Solving Inefficiency of Self-supervised Representation Learning

Guangrun Wang[1,2]   Keze Wang[4]   Guangcong Wang[3]   Philip H.S. Torr[2]   Liang Lin[1*]

[1] Sun Yat-sen University   [2] University of Oxford   [3] Nanyang Technological University   [4] DarkMatter AI Research

{wanggrun,wanggc3,kezewang}@gmail.com,   philip.torr@eng.ox.ac.uk, linliang@ieee.org

Abstract

*Self-supervised learning (especially contrastive learning) has attracted great interest due to its huge potential in learning discriminative representations in an unsupervised manner. Despite the acknowledged successes, existing contrastive learning methods suffer from very low learning efficiency, e.g., taking about ten times more training epochs than supervised learning for comparable recognition accuracy. In this paper, we reveal two contradictory phenomena in contrastive learning that we call **under-clustering** and **over-clustering** problems, which are major obstacles to learning efficiency. Under-clustering means that the model cannot efficiently learn to discover the dissimilarity between inter-class samples when the negative sample pairs for contrastive learning are insufficient to differentiate all the actual object classes. Over-clustering implies that the model cannot efficiently learn features from excessive negative sample pairs, forcing the model to over-cluster samples of the same actual classes into different clusters. To simultaneously overcome these two problems, we propose a novel self-supervised learning framework using a truncated triplet loss. Precisely, we employ a triplet loss tending to maximize the relative distance between the positive pair and negative pairs to address the under-clustering problem; and we construct the negative pair by selecting a negative sample deputy from all negative samples to avoid the over-clustering problem, guaranteed by the Bernoulli Distribution model. We extensively evaluate our framework in several large-scale benchmarks (e.g., ImageNet, SYSU-30k, and COCO). The results demonstrate our model's superiority (e.g., the learning efficiency) over the latest state-of-the-art methods by a clear margin. See Codes[1].*

1. Introduction

Recently, self-supervised learning (SSL) has shown remarkable results in representation learning. Among them, the results of contrastive learning are most promising in the
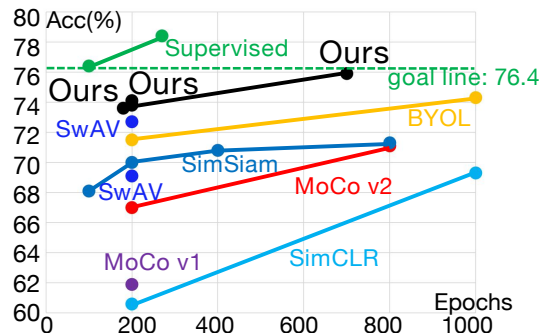

Figure 1. A comparison of learning efficiency among different SSL methods using ResNet-50. Here, the x-axis represents the training epochs of SSL, and the y-axis stands for the top-1 accuracy of ImageNet linear evaluation. All methods have lower learning efficiency than supervised learning, but our approach has a significantly higher learning efficiency than the existing SSL methods. (**best view in color**)

computer vision tasks. Notable works include MoCo v1/v2 [24, 8], SimCLR [7], BYOL [22], and SimSiam [9]. For example, on ImageNet [36], the top-1 accuracy of BYOL is 74.3%, which is close to that of supervised learning, i.e., 76.4% [49, 1, 28, 32] (see *"goal line"* in Figure 1). Despite the promising accuracies and high expectations, the learning efficiency of the state-of-the-art SSL methods is about ten times lower than the supervised learning methods. For instance, the supervised learning method typically takes about 100 epochs to train a ResNet50 on ImageNet. In comparison, SimCLR and BYOL have to cost 1,000 epochs, and MoCo v2 needs to cost 800 epochs (See Figure 1).

Attempting to address this issue, we rethink existing SSL methods' mechanism and attribute their inherited drawback to two opposing problems, i.e., **under-clustering** and **over-clustering**. Specifically, during batch training, contrastive learning randomly crops each image two times to obtain two views and study the similarity between these two views (called a **positive sample pair** [2]). Meanwhile, some methods also study the dissimilarity between cross-image views (called **negative sample pairs**[3]). The optimization objective is to reduce the distance between posi-

---

[2] e.g., View A and View B of Image X

[3] e.g., View A of Image X and View B of Image Y

tive sample pairs and enlarge the distance between negative sample pairs. As suggested by metric learning [12], sufficient negative sample pairs are required to guarantee the learning efficiency. Otherwise, lacking negative samples – whether due to the GPU memory constraints like SimCLR or (ii) algorithm design like BYOL and SimSiam [9] – can make different object categories having overlaps. This is identified as the *under-clustering* problem. One evidence of under-clustering is shown in Table 1[4]. As a result of under-clustering, SimCLR and BYOL have low learning efficiency because the model cannot efficiently discover the dissimilarity between inter-class samples. On the contrary, excessive negative samples can lead to an opposite problem, i.e., *over-clustering*, which implies the negative samples are false negative and the model over-clusters samples of the same actual categories into different clusters. In an extreme case, there would be 1.28M clusters for ImageNet! One evidence of over-clustering is in Table 1[5]. Over-clustering also results in low learning efficiency since it encourages dissimilarity between intra-class samples in vain. As reported by [54, 3], over-clustering can lead to unnecessary harmful representation learning. For example, [15, 14] obtains an unsatisfied performance due to directly clarifying CIFAR-10 into 50K clusters. MoCo v1/v2 cannot further increase the accuracy, even leveraging the momentum to store plenty of negative samples. In summary, existing contrastive learning cannot avoid the under-clustering or over-clustering problems, so their learning efficiency is still low.

