Relative Contrastive Loss for Unsupervised Representation Learning

Shixiang Tang[1,3], Feng Zhu[3], Lei Bai[2,†], Rui Zhao[3,4], and Wanli Ouyang[2,1]

[1] University of Sydney, Australia
[2] Shanghai AI Laboratory, China
[3] Sensetime Research
[4] Qing Yuan Research Institute, Shanghai Jiao Tong University
stan3906@uni.sydney.edu.au, baisanshi@gmail.com † corresponding author

Abstract. Defining positive and negative samples is critical for learning visual variations of the semantic classes in an unsupervised manner. Previous methods either construct positive sample pairs as different data augmentations on the same image (i.e., single-instance-positive) or estimate a class prototype by clustering (i.e., prototype-positive), both ignoring the relative nature of positive/negative concepts in the real world. Motivated by the ability of humans in recognizing relatively positive/negative samples, we propose the Relative Contrastive Loss (RCL) to learn feature representation from relatively positive/negative pairs, which not only learns more real world semantic variations than the single-instance-positive methods but also respects positive-negative relativeness compared with absolute prototype-positive methods. The proposed RCL improves the linear evaluation for MoCo v3 by **+2.0%** on ImageNet.

1 Introduction

Recent progresses on visual representation learning [1, 36, 15, 26, 28, 53, 48, 40] have shown the superior capability of unsupervised learning (also denoted as self-supervised learning in some works [7, 21, 46]) in learning visual representations without manual annotations. Contrastive learning [24, 21, 9, 11, 7, 46, 57, 19, 50, 10], which is the cornerstone of recent unsupervised learning methods, optimizes the deep networks by reducing the distance between representations of positive pairs and increasing the distance between representations of negative pairs in the latent feature space simultaneously. As an amazing achievement, it is shown in [24, 19, 50] that the pretrained feature representation with recent contrastive learning methods is comparable with supervised learning in image classification.

One of the critical components for contrastive learning methods is constructing positive pairs and negative pairs. In particular, *single-instance-positive* methods, such as MoCo [21, 9], SimCLR [7, 8], and BYOL [19], apply random image augmentations (*e.g.,* random crops, color jittering, etc) on the same sample (image) to obtain different views of the same sample as *positive* pairs and optionally take the augmentations of other samples as *negative* pairs. Though demonstrated

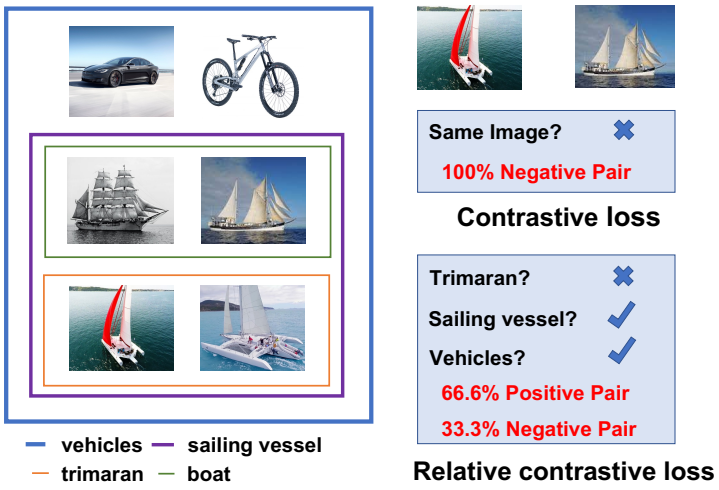

Fig. 1: Motivation of the Relative Contrastive Loss. *Left*: Blue, purple and, orange rectangles denote vehicles, sailing vessel, and trimaran, respectively. The concepts of vehicles, sailing vessels, and trimarans show that the concepts of two images belonging to the same category depend on the level of hyponymy, motivating us to conduct the relative contrastive learning in this paper. *Right:* Any image pair in relative contrastive loss are determined postive or negative by multiple criteria.

effective, such augmentations are insufficient to provide positive pairs with natural intra-class variances such as different camera viewpoints, non-rigid deformations of the same object, or different instances of the same category. *Clustering-based* methods [5, 3] and *neighborhood-based* methods [16, 39] can handle the above problems in *single-instance-positive* methods by using the prototypes of pseudo-classes generated by clustering [5, 3, 2] or k nearest neighbors in the feature space [16, 39] as the positive samples. Despite their great success, all these unsupervised learning methods define positive and negative pairs absolutely, ignoring the relative nature of positive and negative concepts in the real world.

Instead of constructing positive and negative pairs categorically in previous self-supervised learning methods, human beings have the relative recognition ability. In biotaxonomy, the Swedish botanist Carl Linnaeus described the relative similarity of biological organisms under seven hierarchies, *i.e,* Kingdom, Phylum, Class, Order, Family, Genus, Species, forming the current system of Linnaean taxonomy [17]. Popular benchmarks in computer vision such as ImageNet [14], iNat21 [45], and Places365 [56], also respect the positive-negative relativeness and include hierarchical labels. For example, in ImageNet, trimaran and boats all belong to sailing vessels (more general concept) and vehicles (the most general concept in Fig. 1(left)). However, trimaran and boats are different classes when we aim to specify different sailing vessels.

In this paper, we respect the nature of relativeness in human recognition and propose a new *relative contrastive loss* by recognizing a given sample pair

partially positive and negative based on a set of semantic criteria to capture real world instance variation of a class in a relative way (Fig. 1(*right*)). Given two images, *i.e.,* query and key, we feed them into the encoder and momentum encoder to get their features, respectively. Then, the relatively positive-negative relations among them are determined by a set of criteria, which are instantiated by hierarchical clustering. Each level in hierarchical clustering is considered as a specific criterion. The proposed relative contrastive loss leverages the query feature, the key feature and their relatively positive-negative relations to supervise the training process.

In summary, our main contributions are introducing the general idea of relative contrastive loss for self-supervised learning, and accordingly designing a framework incorporating the online hierarchical clustering to instantiate it. The effectiveness of our proposed method is demonstrated via extensive experiments. For instance, on ImageNet linear evaluation, our method well boosts the top-1 accuracy of ResNet-50 by **+2.0%** gain (73.8% $\rightarrow$ 75.8%) compared with MoCov3. Experimental results also validate the effectiveness of our method for semi-supervised classification, object detection, and instance segmentation.

