2019.

[39] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742, 2018.

[40] Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659*, 2020.

[41] Shin'ya Yamaguchi, Sekitoshi Kanai, Tetsuya Shioda, and Shoichiro Takeda. Multiple pretext-task for self-supervised learning via mixing multiple image transformations. *arXiv preprint arXiv:1912.11603*, 2019.

[42] Yang You, Igor Gitman, and Boris Ginsburg. Scaling SGD batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 2017.

[43] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of International Conference on Machine Learning (ICML)*, 2021.

[44] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6688–6697, 2020.

[45] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.