To tackle the above under-clustering and over-training problems, a few pioneering works have been proposed to analyze the negative samples' role in the contrastive loss [10, 35, 4, 27]. As opposed to these methods that use overcomplicated contrastive losses, we propose an SSL framework using a quite simple truncated triplet loss. Specifically, a triplet loss can maximize the relative distance between the positive and negative pairs for each triplet unit. Having plenty of triplets, we can address the underclustering problem because wealthy triplets contain rich negative pairs that guarantee a considerable distance between negative sample pairs. Triplet loss largely addresses the under-clustering issue but raises the over-clustering problem. Hence, we propose a novel truncated triplet loss to avoid over-clustering samples from the same category into different clusters. The truncated triplets are ensured with confidence guaranteed by the Bernoulli Distribution model. This significantly improves SSL's learning efficiency and

Under-clustering (Divergence)	BYOL	**Ours**
	5.3711	**7.6803**
Over-clustering ($\Pr(\omega|\mathcal{A})$)	MoCo v2	**Ours**
	1.0	**0.0110**

Table 1. Qualitative analysis of over-/ under- clustering. We use $\Pr(\omega|\mathcal{A})$ (the larger, the higher over-clustering risk) and class divergence (the smaller, the higher under-clustering risk) to measure the over-/ under- clustering level, respectively.

leads to state-of-the-art performance (See Figure 1).

In summary, our contribution is three-fold.

- We analyze the existing best-performing contrastive learning methods and attribute their low learning inefficiency to the under-clustering and over-clustering, which result in unnecessary harmful representation learning just to memorize the data.

- To address the under-clustering and over-training problem, we propose a novel SSL framework using a truncated triplet loss. Precisely, we employ a triplet loss containing rich negative samples to address the underclustering problem, and our triplet loss uses truncated/trimmed triplets to avoid over-clustering, guaranteed by the Bernoulli Distribution model.

- Our method significantly improves SSL's learning efficiency and thus leads to state-of-the-art performance in several large-scale benchmarks (e.g., ImageNet [36], SYSU-30k [46], and COCO 2017 [29]) and varieties of downstream tasks.

2. Related Work

Vanilla SSL. The recent renaissance of SSL originated from straightforward pretext tasks. Typical pretext tasks included image denoising [41], image inpainting [33], patch ordering [13], solving jigsaw puzzles [31], color jittering [55], and rotation prediction [19]. Although these methods contributed to the renaissance of SSL, their learned representations did not generalize well.

Contrastive learning. Currently, the most effective SSL method in computer vision is contrastive learning [7, 24, 8, 22, 6], in which the intra-class distances are encouraged to be small, and the inter-class distances are forced to be large[6]. Plenty of positive and negative samples are needed to discover the similarity and dissimilarity, which requires large GPU memories [7]. To address this problem, SimCLR [7] employs multi-machine distributed computing to enlarge the batch. Nevertheless, due to GPU memory limitation, further increasing positive/negative samples is prohibitive in practice, which forms a barrier to improving SSL. We identify this as an under-clustering problem.

To avoid under-clustering, more elegantly, Mean Teacher [38] is applied to produce sufficient negative [24, 8] and

---

[4]We calculate the class center for each category and compute the distances for every two class centers. These center-to-center distances are averaged to form a class divergence. We keep the variance equal so we can just compare class divergence. The small class divergence in Table 1 indicates BYOL does suffer under-clustering.

[5]We use $\Pr(\omega|\mathcal{A})$ (defined in Section 5) to denote the possibility of containing a false negative samples in a batch. The high probabilities in Table 1 verify MoCo v2 indeed suffers over-clustering.

---

[6]although the class labels are unknown.

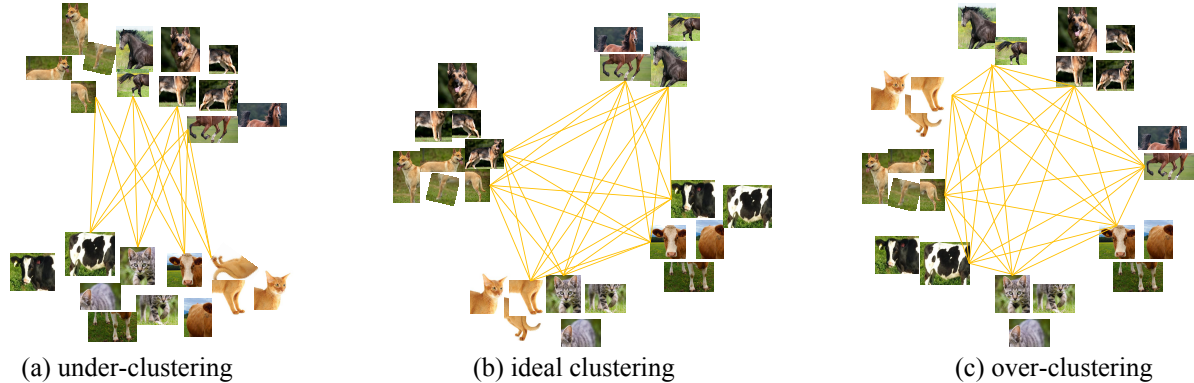(a) under-clustering   (b) ideal clustering   (c) over-clustering

Figure 2. Illustration of under-clustering and over-clustering. Each sample pair connected by a yellow line represents a negative pair.

positive samples [22, 6]. Moreover, Exemplar-CNN [15, 14] directly clarifies each image in a dataset into a cluster, i.e., it categorizes CIFAR-10 into 50K clusters. However, it obtains an unsatisfactory performance. We identify this as an over-clustering problem. Specifically, since each image can be regarded as a cluster, excessive negative sample pairs can over-cluster samples from the same category into different clusters. This over-clustering can lead to bad representation learning since the network just memorizes the data instead of learning from the data [54, 3].

To reduce over-clustering, recent works rethink the necessity of negative samples and propose to remove negative samples at all. Notable works include BYOL [22] and SimSiam [9]. However, once the negative samples are removed, under-clustering *may probably* reoccur because the model cannot efficiently discover the dissimilarity between inter-class samples (see Table 1 for the evidence). Besides, a few other pioneering works are also proposed to analyze the negative samples' role in the contrastive loss. [4] used empirical evidence to show that not all negatives are equally important for contrastive learning. [27] used a complicated way to cancel false negatives. [10, 35] observed that using extremely close samples is bad for contrastive learning and leverage distribution knowledge to solve the problem. As opposed to these methods that use over-complicated contrastive losses, we use a quite simple triplet loss.