2 Related Work

Single-instance-positive Methods. Instead of designing new pre-text tasks [15, 54, 33, 34, 3], recent unsupervised learning methods are developed upon contrastive learning, which tries to pull the representations of different augmented views of the same sample/instance close and push representations of different instances away [7, 24, 21, 9, 16, 12, 25, 20]. Contrastive methods require to define positive pairs and negative pairs in an absolute way, which violates the relativeness of human recognition. This issue of previous contrastive methods strongly motivates the need for relative-contrastive approaches that can reflect the nature of relativeness when human recognize objects. We achieve this goal by introducing a new relative contrastive loss. Instead of defining positive and negative pairs according to one absolute criterion, we assign a sample pair positive or negative by a set of different criteria to mimic the relative distinguish ability.

Clustering-based Methods. Instead of viewing each sample as an independent class, clustering-based methods group samples into clusters [3, 5, 52, 57]. Along this line, DeepCluster [3] leverages k-means assignments of prior representations as pseudo-labels for the new representations. SwAV [5] learns the clusters online through the Sinkhorm-Knopp transform [27, 6]. Our method is also related to these clustering-based methods in that we instantiate our relative contrastive loss with an online hierarchical clustering. [4] leverages the hierarchical clustering to tackle non-curated data [41], instead of tackling curated data, *i.e.,* ImageNet-1K, in our paper. However, these clustering-based methods define positive and negative pairs absolutely. In our method, a pair of samples can be partially positive, respecting the relativeness of similarity between a pair of samples.

Neighborhood-based Methods. Neighborhood-based methods stand the recent states-of-the-art methods in unsupervised learning. NNCLR [16] replaces

one of the views in single-instance-positive methods with its nearest neighbor in the feature space as the positive sample. MSF [39] makes a further step by using the k nearest neighbors in the feature space as the positive samples. Neighborhood-based methods perform better than single-instance-positive methods because they can capture more class-invariances that can not be defined by augmentations and better than clustering methods because the query and the positive samples are more likely to belong to the same class. Our work also consider neighbors, but in a relative way.

3 Background: Contrastive Learning

Given an input image $\mathbf{x}$, two different augmentation parameters are employed to get two different images/views: image $\mathbf{v}$ and image $\mathbf{v}'$ for the query and the key branch, which output $\mathbf{q} = \mathcal{P}(\mathcal{Q}(\mathbf{v}, \theta), \theta_p)$ and $\mathbf{z}' = \mathcal{K}(\mathbf{v}', \xi)$, respectively. Here, $\mathcal{Q}$ and $\mathcal{K}$ respectively denote feature transformations parameterized by θ and ξ. $\mathcal{P}$ is an optional prediction [10, 19, 36] of $\mathbf{z} = \mathcal{Q}(\mathbf{v}, \theta)$ implemented by MLP. The contrastive loss is presented in InfoNCE [23], *i.e.,*

$$\mathcal{L}_{ctr}(\mathbf{x}, \theta) = -\log \left[\frac{\exp\left(\mathbf{q}^\top \mathbf{z}'\right)/\tau}{\exp\left(\mathbf{q}^\top \mathbf{z}'/\tau\right) + \sum_{k=1}^{K} \exp\left(\mathbf{q}^\top \mathbf{s}_k/\tau\right)} \right], \tag{1}$$

where $\mathcal{S} = \{\mathbf{s}_k | k \in [1, K]\}$ is a support queue storing negative features and $\tau = 0.1$ is the temperature. Contrastive loss pulls the features of the query-key pair $(\mathbf{q}, \mathbf{z}')$ together and pushes features of the query-negative pairs $(\mathbf{q}, \mathbf{s}_k)$ apart.

4 Relative Contrastive Learning

We are interested in defining a query-key pair $(\mathbf{q}, \mathbf{z}')$ positive or negative relatively. Therefore we propose a relative contrastive loss and present an instantiation by online hierarchical clustering method to achieve it. Specifically, we generate a set of semantic criteria $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_H\}$ (H denotes the number of criteria) to define $(\mathbf{q}, \mathbf{z}')$ positive or negative by online hierarchical clustering (Sec. 4.3), and then compute our relative contrastive loss (Sec. 4.1). This loss (defined in Eq. 2) is obtained by aggregating the vanilla contrastive losses in Eq. 1 with $(\mathbf{q}, \mathbf{z}')$ defined as positive or negative by every semantic criterion $\mathcal{M}_i$ in $\mathcal{M}$.

Overview. As shown in Fig. 2, the relative contrastive learning has the following steps.

Step 1: Image $\mathbf{x}$ *to features* $\mathbf{z}$ *and* $\hat{\mathbf{z}}'$. Specifically, given two different views $(\mathbf{v}, \mathbf{v}')$ of an image $\mathbf{x}$, their projections can be computed by $\mathbf{z} = \mathcal{Q}(\mathbf{v}, \theta)$ and $\hat{\mathbf{z}}' = \mathcal{K}(\mathbf{v}', \xi)$. Following [21, 19], the query branch $\mathcal{Q}(*, \theta)$ is a deep model updated by backward propagation, while the key branch $\mathcal{K}(*, \xi)$ is the same deep model as the query branch but with parameters obtained from the moving average of $\mathcal{Q}(*, \theta)$.

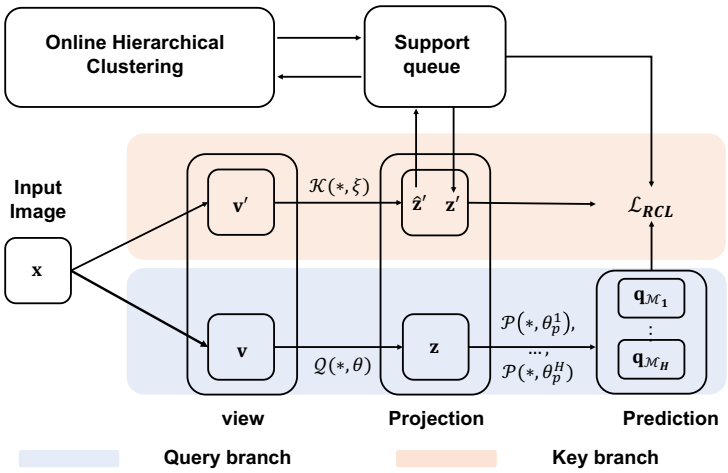

Fig. 2: The pipeline of relative contrastive learning. In the key branch, the feature $\hat{\mathbf{z}}'$ after projection is used to search the relative keys $\mathbf{z}'$from the support queue by hierarchical clustering. For the feature $\mathbf{z}$ after projection in the query branch, we feed it into criterion-specific projectors to generate multiple predictions $\{\mathbf{q}_{\mathcal{M}_1}, \mathbf{q}_{\mathcal{M}_2}, ..., \mathbf{q}_{\mathcal{M}_H}\}$. Multiple predictions, $\mathbf{z}$ and $\mathbf{z}'$ are then fed into the relative contrastive loss $\mathcal{L}_{RCL}$.