Triplet loss. Triplet loss was proposed by Ding et al. [12] and Schroff et al. [37] independently for person re-identification and face recognition, respectively. It tends to maximize the relative distance between the positive pair and the negative pair for each triplet unit. Several improvements over triplet loss are conducted to discover the valuable triplets [26], to perform cross-batch triplet loss [48], and to apply to weakly supervised scenario [46]. However, these classical triplet losses can also result in over-clustering. In contrast, we propose a truncated triplet loss to address the over-clustering problem guaranteed by the Bernoulli Distribution model.

Recognizing the hardest negatives can lead to bad local minima in practice, [37] proposes a semi-hard negative sampling strategy, sharing the merit of our method in avoiding over-trusting the hardest negative sample and benefiting representation learning. The differences between our truncated triplet loss and semi-hard sampling strategy are two-fold. First, if we read the widely-used code in TensorFlow and Pytorch, we can find that a semi-hard triplet loss is a margin-based loss rather than a ranking loss. Second, our method trims negative samples while [37] performs vanilla sampling. Our approach differs from [37]'s vanilla sampling and mixture sampling and is thus a new one.

3. Self-supervised Representation Learning

We first present the under-clustering and over-clustering problems of contrastive learning in Section 3.1. Then, we present our method in Section 3.2. The analysis of the effectiveness of our approach is presented in Section 3.3.

3.1. Under-clustering and Over-clustering

Contrastive learning is proposed by [23] and is widely used in SSL, achieving best-performing results on ImageNet. The most widely-adopted loss for contrastive learning is InfoNCE [40]. Let x be a query image which has 1 positive sample x^+ and m negative samples $\{x_j^-\}_{j=1,\cdots,m}$. InfoNCE calculates their inner products and normalize the products using softmax and have:

$$\{\widetilde{x^T x^+}, \widetilde{x^T x_1^-}, \widetilde{x^T x_2^-}, \cdots, \widetilde{x^T x_m^-}\}$$
$$= softmax(\{x^T x^+, x^T x_1^-, x^T x_1^-, \cdots, x^T x_m^-\}).$$

Then, the goal of contrastive learning is to minimize:

$$-1\log \widetilde{x^T x^+} - 0\log \widetilde{x^T x_1^-} - 0\log \widetilde{x^T x_2^-} \cdots - 0\log \widetilde{x^T x_m^-}.$$

(see footnote[7]), which can be interpreted as forcing $\widetilde{x^T x^+}$ to be close to 1 and forcing $\widetilde{x^T x_1^-}, \widetilde{x^T x_2^-}, ..., \widetilde{x^T x_m^-}$ to be close to 0. This indicates that plenty of negative sample pairs are needed to guarantee the learning efficiency because the

---

[7]Generally, it's written as: $-\log \frac{\exp(x^T x^+)/\tau}{\exp(x^T x^+)/\tau + \sum\limits_{j=1}^{m} \exp(x^T x_j^-)/\tau}$,

where τ is a temperature.

model needs sufficient negative samples to discover the dissimilarity between inter-class samples. Especially, plenty of positive samples and negative samples are needed to enrich the similarity and dissimilarity *in each batch of data*.

Under-clustering. Insufficient positive and negative examples can lead to under-clustering. Under-clustering is a critical problem in which different categories have a valid (but unwelcome) overlap. For example, in Figure 2 (a), a cluster may contains {dog, horse} or {cat, cow}, i.e., dogs and horse are mixed up. Without the annotation, we cannot identify the actual label of each data point. In other words, the dogs and horses have an overlap. An under-clustering problem occurs when insufficient positive and negative samples are present.

Over-clustering. As opposed to under-clustering caused by lacking negative samples, over-clustering is caused by overwhelming negative samples. Although contrastive learning implicitly regards each image as a class, we do not expect over-clustering. Excessive negative sample pairs can result in over-clustering that forces samples from the same category into different clusters. As is shown in Figure 2 (c), if excessive negative examples are provided, the two dogs that belong to the same category are now assigned to two clusters. Similar phenomena also appear in cats and cows. This non-ideal over-clustering would prevent the model from learning discriminative representations summarizing essential features of a category since the network just memorizes the data instead of learning from the data [54, 3].

Ideally, we would like to use just the right amount of negative sample pairs to ensure that the images from the same category are close to each other and that the images from different classes are far away. As shown in Figure 2 (b), all the dogs, cats, and cows are clustered correctly. Note that this is achieved in an unsupervised manner.

3.2. Truncated Triplet Loss

Triplet loss. Inspired by relative distance comparison [57], triplet loss was proposed by [12] and [37] independently for person re-identification and face recognition, respectively. In a triplet-loss method, a set of triplets, i.e., $\{(x_i, x_i^+, x_i^-)\}_{i=1,\cdots,m}$, are first generated. In general, a query image will have far more negative samples than positive samples (see Figure 3 (a) for detail). For presentation simplicity, we use only one query image and one positive sample for illustration, i.e., we have a triplet set $\{(x, x^+, x_i^-)\}_{i=1,\cdots,m}$. The oldest triplet loss is defined as:

$$\mathcal{L}oss = \sum_{i=1}^{m} \max\left(d(x, x^+) - d(x, x_i^-), \mathcal{C}\right),$$

where d is a distance metric (e.g., cosine distance or Euclidean distance). Here, $\mathcal{C}$ is a margin deciding whether or not to drop a triplet. This is critical in machine learning algorithms since we usually drop the simple data and focus on the hard data near the decision boundary, as support vector machine [11] suggests. To improve the learning efficiency

of triplet loss, in practice, we usually use the hardest triplet to represent the overall triplets, i.e., only the triplet containing the negative sample of the highest similarity score overall negative samples are used (please refer to Figure 3 (b) for detail). Finally, a triplet is formally defined as:

$$\mathcal{L}oss = \max\left(d(x, x^+) - d(x, x_{hardest}^-), \mathcal{C}\right). \quad (1)$$

Since $x_{hardest}^-$ is the hardest negative sample, we have $d(x, x_{hardest}^-) \leq d(x, x_i^-)$ for all i. This indicates that when the hardest triplet loss meets the condition $d(x, x^+) \leq d(x, x_{hardest}^-)$, all others triplets meet the condition. Therefore, the hardest triplet loss guarantees a considerable distance between negative sample pairs. Using triplet loss, we can address the under-clustering problem.