Step 2: Key-branch features $\hat{\mathbf{z}}'$ to retrieved features $\mathbf{z}'$. On the key branch, we retrieve key features $\mathbf{z}'$ from the support queue $\mathcal{S}$ with multiple criteria $\mathcal{M}$ implemented by hierarchical clustering (Sec. 4.3). On the query branch, similar to [19, 10], we add criterion-specific predictors $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_H\}$ on $\mathbf{z}$ to get $\{\mathbf{q}_{\mathcal{M}_1}, \mathbf{q}_{\mathcal{M}_2}, ..., \mathbf{q}_{\mathcal{M}_H}\}$.

Step 3: Backpropagation using the relative contrastive loss. The retrieved feature $\mathbf{z}'$, multiple predictions $\{\mathbf{q}_{\mathcal{M}_1}, \mathbf{q}_{\mathcal{M}_2}, ..., \mathbf{q}_{\mathcal{M}_H}\}$, and whether $(\mathbf{z}, \mathbf{z}')$ is positive or negative according to semantic criteria $\mathcal{M}$ (designed by online hierarchical clustering in Sec. 4.3) are then fed into the relative contrastive loss (Eq. 2).

4.1 Relative Contrastive Loss

In the conventional contrastive learning, the positive-negative pairs are defined absolutely, *i.e.*, only augmentations of the same image are considered as positive pair. Motivated by the relative recognition ability of human beings, we introduce a relative contrastive loss to explore the potential of relative positive samples defined in diverse standards.

Semantic criteria for assigning labels. For a set of semantic criteria $\{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_H\}$, relative contrastive loss determines any given query-key pair $(\mathbf{z}, \mathbf{z}')$ as positive or negative based on the criteria $\mathcal{M}_i$ for $i = 1, \ldots, H$. Denote $\mathcal{Y}_i(\mathbf{z})$ and $\mathcal{Y}_i(\mathbf{z}')$ respectively as the labels of $\mathbf{z}$ and $\mathbf{z}'$ generated using criterion $\mathcal{M}_i$. The query-key pair $(\mathbf{z}, \mathbf{z}')$ is defined positive under $\mathcal{M}_i$ if $\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')$, and negative under $\mathcal{M}_i$ if $\mathcal{Y}_i(\mathbf{z}) \neq \mathcal{Y}_i(\mathbf{z}')$. Different from the vanilla contrastive loss

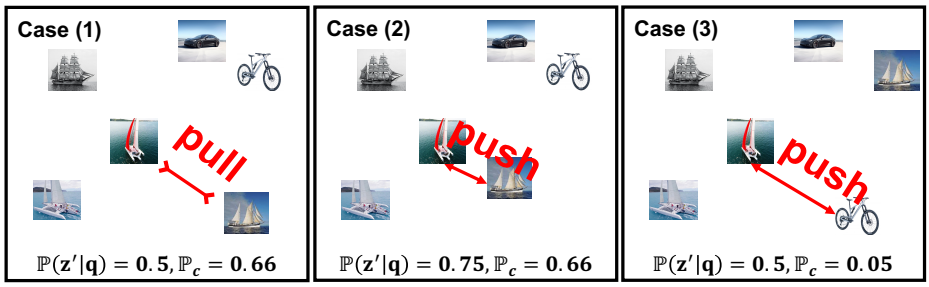

Fig. 3: Analysis of the relative contrastive loss with multiple criteria. Both $\mathbb{P}(\mathbf{z}'|\mathbf{q})$ and $\mathbb{P}_c$ represent the probability that $\mathbf{z}'$ and $\mathbf{q}$ have the same label. The difference is that $\mathbb{P}(\mathbf{z}'|\mathbf{q})$ is based on the cosine similarity of $\mathbf{z}'$ and $\mathbf{q}$, and $\mathbb{P}_c$ is based on the set of defined semantic criteria. Whether to pull $(\mathbf{q}, \mathbf{z})$ together or push $(\mathbf{q}, \mathbf{z})$ apart is determined by $\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c$. If $\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c < 0$, $(\mathbf{q}, \mathbf{z})$ should be pulled together. If $\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c > 0$, $(\mathbf{q}, \mathbf{z})$ should be pushed apart.

in Eq. 1, where $\mathbf{z}$ and $\mathbf{z}'$ are generated by different views of the same sample and naturally a positive pair, the $\mathbf{z}$ and $\mathbf{z}'$ in the relative contrastive loss can be generated by different samples and are considered positive or negative relatively. As an example in Fig. 1, the bicycle and the sailing ship have the same label when the semantic criterion is whether they are vehicles, but they have different labels when the semantic criterion is whether they are sailing vessels.

With the semantic criteria and their corresponding labels defined above, the relative contrastive loss is defined as

$$\mathcal{L}_{RCL}\left(\mathbf{z}, \mathbf{z}', \theta; \{\mathcal{M}_i\}_{i=1}^{H}\right) = \sum_{i=1}^{H} \alpha_i \mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i), \tag{2}$$

where α_i is trade-off parameter among different criteria. $\alpha_i = 1/H$ in our implementation. Loss $\mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i)$ in Eq. 2 for criterion $\mathcal{M}_i$ can be defined as

$$\begin{aligned} &\mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i) \\ &= -\log\left[\frac{\mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right] \cdot \exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) \neq \mathcal{Y}_i(\mathbf{z}')\right]}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{z}'/\tau\right) + \sum_{k=1}^{K}\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top}\mathbf{s}_k/\tau\right)}\right], \end{aligned} \tag{3}$$

where $\mathbf{z} = \mathcal{Q}(\mathbf{v}, \theta)$, $\mathbf{z}' = \mathcal{K}(\mathbf{v}', \xi)$, $\mathbf{s}_k$ is the feature in the support queue $\mathcal{S}$, K is the size of $\mathcal{S}$ and $\mathbb{I}(x)$ is an indication function, $\mathbb{I}(x) = 1$ when x is true, while $\mathbb{I}(x) = 0$ when x is false. $\mathbf{q}_{\mathcal{M}_i} = \mathcal{P}(\mathbf{z}, \theta_p^i)$ is the output of the criterion-specific predictor $\mathcal{P}(*, \theta_p^i)$ for the query projection $\mathbf{z}$, which is explained in the following.

Criterion-specific predictor. Inspired by BYOL [19] and Simsiam [10], the predictor layer aims to predict the expectation of the projection $\mathbf{z}$ under a specific transformation. Therefore, we propose to use the multiple criterion-specific predictors, each of which is to estimate the expectation of $\mathbf{z}$ under its corresponding semantic criterion. Specifically, we add H MLPs, forming predictors $\{\mathcal{P}(*, \theta_p^1), \mathcal{P}(*, \theta_p^2), ..., \mathcal{P}(*, \theta_p^H)\}$ after the projectors in the query branch.

4.2 Analysis of Relative Contrastive Loss

In this section, we mathematically illustrate how relative contrastive loss supervises the feature distance between a query-key sample pair. We will show the feature distance of a image pair with higher possibility of being positive should be smaller than that with lower possibility of being positive.

We derive the gradient of our relative contrastive loss. The gradient of $\mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i)$ in Eq. 3 is

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i)}{\partial \mathbf{z}} &= \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \frac{\partial \mathcal{L}(\mathbf{z}, \mathbf{z}', \theta; \mathcal{M}_i)}{\partial \mathbf{q}_{\mathcal{M}_i}} \\ &= \left(\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right]\right) \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \frac{\mathbf{z}'}{\tau} \\ &+ \sum_{k=1}^{K} \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right) \frac{\mathbf{s}_k}{\tau},\end{aligned} \tag{4}$$

where

$$\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right) = \frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top} \mathbf{z}'/\tau\right)}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top} \mathbf{z}'/\tau\right) + \sum_{k=1}^{K} \exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top} \mathbf{s}_k/\tau\right)}, \tag{5}$$

$$\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right) = \frac{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top} \mathbf{s}_k/\tau\right)}{\exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top} \mathbf{z}')/\tau\right) + \sum_{k=1}^{K} \exp\left(\mathbf{q}_{\mathcal{M}_i}^{\top} \mathbf{s}_k/\tau\right)}. \tag{6}$$

The $\mathbb{P}\left(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}\right)$ and $\mathbb{P}\left(\mathbf{s}_k|\mathbf{q}_{\mathcal{M}_i}\right)$ above are the conditional probabilities of assigning the query prediction $\mathbf{q}_{\mathcal{M}_i}$ to the label of projection $\mathbf{z}'$ and the label of negative samples $\mathbf{s}_k$. We skip the analysis to the query-negative pair $(\mathbf{z}, \mathbf{s}_k)$ and focus on analyzing the dynamics between query-key pair $(\mathbf{z}, \mathbf{z}')$. Therefore, we drop the terms $(\mathbf{q}_{\mathcal{M}_i}, \mathbf{s}_k)$ in Eq. 4. When the gradient above for $\mathcal{L}$ is considered for the loss $\mathcal{L}_{RCL}$ defined in Eq. 2, $\mathbf{z}$ is optimized by gradient descent with the learning rate γ as

$$\mathbf{z} \leftarrow \mathbf{z} - \underbrace{\frac{\gamma}{\tau} \sum_{i=1}^{H} \alpha_i \frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}} \left(\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right]\right) \mathbf{z}'}_{\eta}. \tag{7}$$

When $\eta > 0$, $\mathbf{z}$ and $\mathbf{z}'$ will be pushed apart, and when $\eta < 0$, $\mathbf{z}$ and $\mathbf{z}'$ will be pulled together. Following [44], we assume that $\frac{\partial \mathbf{q}_{\mathcal{M}_i}}{\partial \mathbf{z}}$ is positive definite. Because γ, τ and α_i are positive, we define

$$\eta' = \sum_{i=1}^{H} \left(\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i}) - \mathbb{I}\left[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')\right]\right), \tag{8}$$

which is the only term that determines the sign of η.

In the following, we focus on η' for analyzing the dynamics of relative contrastive loss on network optimization in Eq. 7. We will reveal that the relativeness of positive-negative samples is based on 1) the probability $\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_i})$ of assigning the query prediction $\mathbf{q}_{\mathcal{M}_i}$ to the label of projection $\mathbf{z}'$, and 2) the constructed criteria that determines the labeling function $\mathcal{Y}_i(\cdot)$.

Single Criterion. When there is only one criterion for determining query-key pairs positive or negative, *i.e.,* $\eta' = (\mathbb{P}(\mathbf{z}'|\mathbf{q}_{\mathcal{M}_1}) - \mathbb{I}[\mathcal{Y}_1(\mathbf{z}) = \mathcal{Y}_1(\mathbf{z}')])$, our method collapses to the typical contrastive loss which pulls positive pairs close ($\mathbb{I}[\mathcal{Y}_1(\mathbf{z})=\mathcal{Y}_1(\mathbf{z}')] = 1, \eta' < 0$) and pushes negative pairs apart ($\mathbb{I}[\mathcal{Y}_1(\mathbf{z}) = \mathcal{Y}_1(\mathbf{z}')] = 0$ and $\eta' > 0$).

Multiple Criteria. When there are multiple criteria, to facilitate analysis, we assume the criterion-specific predictors are identical $\mathcal{P}_i = \mathcal{P}, i \leq H$ and thus predictions $\mathbf{q}_{\mathcal{M}_i} = \mathbf{q}, i \leq H$ are the same. With these assumptions, Eq. 8 is modified as