Although triplet loss largely reduces under-clustering, it increases over-clustering risk. Specifically, since contrastive learning can be considered a classification problem that identifies each image as a class, using the hardest triplet loss can lead to over-clustering. For example, in Figure 3 (a), the two dogs belong to the same object category. Unsurprisingly, their feature similarity is high. But in SSL, the actual category labels are absent; thus, these two dogs can be reluctantly considered negative sample pairs (Figure 3 (b) and (c), **top**). This indicates they are the hardest negative sample pairs. Using the hardest triplet loss, the distance between these two dogs is enlarged. This results in an over-clustering problem in that the two dogs from the same category are over-clustered into two different clusters.

Truncated triplet loss. To avoid over-clustering, we construct the negative pair by truncating/trimming the hardest negative samples. We select a negative sample deputy to form a truncated triplet, i.e., we have:

$$\mathcal{L}oss = \max\left(\gamma d(x, x^+) - d(x, x_{deputy}^-), \mathcal{C}\right). \quad (2)$$

Specifically, $d(x, x_{deputy}^-)$ is obtained using the following steps. First, we compute the distance $\{d(x, x_i^-)\}, \forall i$. Then, we sort $\{d(x, x_i^-)\}$ by ascending. Finally, we obtain $d(x, x_{deputy}^-)$ in two ways:

- *rank-k triplet loss:* the k-th element are selected from $\{d(x, x_i^-)\}$, yielding: $d(x, x_{deputy}^-) = d(x, x_{rank-k}^-)$.

- *smoothed-rank-k triplet loss:* selecting the top-2, top-3, ..., top-(2k+1) elements from $\{d(x, x_i^-)\}$ and yielding: $d(x, x_{deputy}^-) = \frac{1}{2k} \sum_{j=2}^{2k+1} d(x, x_{rank-j}^-)$.

Note that when $k = 1$, the rank-k triplet loss reduces to the hardest triplet loss. We can show in Section 3.3 that replacing the hardest triplet with the rank-k triplet can indeed reduce the risk of over-clustering, guaranteed by the Bernoulli Distribution model. By default, we use $k = \frac{m}{2}$ (i.e., the triplet deputy) for most of the experiments, although using other values (e.g., $k = 5$) also yields good
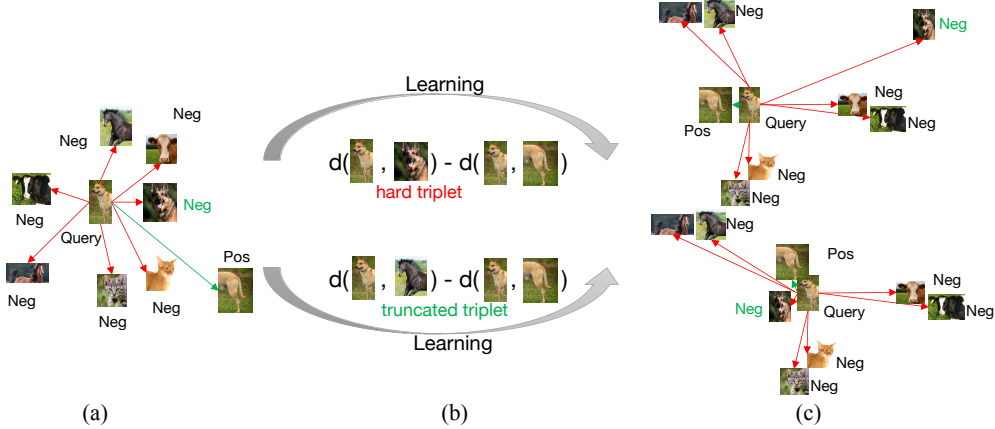
Figure 3. Illustration of our truncated triplet loss. Each sample pair connected by a red line represents a negative sample pair. Note that although labeled by a green word "Neg", in fact, the dog is a positive sample to the query image because they belong to the same category, i.e., "a dog". Traditional triplet loss using the hardest triplet (see Figure (b)) results in over-clustering, in which the distance between these two dogs will be enlarged. The over-clustering result is shown at (c) **top**, and the ideal learning results is shown at (c) **bottom**.

performance. We use the widely-used cosine distance for d, i.e., $d(x, y) = -\frac{xy}{\|x\|_2 \|y\|_2}$, where $\| \cdot \|_2$ represents the L2 norm. The negative sign ("-") is used here because we usually consider that a higher similarity indicates a smaller distance. We set γ to be 2. *Note that, according to our experiments, the smoothed-rank-k triplet loss achieves better performance than the rank-k triplet loss.*

3.3. Analysis with Bernoulli Distribution Model

In contrastive learning, views from different images are considered negative sample pairs even they are from the same actual category (e.g., the two dogs in Figure 3 (a)). Unsurprisingly, these kinds of negative sample pairs have a high feature similarity. With a high probability, they will be in the hardest triplets (see Figure 3 (b)). Using the hardest triplet loss, the distance between these false-negative pairs is enlarged. This results in an over-clustering problem that the false-negative pairs from the same category are over-clustered into two different clusters (see Figure 3 (c) **top**).

But in our truncated triplet loss, we sort $\{d(x, x_i^-)\}$ by ascending and select the k-th element from $\{d(x, x_i^-)\}$ to form $d(x, x_{deputy}^-)$. An over-clustering risk exists if this rank-k negative sample and the query image belong to the same actual category. We need to estimate the probability that these two images belong to the same category. We first have a reasonable assumption: with a high probability, the image pairs from the same category have higher feature similarities than other pairs, and the distances between these pairs are smaller than other pairs. Because we have sorted $\{d(x, x_i^-)\}$ by ascending, the event that the rank-k negative sample and the query belong to the same category indicates an event that at least k negative samples and the query belong to the same category. The probability of this event can be computed by using a Bernoulli Distribution model, i.e., $\mathrm{Pr} = \sum_{j=k}^{m} \mathbf{C}_m^j p^j (1-p)^{m-j}$. Here, $\mathbf{C}$ is used to denote the

combinations, and p is used to denote the probability that a negative sample and the query belong to the same class. For example, on ImageNet, we have $p = \frac{1}{1000}$. In our experiment, we let m be 104 and k be $\frac{m}{2}$. Putting m, k into the above equation, we have $\mathrm{Pr} = 6.53e^{-121}$, which is almost zero. Even we let m be 104 and k be 5, we have $\mathrm{Pr} = 3.03e^{-94}$. This indicates it is a rare event that the rank-k negative sample and the query belong to the same category. Thus, our truncated triplet loss can avoid over-clustering, guaranteed by the Bernoulli Distribution model, e.g., the two dogs at the bottom of Figure 3 (c) can be identified correctly. Experimental results in Section 5 also verify the effectiveness of our method[8].

4. Main results

Our SSL training protocols are as follows.

Data augmentation protocol. Our augmentations are straightforward, including randomly cropping, randomly resizing, randomly flipping horizontally, arbitrary gray scaling, stochastic color jittering, Gaussian blurring, and solarization. Please see our codes.

Other protocols. In the unsupervised learning stage, the batch size is 104 images per GPU, and we use eight GPUs. The gradient update interval is five steps. The maximum epoch is 200. The learning rate starts from 4.8 and gradually decreases with cosine annealing. The weight decay factor is $1e^{-6}$. The optimizer is LARS [20] with a momentum 0.9. The backbone is ResNet-50, which is the same as the previous methods. The models are trained by using the 1.28M training images of ImageNet but without their annotations. The protocols are in line with [56]: we use the

---

[8]The Bernoulli Distribution model is simple yet can be justified by the experiment (Section 5). Meanwhile, [10, 35] used elegant mathematical boundedness tools to show the generalization ability of contrastive learning. A more elegant justification using boundedness tools is welcome.

Table 2. Top-1 accuracy and training epochs of state-of-the-art methods on ImageNet using linear classification for evaluation.

Method	top-1 acc.	train epochs
Random	4.4	0
Relative-Loc [13]	38.8	200
Rotation-Pred [19]	47.0	200
DeepCluster [5]	46.9	200
NPID [50]	56.6	200
ODC [53]	53.4	200
SimCLR [7]	60.6	200
SimCLR [7]	69.3	1000
MoCo [24]	61.9	200
MoCo v2 [8]	67.0	200
MoCo v2 [8]	71.1	800
SwAV [6] (single-crop)	69.1	200
SwAV [6] (multi-crop)	72.7	200
BYOL [22]	71.5	200
BYOL [22]	72.5	300
BYOL [22]	74.3	1000
SimSiam [9]	68.1	100
SimSiam [9]	70.0	200
SimSiam [9]	70.8	400
SimSiam [9]	71.3	800
truncated triplet	73.6	180
truncated triplet (smoothed)	73.8	200
truncated triplet (multi-crop)	74.1	200
truncated triplet	75.9	700
supervised	76.3	100
supervised + linear eval	74.1	100
supervised	78.4	270

Table 3. Object detection results on COCO 2017 for Mask-RCNN.

Method	AP^{Box}	AP^{Mask}
Random	35.6	31.4
Relative-Loc [13]	40.0	35.0
Rotation-Pred [19]	40.0	34.9
NPID [50]	39.4	34.5
MoCo [24]	40.9	35.5
MoCo v2 [8]	40.9	35.5
SimCLR [7]	39.6	34.6
BYOL [22]	40.3	35.1
truncated triplet	41.3	37.3
supervised-100	40.0	34.7
supervised-270	42.0	37.7

Table 4. Object detection results on VOC07+12 for Faster-RCNN.

Method	$AP50^{Box}$	AP^{Box}	$AP75^{Box}$
Random	59.0	32.8	31.6
Relative-Loc [13]	80.4	55.1	61.2
Rotation-Pred [19]	80.9	55.5	61.4
NPID [50]	80.0	54.1	59.5
MoCo [24]	81.4	56.0	62.2
MoCo v2 [8]	82.0	56.6	62.9
SimCLR [7]	79.4	51.5	55.6
BYOL [22]	81.0	51.9	56.5
truncated triplet	81.8	56.4	62.9
supervised-100	81.6	54.2	59.8
supervised-270	82.2	56.9	63.1

same momentum network, the same multilayer-perceptron head & neck as BYOL. Also, following BYOL, our loss is symmetric *w.r.t* the positive pairs. Using protocols similar to [56] makes it feasible to present comparisons on multiple datasets/tasks without the extra hyper-parameter search.

We evaluate our method by comparing it with state-of-the-art methods in four tasks, involving linear evaluation on ImageNet, person re-identification on SYSU-30k, and object detection on both COCO 2017 and "VOC07+12".