$$\eta' = H(\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c), \tag{9}$$

where $\mathbb{P}_c = \sum_{i=1}^{H} \mathbb{I}[\mathcal{Y}_i(\mathbf{z}) = \mathcal{Y}_i(\mathbf{z}')]/H$ is possibility of $(\mathbf{z}, \mathbf{z}')$ being labeled by the H criteria as positive pair. We show the difference between the probability $\mathbb{P}_c$ define by the criteria and the probability $\mathbb{P}(\mathbf{z}'|\mathbf{q})$ estimated from the model, *i.e.,* $\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c$, will adaptively determine the relative decision of pushing or pulling. We use three different cases for illustration (Fig. 3). (1) $\mathbb{P}(\mathbf{z}'|\mathbf{q}) = 0.50$ and $\mathbb{P}_c = 0.66$; (2) $\mathbb{P}(\mathbf{z}'|\mathbf{q}) = 0.75$ and $\mathbb{P}_c = 0.66$; (3) $\mathbb{P}(\mathbf{z}'|\mathbf{q}) = 0.50$ and $\mathbb{P}_c = 0.05$. **In case (1)**, $\mathbb{P}_c$ is large, *i.e.* most of the criteria label two samples as belonging to the same class. But $\mathbb{P}(\mathbf{z}'|\mathbf{q}) = 0.5$, *i.e.* the probability estimated from the learned features for $\mathbf{z}$ and $\mathbf{z}'$ belonging to the same class is not so high. In this case, because the term $\eta' = H(\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c)$ is negative, gradient descent will pull $\mathbf{z}$ towards $\mathbf{z}'$. **In case (2)**, since $\eta' = H(\mathbb{P}(\mathbf{z}'|\mathbf{q}) - \mathbb{P}_c) > 0$, the loss will pull $\mathbf{z}$ and $\mathbf{z}'$ together. Comparing cases (1) and (2), the loss changes its behavior from pushing samples away to pulling together because of the change of $\mathbb{P}(\mathbf{z}'|\mathbf{q})$. Cases (1) and (3) have the same estimated probability $\mathbb{P}(\mathbf{z}'|\mathbf{q})$. **In case (3)**, most of the criteria label the two samples as not belonging to the same class, *i.e.* $\mathbb{P}_c = 0.05$, and the loss will push $\mathbf{z}$ and $\mathbf{z}'$ away. Comparing cases (1) and (3), if the probability $\mathbb{P}_c$ defined by the criteria changes from high to low, the loss changes its behavior from pulling feature close to pushing features away.

4.3 Criteria Generation

In this section, we introduce an implementation of the semantic criteria $\mathcal{M}_{1:H} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_H\}$ used in the relative contrastive loss, where $\mathcal{M}_h$ is used for defining query-key pair $(\mathbf{z}, \mathbf{z}')$ to be positive or negative. The criteria are implemented by online hierarchical clustering, which constrains the relativeness among different criteria with a hierarchy relationship, *i.e.,* $\mathcal{M}_1 \subset \mathcal{M}_2 \subset ... \subset \mathcal{M}_H$ (if $\mathcal{Y}_h(\mathbf{x}) = \mathcal{Y}_h(\mathbf{x}')$, then $\mathcal{Y}_j(\mathbf{x}) = \mathcal{Y}_j(\mathbf{x}')$, $\forall j > h$). At hierarchical clustering level h, a query-key pair $(\mathbf{x}, \mathbf{x}')$ in the same cluster are consider to be positive pair, $\mathcal{Y}_h(\mathbf{x}) = \mathcal{Y}_h(\mathbf{x}')$. Inspired by [55], the implementation of hierarchical clustering is required to conform with the following property.

Cluster preserve property: samples in the same cluster at the low level are also in the same cluster at higher levels.

There are two stages in the online hierarchical clustering: 1) warm-up stage to obtain the initial clustering results, 2) online cluster refinement stage along with feature learning.

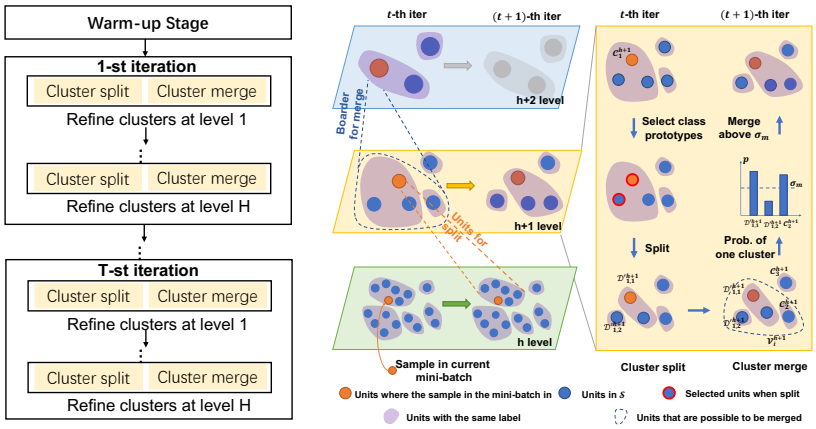

Fig. 4: Online hierarchical clustering. The label refinement at $(h+1)$-th level from the t-th to the $(t+1)$-th iteration is constrained by labels at h-th level and $(h+2)$-th level. The clusters at the h-th level are the basic units for cluster split at the $(h+1)$-th level, and the clusters at the $(h+2)$-th level provides a boarder to identify clusters at the $(h+1)$-th level that may be merged.

Warm-up Stage. Following other clustering-based methods [57], we train our model with the contrastive loss in Eq.1 for 10 epochs. Then, the extracted features of all samples in the dataset are clustered by DBSCAN [18] to obtain initial clusters in level 2 to H. We use each sample as a cluster at level 1.

Online Cluster Refinement Stage. Initial clusters are not accurate due to the poor representations, and therefore need to be progressively adjusted along with the feature optimization. As illustrated in Fig. 4, for each training iteration t, the cluster refinement is conducted from the bottom to the top level, where a cluster contains the most samples. We take i-th cluster $\mathcal{C}_i^{h+1}$ at $(h+1)$-th level to elaborate the process of cluster split and merge.

Cluster Split. Cluster split aims to divide a cluster $\mathcal{C}_i^{h+1}$ into several smaller but more accurate clusters. To conform with the cluster preserve property, the basic units considered for splitting $\mathcal{C}_i^{h+1}$ are clusters in h-level whose samples all belong to $\mathcal{C}_i^{h+1}$, *i.e*, $\mathcal{U}_i^{h+1} = \{\mathcal{C}_j^h | \mathcal{C}_j^h \subset \mathcal{C}_i^{h+1}, j = 1, \ldots, k^h\}$, where k^h is the number of clusters in $\mathcal{H}_h$. Each unit in $\mathcal{U}_i^{h+1}$ is a cluster. When splitting $\mathcal{C}_i^{h+1}$ into m smaller clusters ($m < k^h$), m most dissimilar split units in $\mathcal{U}_i^{h+1}$ are selected using the density peak selection algorithm [37] as the prototype of m different clusters, each of which contains one selected unit. The remaining units in $\mathcal{U}_i$ are merged to the m clusters according to their nearest prototype or label propagation [31] (detailed in supplementary materials). With this procedure, cluster $\mathcal{C}_i^{h+1}$ is split into a set containing m divided clusters, denoted by $\mathcal{D}_i^{h+1} = \{\mathcal{D}'^{h+1}_{i,j}\}_{j=1}^m$.