4.1. Linear evaluation on ImageNet

Linear evaluation is the most widely adopted evaluation protocol for validating the representation ability of different SSL methods. Standardly, the backbones of ResNet-50 are trained by using the above SSL training protocols and are frozen. A linear classifier is then added to the top of the frozen representation and is trained for each method. All methods are trained using the 1.28M training images of ImageNet and are evaluated using the 50K validation images of ImageNet. For the linear classification stage, the batch size is 256. The maximum epoch is 100. There is no weight decay in linear classification training. The optimizer is SGD. Single-scale center-crop top-1 accuracy is used.

Currently, the widely-used evaluation standard of SSL merely values the accuracy but regardless of the training epochs. Following this standard, we first compare our method with the competitors without considering the training epochs. Table 2 shows that our method achieves a promising result on ImageNet, i.e., 75.9%, outperforming the latest state-of-the-art methods by a clear margin.

Regarding learning efficiency, existing SSL methods' efficiency is lower than supervised learning. As shown in Table 2, SSL models are trained for about 1,000 epochs, while

the supervised counterpart is merely trained for 100 epochs. SimCLR [7] explains that training for a longer time does not bring gain for the supervised learning model (i.e., it reported a result of 76.4% vs. 76.3% for 1000 epochs' vs. 100 epochs' supervised training). But our observation is the opposite, i.e., our reproduction of a supervised model trained for 270 epochs can achieve 78.4% top-1 accuracy, which is significantly higher than all of the SSL models.

Put together, a complete comparison regarding both accuracies and training epochs is presented in Figure 1 and Table 2, where we have three observations. **First**, previous SSL methods still have a long way to go. They have significantly lower learning efficiency than supervised learning. **Second**, as shown in Figure 1, our approach lies in the top-left corner of the figure, indicating our method achieves the best performance among the compared SSL methods. For example, SwAV [6] achieves 72.7% (200 epochs), which is lower than our method (73.6%, 180 epochs). **Note** that SwAV uses an additional multi-crop augmentation that we don't use, which has a 3.6% gain[9] (without multi-crop augmentation, SwAV only achieves 69.1%). For a fair comparison, we add multi-crop augmentation to our method, leading to a further state-of-the-art result of 74.1% (200 epochs). This comparison verifies the effectiveness and efficiency of our approach. **Third**, the smoothed truncated triplet loss achieves better performance than the truncated triplet loss.

4.2. Transferring to downstream tasks

Transferring to COCO 2017 object detection. One of SSL's goals is to learn transferrable features. We test our learned representation's generalization ability by transferring to COCO 2017 object detection [29], *which is (one of)*

---

[9]Many methods (e.g., HSA [51] / MoCo v2 [8] / SimCLR [7] / SeLa [2] / DeepCluster [5]) can benefit significantly from multi-crop augmentation, but BYOL [22] cant't.

the largest benchmarks for general object detection, containing about 119K training images. Specifically, the backbones of ResNet-50 are trained by using the above SSL training protocols, and the trained network weights serve as the initialization of Mask-RCNN [25] with C4. We fine-tune all layers on the *train2017* set. The training schedule is the default 2× schedule in [21]. Following [24], we fine-tune BN instead of freezing it. Overall, the SSL pretraining methods use the same training protocol as the ImageNet supervised counterpart. The accuracy is tested on the *val2017* set. We report the standard metric for the detection and instance segmentation: AP^{Box} and AP^{Mask}.

Table 3 shows that using our approach for pretraining surpasses other SSL ImageNet pretraining on COCO 2017 detection. *Our method indeed surpasses prior arts (including MoCo / MoCo v2) on object detection on COCO 2017.* Moreover, our SSL pretraining even outperforms the supervised ImageNet pretraining, implying that SSL can obtain more universal representations. This is in line with previous works that also show that SSL pretraining can outperform supervised pretraining on object detection [18, 24, 30, 6].

Transferring to VOC07+12 object detection. In addition to COCO 2017, we also evaluate our method's transferability in PASCAL VOC object detection. Following [24] and [56], the backbones of ResNet-50 are trained by using the above SSL training protocols, and the trained network weights serve as the initialization of Faster R-CNN [34] with C4. Then, we fine-tune all layers on the *trainval07+12* set of PASCAL. The image scale is [480, 800] pixels during training and 800 in the testing. We report the default VOC metric of AP^{50} and the COCO-style AP and AP^{75}. The evaluation is on the VOC *test2017* set.

We show the results of different methods in Table 4. As shown, our approach achieves comparable performance to MoCo / MoCo v2 (our method is better than MoCo and slightly worse than MoCo v2). Note that only MoCo v2 and our approach can catch up with the supervised pretraining counterpart that performs pretraining for 100 epochs. This verifies the effectiveness of our method and implies that our SSL can obtain universal representations.

4.3. Person re-identification on SYSU-30k

In a general sense, all the above tasks (image classification, object detection, and segmentation) belong to visual categorization since detection and segmentation can be considered categorizing regions and pixels for a given image. The effectiveness of our approach beyond visual classification remains uninvestigated. In the following, we investigate a different task, i.e., person re-identification (re-ID), which is fundamental in video surveillance [17]. Re-ID refers to the problem of re-identifying individuals across cameras [45]. Mathematically, re-ID is a matching problem rather than a classification problem because it requires cal-

Table 5. Comparison with state-of-the-art methods on SYSU-30k.

supervision	method	rank-1
Transfer learning	DARI [44]	11.2
	DF [12]	10.3
	Local CNN [52]	23.0
	MGN [47]	23.6
Weakly supervised	W-Local CNN [46]	28.8
	W-MGN [46]	29.5
Self-supervised	SimCLR [7]	10.9
	MoCo v2 [8]	11.6
	BYOL [22]	12.7
	truncated triplet	**14.8**

culating distance metrics between two given images. As is proved by [16, 43, 46, 58, 42], unsupervised representation learning is critical to visual matching; therefore, validating the effectiveness of our approach in re-ID is nontrivial.

Dataset and protocol. We conduct experiments on the SYSU-30k dataset [46], which is the largest database for re-ID. This database contains 29,606,918 images of 30,508 pedestrians, which is about 30 times larger than ImageNet in terms of category number. Please note that the exact label of each image is unknown in this dataset. Both the lack of precise annotation and the massive number of images make this dataset set very suitable for unsupervised learning, especially SSL. Since we are the first to perform SSL on this database, no previous work provides an evaluation protocol for this dataset. We propose a new evaluation protocol for it: the training set of SYSU-30k is employed to perform SSL. Once the model is learned, we directly use it to extract features for visual matching on the test set of SYSU-30k without any finetuning. This is even more challenging than linear evaluation on ImageNet because linear evaluation learns an extra classifier for recognition, but no extra classifier is learned here. Regarding the superiority of using SYSU-30k mentioned above, we believe SYSU-30k is a perfect database to evaluate SSL effectiveness and recommend it to future SSL researchers.

Result analysis. We compare our method with SimCLR, MoCo-v2, BYOL, and current state-of-the-art results (not SSL methods). We use ResNet-50 as the backbone. The results in Table 5 show that our models achieve new state-of-the-art performance, i.e., a rank-1 accuracy of 14.8%. Note that this number is low, i.e., even lower than existing transfer learning and weakly supervised learning methods. This is attributed to the challenge of the SYSU-30k test set, which contains about 480,000 **testing** images. Moreover, there are 478,730 mismatching images as the wrong answer in the gallery. Thus, evaluation using the SYSU-30k test set is like searching for a needle in a haystack. We encourage future SSL researchers to use this dataset to evaluate the effectiveness of SSL. We can also observe that our approach surpasses other SSL methods by a clear margin (14.8 vs. 12.7 for ours vs. BYOL). This verifies the effectiveness of our approach on visual matching tasks like re-ID.

Table 6. Effect of avoiding over-clustering.

Training epoch	event	0	180
$k = 1$	$\Pr(\Omega|\mathcal{A})$	0.1538	0.9656
	$\Pr(\Omega|\mathcal{B})$	0.1618	0.9948
$k = 5$	$\Pr(\Omega|\mathcal{A})$	0.1167	0.2105
	$\Pr(\Omega|\mathcal{B})$	0.1230	0.2132
$k = 52$	$\Pr(\Omega|\mathcal{A})$	0.1220	0.0110
	$\Pr(\Omega|\mathcal{B})$	0.1288	0.0233

Table 7. Impact of margin.

Margin	$\mathcal{C} = -0.3$	$\mathcal{C} = -1.2$	$\mathcal{C} = -100$
Top-1 accuracy	28.3	29.8	30.0

Table 8. Impact of rank-k.

Rank-k	rank-1	rank-5	rank-52
Top-1 accuracy	28.9	29.5	30.0

5. Ablation studies

In the rest of our paper, to reduce the training time and fast access to the results, we perform ablation studies using 20 training epochs. Please note that, due to our method's high learning efficiency, training for 20 epochs is sufficient for ablation studies. The linear evaluation training is also reduced to one epoch [10]. Actually, previous works also use few training epochs for ablation studies, e.g., [7] and [39]. All the training protocols are the same as Section 4, except that we take the 20th epoch's checkpoint for one epoch's evaluation. This section only reports the results of top-1 accuracy on ImageNet under the linear evaluation protocol since it is the most widely adopted metric for validating the effectiveness of SSL methods.

Effect of avoiding over-clustering. As we discussed in Section 3.2, thanks to the truncated triplet loss, we can avoid over-clustering. For example, if $k = 5$, the probability of over-clustering is $3.03e^{-94}$. If $k = \frac{m}{2} = 52$, the probability of over-clustering is $6.53e^{-121}$. However, whether this analysis is correct remains unclear. In the following, we provide empirical analysis. During batch training, all the batch samplings are considered the total event $\mathcal{A}$. If a batch contains at least two images belonging to the same actual category, we call it an event $\mathcal{B}$. If (at least) these two images are mis-considered a false-negative pair, we call this an event Ω. We report the frequency $\Pr(\Omega|\mathcal{A})$ and $\Pr(\Omega|\mathcal{B})$ in Table 6 for different training epochs and different ks.

We have three observations from Table 6. **First**, the rank-52 and rank-5 negative samples rarely belong to the same category as the query image, i.e., $\Pr(\Omega|\mathcal{A})$ and $\Pr(\Omega|\mathcal{B})$ are low. **Second**, with the training epochs increasing, the probability that the rank-k negative sample is a false negative decreases. This is attributed to the more and more discriminative features that have been learned as the training goes. **Third**, with k increasing, the probability that the rank-k negative sample is false negative decreases. Especially when $k = 1$, our truncated triplet loss reduces to

---

[10] After running plenty of experiments, our practical experience shows that 20 epochs of training are enough for ablation studies.

the hardest triplet loss. As shown, the hardest triplet loss indeed has a risk of over-clustering because the probability $\Pr(\Omega|\mathcal{B})$ is high. With k increasing, the probability $\Pr(\Omega|\mathcal{B})$ decreases. This indicates that our truncated triplet loss can avoid the over-clustering problem, guaranteed by the Bernoulli Distribution model.

Please note that if a batch contains even one false negative sample pair, we consider the whole batch has the over-clustering risk. Therefore, the probability in Table 6 (e.g., 0.0110 or 0.2105) is higher than that in the analysis (e.g., $3.03e^{-94}$ or $6.53e^{-121}$).

Impact of margin. As we discussed in Section 3.2, there is a margin $\mathcal{C}$ deciding whether or not to drop a triplet. This is critical in machine learning algorithms since we usually drop the simple data and focus on the complex data near the decision boundary, as support vector machine [11] suggests. To empirically verify this hypothesis, we train our method using different margins. The results are shown in Table 7. As shown, different margins lead to a performance fluctuation. When $\mathcal{C} = -100$ or $\mathcal{C} = -1.2$, the performance is best. Hence, we use $\mathcal{C} = -100$ for default in all of the experiments if no otherwise specified.