Cluster Merge. Cluster merge aims to merge the divided clusters $\mathcal{D}_i^{h+1}$ and clusters at level $h+1$ if they are highly possible to one cluster. To conform with the cluster preserve property, we can only try to merge the clusters belonging to the

same cluster $\mathcal{C}_{pa(i)}^{h+2}$ at the $(h+2)$-th level, where $\mathcal{C}_{pa(i)}^{h+2} \supset \mathcal{C}_i^{h+1}$ (clusters circled by the merge boarder in Fig. 4). Therefore, we construct a set of clusters that may be merged as $\mathcal{V}_i^{h+1} = \left\{\bigcup_j \mathcal{C}_{pa(j)=pa(i)}^{h+1}\right\} \bigcup \mathcal{D}_i^{h+1}$, and all elements in $\mathcal{V}_i^{h+1}$ belong to the same cluster $\mathcal{C}_{pa(i)}^{h+2}$. As shown in Fig. 4(Cluster merge), to merge clusters in $\mathcal{V}_i^{h+1}$, we compute the possibility of two clusters belonging to the same class, *i.e.,* according to the distance of cluster centers or label propagation [23] (in supplementary materials). Clusters whose possibilities of belonging to the same cluster are larger than a hyper-parameter σ_m will be merged.

5 Experiment

5.1 Implementation Details

Architecture. Our architecture is similar to MoCo-v2 and MoCo-v3. Compared with MoCo-v2, we use the symmetric loss proposed in BYOL [19] and add predictors after the projector in the query branch. Compared with MoCo-v3, we construct a negative queue as MoCo-v2. Specifically, we use ResNet-50 as our encoder following the common implementations in self-supervised learning literature. We spatially average the output of ResNet-50 which makes the output of the encoder a 2048-dimensional embedding. The projection MLP is composed of 3 fully connected layers having output sizes $[2048, 2048, d]$, where d is the feature dimension applied in the loss, $d = 256$ if not specified. The projection MLP is *fc*-BN-ReLU for the first two MLP layers, and *fc*-BN for the last MLP layer. The architecture of the MLP predictor is 2 fully-connected layers of output size $[4096, d]$, which can be formulated as $fc_2(ReLU(BN(fc_1)))$.
Training. For fair comparison, we train our relative contrastive learning method on the ImageNet2012 dataset [14] which contains 1,281,167 images without using any annotation or class label. In the training stage, we train for 200, 400 and 800 epochs with a warm-up of 10 epochs and cosine annealing schedule using the LARS optimizer [49] by the relative contrastive loss Eq. 2. The base learning rate is set to 0.3. Weight decay of 10^{-6} is applied during training. As is common practice, we do not use weight decay on the bias. The training settings above are the same as BYOL. We also use the same data augmentation scheme as BYOL. For loss computation, we set temperature τ in Eq. 2 to 0.1.

5.2 Comparison with State-of-the-art Methods

Linear Evaluations. Following the standard linear evaluation protocol [46, 57, 21, 9], we train a linear classifier for 90 epochs on the frozen 2048-dimensional embeddings from the ResNet-50 encoder using LARS [49] with cosine annealed learning rate of 1 with Nesterov momentum of 0.9 and batch size of 4096. Comparison with state-of-the-art methods is presented in Tab. 1. Firstly, our proposed RCL achieves better performance compared to other state-of-the-art methods using a ResNet-50 encoder without multi-crop augmentations. Specifically, RCL improves MoCo v2 by 4.7% and MoCo v3 by 2.0%, which generates

Method	Arch.	epochs	Top1	Top5	Method	Arch.	epochs	Top1	Top5
ODC [52]	R50	100	57.6	-	PIRL [32]	R50	800	63.6	-
InstDisc [46]	R50	200	58.5	-	MoCo v2 [9]	R50	800	71.1	-
LocalAgg [57]	R50	200	58.8	-	SimSiam [10]	R50	800	71.3	90.7
MSF [39]	R50	200	71.4	-	SimCLR [7]	R50	800	69.3	89.0
MSF w/s [39]	R50	200	72.4	-	SwAV [5]	R50	800	71.8	-
CPC v2 [22]	R50	200	63.8	85.3	BYOL [19]	R50	1000	74.3	91.6
CMC [42]	R50	240	66.2	87.0	InfoMin Aug. [43]	R50	800	73.0	91.1
Adco [36]	R50	200	68.6	-	MoCo v3 [11]	R50	800	73.8	-
NNCLR [16]	R50	200	70.7	-	NNCLR [16]	R50	800	75.4	92.4
RCL (Ours)	R50	200	72.6	90.8	RCL (Ours)	R50	800	75.8	92.6

Table 1: Comparison with other self-supervised learning methods under the linear evaluation protocol [21] on ImageNet. We omit the result for SwAV with multi-crop for fair comparion with other methods.

Method	ImageNet 1%		ImageNet 10%	
	Top1	Top5	Top1	Top5
Supervised baseline [51]	25.4	48.4	56.4	80.4
Pseudo label [29]	-	-	51.6	82.4
UDA [47]	-	-	68.8†	88.5†
FixMatch [38]	-	-	71.5†	89.1†
MPL [35]	-	73.5†	-	-
InstDisc [46]	-	39.2	-	77.4
PCL [30]	-	75.6	-	86.2
SimCLR [7]	48.3	75.5	65.6	87.8
BYOL [19]	53.2	78.4	68.8	89.0
SwAV (multicrop) [5]	53.9	78.5	70.2	89.9
Barlow Twins [50]	55.0	79.2	69.7	89.3
NNCLR [16]	56.4	80.7	69.8	89.3
RCL (Ours)	57.2	81.0	70.3	89.9

Table 2: Comparison with the state-of-the-art methods for semi-supervised learning. Pseudo Label, UDA, FixMatch and MPL are semi-supervised learning methods. † denotes using random augment [13]. We use the same subset as in SwAV.