Impact of rank-k. As we analyze in Section 3.3, we can use different ks for our truncated triplet loss. When $k = 1$, our truncated triplet loss reduces to the traditional hardest triplet loss. When k increases, the risks of over-clustering reduces exponentially. To know the impact of using different ks, we train our method using different ks. As shown in Table 8, different ks lead to a performance fluctuation. When $k = 5$ and $k = 52$, the performances are satisfied. Hence, we use $k = 5$ or $k = 52$.

6. Conclusion

Although SSL has shown promising results on ImageNet, its learning efficiency is still low. We attribute the inherited drawback of contrastive learning to under-clustering and over-clustering. To overcome these two problems, we propose a novel SSL framework using a truncated triplet loss. We employ triplet loss containing rich negative sample information to address the under-clustering problem. We trim negative samples to prevent the over-clustering problem, guaranteed by the Bernoulli Distribution model. Our method significantly improves the learning efficiency of SSL, leading to a state-of-the-art performance in several large-scale benchmarks and varieties of downstream tasks.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[3] Guy Bresler and Dheeraj Nagaraj. A corrective view of neural networks: Representation, memorization and learning. In *Conference on Learning Theory*, pages 848–901. PMLR, 2020.

[4] Tiffany Tianhui Cai, Jonathan Frankle, David J. Schwab, and Ari S. Morcos. Are all negatives created equal in contrastive instance discrimination? *CoRR*, abs/2010.06682, 2020.

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*, pages 139–156, 2018.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *CoRR*, abs/2006.09882, 2020.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

[8] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2020, virtual conference online, June 19-25, 2021*, page , 2021.

[10] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[12] Shengyong Ding, Liang Lin, Guangrun Wang, and Hongyang Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognit.*, 48(10):2993–3003, 2015.

[13] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1422–1430, 2015.

[14] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(9):1734–1747, 2016.

[15] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 766–774, 2014.

[16] Hehe Fan, Liang Zheng, Chenggang Yan, and Yi Yang. Unsupervised person re-identification: Clustering and fine-tuning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(4):1–18, 2018.

[17] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2360–2367. IEEE, 2010.

[18] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6926–6936, 2020.

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[20] Boris Ginsburg, Igor Gitman, and Yang You. Large batch training of convolutional networks with layer-wise adaptive rate scaling, 2018.

[21] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020.

[23] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742, 2006.

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2020*, 2020.

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988, 2017.

[26] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017.

[27] Tri Huynh, Simon Kornblith, Matthew R. Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. *CoRR*, abs/2011.11765, 2020.

[28] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, pages 675–678, 2014.

[29] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755, 2014.

[30] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6706–6716, 2020.

[31] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, pages 69–84, 2016.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.

[33] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2536–2544, 2016.

[34] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[35] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR 2021, virtual*, 2021.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[37] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823, 2015.

[38] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1195–1204, 2017.

[39] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *CoRR*, abs/2005.10243, 2020.

[40] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[41] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 1096–1103, 2008.

[42] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 8933–8940. AAAI Press, 2019.

[43] Guangcong Wang, Jian-Huang Lai, Wenqi Liang, and Guangrun Wang. Smoothing adversarial domain attack and p-memory reconsolidation for cross-domain person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10568–10577, 2020.

[44] Guangrun Wang, Liang Lin, Shengyong Ding, Ya Li, and Qing Wang. DARI: distance metric and representation integration for person verification. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3611–3617. AAAI Press, 2016.

[45] Guangrun Wang, Guangcong Wang, Keze Wang, Xiaodan Liang, and Liang Lin. Grammatically recognizing images with tree convolution. In Rajesh Gupta, Yan Liu, Jiliang

Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 903–912. ACM, 2020.

[46] Guangrun Wang, Guangcong Wang, Xujie Zhang, Jianhuang Lai, Zhengtao Yu, and Liang Lin. Weakly supervised person re-id: Differentiable graphical learning and a new benchmark. In *IEEE Transactions on Neural Networks and Learning Systems (T-NNLS)*, 2020.

[47] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In Susanne Boll, Kyoung Mu Lee, Jiebo Luo, Wenwu Zhu, Hyeran Byun, Chang Wen Chen, Rainer Lienhart, and Tao Mei, editors, *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, pages 274–282. ACM, 2018.

[48] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R. Scott. Cross-batch memory for embedding learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6387–6396. IEEE, 2020.

[49] Yuxin Wu et al. Tensorpack. `https://github.com/tensorpack/`, 2016.

[50] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *CoRR*, abs/1805.01978, 2018.

[51] Haohang Xu, Xiaopeng Zhang, Hao Li, Lingxi Xie, Hongkai Xiong, and Qi Tian. Hierarchical semantic aggregation for contrastive representation learning. *CoRR*, abs/2012.02733, 2020.

[52] Jiwei Yang, Xu Shen, Xinmei Tian, Houqiang Li, Jianqiang Huang, and Xian-Sheng Hua. Local convolutional neural networks for person re-identification. In Susanne Boll, Kyoung Mu Lee, Jiebo Luo, Wenwu Zhu, Hyeran Byun, Chang Wen Chen, Rainer Lienhart, and Tao Mei, editors, *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, pages 1074–1082. ACM, 2018.

[53] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6687–6696, 2020.

[54] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[55] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, pages 649–666, 2016.

[56] Xiaohang Zhang, Jiahao Xie, Enze Xie, and Ziwei Liu. Openselfsup. `https://github.com/open-mmlab/OpenSelfSup`, 2020.

[57] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(3):653–668, 2013.

[58] Jiaxuan Zhuo, Zeyu Chen, Jianhuang Lai, and Guangcong Wang. Occluded person re-identification. In *2018 IEEE International Conference on Multimedia and Expo, ICME 2018, San Diego, CA, USA, July 23-27, 2018*, pages 1–6. IEEE Computer Society, 2018.