positive samples by implementing a different augmentation on the query image. Furthermore, our method is better than InfoMin Aug., which carefully designs the "good view" in the contrastive learning for providing positive samples, by 2.8%. The significant improvements empirically verifies one of our motivation that manually designed augmentations cannot cover the visual variations in a semantic class. Compared with other state-of-the-art methods, our method also achieves higher performance than BYOL by 1.5%. Clustering-based methods, *e.g.,* SwAV [5], and nearest-neighbor-based methods go beyond *single positives*. Clustering-based methods utilize the cluster prototypes as the positive samples. However, our method also achieves 4.0% improvement without the multi-crop augmentation. SwAV leverages an online clustering algorithm and uses only its cluster centers as its positives, which ignores the relative proximity built by our relative contrastive loss. NNCLR [16] is the recent states-of-the-art method, which utilizes the nearest neighbor as the positive sample. Our method is better than NNCLR at 200 epochs are comparable at 800 epochs, because NNCLR

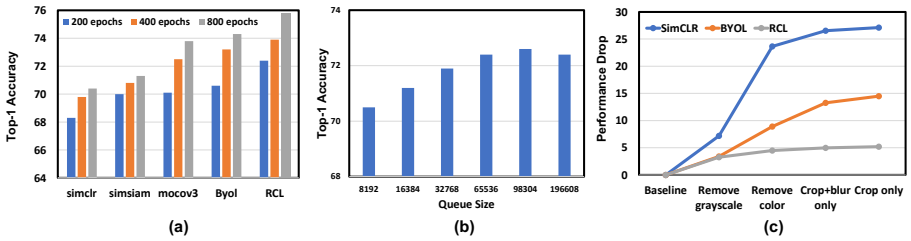

Fig. 5: Ablation studies. (a) Comparison with state-of-the-art methods when training 200, 400 and 800 epochs under linear evaluation on ImageNet. (b) ImageNet top-1 accuracy with different sizes of the support queue. (c) Top-1 Accuracy drop (Y-axis) by removing augmentations (X-axis).

defines positive samples without relativeness. Furthermore, our RCL can be the same as NNCLR when we set only one criterion and only cluster the nearest neighbor. We also compare our method with existing methods in various epochs, which is presented in Fig 5 (a). Our method achieves better performance than SimCLR, Simsiam, MoCo-v3 and BYOL for 200, 400, 800 epochs.

Semi-Supervised Learning Evaluations. To further evaluate the effectiveness of the learned features, we conduct experiments in a semi-supervised setting on ImageNet following the standard evaluation protocol [8, 7], thus fine-tuning the whole base network on 1% or 10% ImageNet data with labels without regularization after unsupervised pre-training. The experimental results are presented in Tab. 2. Firstly, our method outperforms all the compared self-supervised learning methods with the semi-supervised learning setting on ImageNet 1% subset, even when compared with the SwAV method with strong multi-crop augmentation (our RCL does not use multi-crop augmentation). Second, in the ImageNet 10% setting, our method still leads to a better result than most popular self-supervised learning methods, such as SimCLR, BYOL, Barlow Twins and NNCLR. The results indicate the good generalization ability of the features learned by our relative contrastive loss.

5.3 Ablation Study

Default Settings. The size of the support set $\mathcal{S}$ is set to be 1.5×2^{16} and the batch size of our algorithm is 4096. We train for 200 epochs with a warm-up of 10 epochs. The learning rate is 0.3 and we leverage cosine annealing schedule using the LARS optimizer [49]. The results in this section are tested by linear evaluations on ImageNet.

Different Clustering Methods. To illustrate the effectiveness of our online hierarchical clustering method, we compare it with K-means and DBSCAN. Because both K-means and DBSCAN are offline clustering methods, we extract the features of all images in ImageNet-1K, and conduct clustering on these features before each epoch. For K-means, we set the number of clusters to (250000,

No.	#Predictors	#Hierarchies	Top1
1	1	1	70.2
2	1	2	71.3
3	1	3	71.8
4	1	4	71.4
5	3	3	72.6

Table 3: Ablation studies on multiple predictions and the number of levels in the hierarchical clustering. #Predictors: number of criterion-specific predictors. #Hierarchies: number of levels in the hierarchical clustering.

Method	Hierarchy	Online	Top1	Time / Ep
No (NNCLR [16])	x	x	70.7	659s
K-means	low	x	71.8	1056s
DBSCAN	high	x	72.3	986s
Online Hierarchical Clustering	✓	✓	72.6	776s

Table 4: Ablation studies on different clustering methods. Mixed precision time for training 1 epoch using 64 GeForce GTX 1080 Tis with 64 samples in each GPU is reported.

500000, 1000000), where we verify there are about 73.88% samples that conform the hierarchy in Sec. 4.3. For DBSCAN, we keep the minimum number of samples within r to 4, and select $r = 0.8, 0.7, 0.6$ to construct hierarchical label banks, leading to 97.3% samples conforming the hierarchy. As shown in Tab. 4, we can see that K-means improves the NNCLR by 1.1%, which verifies the effectiveness of relativeness. DBSCAN is better than K-means, which verifies the effectiveness of the hierarchical labels. Our online hierarchical clustering is better than above methods, because it can refine labels along with network optimization, which avoids the problem that label refinement is slower than network optimization when using offline clustering. Our online hierarchical clustering is faster than offline clustering algorithms, *e.g.,* kmeans and DBSCAN, because it only deals with samples in the current mini-batch while kmeans and DBSCAN needs to operate on the whole dataset. Compared with NNCLR, our method is about 18% slower, but shows better performance on 200 epochs setting.

Number of Levels in Online Hierarchical Clustering. To assess the effectiveness of relativeness, we ablate on different number of levels in the hierarchical label bank. As illustrated in Tab. 3, the top-1 accuracy improves from 70.2% to 71.3% by 1.1% when we change the number of levels, which indicates the adding relativeness can benefit the contrastive learning in self-supervised image classification tasks. When we continue to increase the number of levels, we can see the top-1 accuracy improves by 0.5% from 2 levels to 3 levels, but will decrease to 71.4% when we changes 3 levels to 4 levels. This phenomenon motivates us to design more appropriate criteria as the future work when implementing relative contrastive loss in the feature.

Multiple Predictors. Multiple predictors are used to predict the multiple projection expectations $\{\mathbb{E}_{\mathcal{T}_1}(\mathbf{z}_\theta), \mathbb{E}_{\mathcal{T}_2}(\mathbf{z}_\theta), ..., \mathbb{E}_{\mathcal{T}_H}(\mathbf{z}_\theta)\}$ based on the various image

transformations that wilBl not change the label under different criteria. When implementing a single predictor after the projection, we actually impose to predict the expectation of the projection regardless of the semantic criterion. When using multiple predictors, we impose each predictor to predict the projection expectation based on the image transformation that will not change the label under a specific criterion. Comparing Exp. 3 and Exp. 5 in Tab. 3, we can conclude that multiple predictors can outperform single predictor by 0.7%.

Size of Support Queue. Similar with MoCo that utilizes a memory bank to store the representations of other samples, our method has a support queue to provide diverse image variations. We evaluate the performance of our method with different support queue size in Fig. 5(b). As can be observed, when the size of the support queue increases to 98304, the performance of our method also improves, reflecting the importance of using more diverse variation as positive samples. Specifically, increasing the size from 65536 to 98304 leads to 0.36% top-1 accuracy improvement. However, further increasing the size of the support queue does not provide further improvement.

Sensitivity to Augmentations. Previous methods leverage the manually designed augmentations to model the visual variation between a semantic class, and therefore augmentations are very critical to their self-supervised learning methods. In contrast, we utilize similar samples/images in the dataset to be positive samples. As illustrated in Fig. 5(c), Our proposed RCL is much less sensitive to image augmentations when compared with SimCLR and BYOL.

6 Limitations and Conclusions

In this paper, we propose a new relative contrastive loss for unsupervised learning. Different from typical contrastive loss that defines query-key pair to be absolutely positive or negative, relative contrastive loss can treat a query-key pair relatively positive, which is measured by a set of semantic criteria. The semantic criteria are instantiated by an online hierarchical clustering in our paper. Representations learnt by the relative contrastive loss can capture diverse semantic criteria, which is motivated by human recognition and fit the relationship among samples better. Extensive results on self-supervised learning, semi-supervised learning and transfer learning settings show the effectiveness of our relative contrastive loss. While our relative loss largely benefits from multiple criteria, the optimal criteria design is still under-explored.

7 Acknowledgement

This work was supported by the Australian Research Council Grant DP200103223, Australian Medical Research Future Fund MRFAI000085, CRC-P Smart Material Recovery Facility (SMRF) – Curby Soft Plastics, and CRC-P ARIA - Bionic Visual-Spatial Prosthesis for the Blind.

References

1. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: Proceedings of the IEEE international conference on computer vision. pp. 37–45 (2015)
2. Asano, Y.M., Rupprecht, C., Vedaldi, A.: Self-labelling via simultaneous clustering and representation learning. arXiv preprint arXiv:1911.05371 (2019)
3. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 132–149 (2018)
4. Caron, M., Bojanowski, P., Mairal, J., Joulin, A.: Unsupervised pre-training of image features on non-curated data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2959–2968 (2019)
5. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882 (2020)
6. Chakrabarty, D., Khanna, S.: Better and simpler error analysis of the sinkhorn–knopp algorithm for matrix scaling. Mathematical Programming (2020)
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
8. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029 (2020)
9. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
10. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)
11. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised visual transformers. arXiv e-prints pp. arXiv–2104 (2021)
12. Chen, X., Chen, W., Chen, T., Yuan, Y., Gong, C., Chen, K., Wang, Z.: Self-pu: Self boosted and calibrated positive-unlabeled training. In: International Conference on Machine Learning. pp. 1510–1519. PMLR (2020)
13. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 702–703 (2020)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
15. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision. pp. 1422–1430 (2015)
16. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. arXiv preprint arXiv:2104.14548 (2021)
17. Ereshefsky, M.: The poverty of the Linnaean hierarchy: A philosophical study of biological taxonomy. Cambridge University Press (2000)
18. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)

19. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
20. Han, T., Xie, W., Zisserman, A.: Self-supervised co-training for video representation learning. arXiv preprint arXiv:2010.09709 (2020)
21. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
22. Henaff, O.: Data-efficient image recognition with contrastive predictive coding. In: International Conference on Machine Learning. pp. 4182–4192. PMLR (2020)
23. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018)
24. Hu, Q., Wang, X., Hu, W., Qi, G.J.: Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1074–1083 (2021)
25. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies (2021)
26. Kim, D., Cho, D., Yoo, D., Kweon, I.S.: Learning image representations by completing damaged jigsaw puzzles. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 793–802. IEEE (2018)
27. Knight, P.A.: The sinkhorn–knopp algorithm: convergence and applications. SIAM Journal on Matrix Analysis and Applications (2008)
28. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: European conference on computer vision. pp. 577–593. Springer (2016)
29. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3, p. 896 (2013)
30. Li, J., Zhou, P., Xiong, C., Socher, R., Hoi, S.C.: Prototypical contrastive learning of unsupervised representations. arXiv preprint arXiv:2005.04966 (2020)
31. Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y.: Learning to propagate labels: Transductive propagation network for few-shot learning. arXiv preprint arXiv:1805.10002 (2018)
32. Misra, I., Maaten, L.v.d.: Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6707–6717 (2020)
33. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016)
34. Noroozi, M., Pirsiavash, H., Favaro, P.: Representation learning by learning to count. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5898–5906 (2017)
35. Pham, H., Dai, Z., Xie, Q., Le, Q.V.: Meta pseudo labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11557–11568 (2021)
36. Qi, G.J., Zhang, L., Lin, F., Wang, X.: Learning generalized transformation equivariant representations via autoencoding transformations. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)

37. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. science **344**(6191), 1492–1496 (2014)
38. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685 (2020)
39. Soroush Abbasi, K., Tejankar, A., Pirsiavash, H.: Mean shift for self-supervised learning. In: International Conference on Computer Vision (ICCV) (2021)
40. Tang, S., Chen, D., Bai, L., Liu, K., Ge, Y., Ouyang, W.: Mutual crf-gnn for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2329–2339 (2021)
41. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Communications of the ACM **59**(2), 64–73 (2016)
42. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 776–794. Springer (2020)
43. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning. arXiv preprint arXiv:2005.10243 (2020)
44. Tian, Y., Chen, X., Ganguli, S.: Understanding self-supervised learning dynamics without contrastive pairs. arXiv preprint arXiv:2102.06810 (2021)
45. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8769–8778 (2018)
46. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3733–3742 (2018)
47. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848 (2019)
48. Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., Hu, H.: Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16684–16693 (2021)
49. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
50. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. arXiv preprint arXiv:2103.03230 (2021)
51. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: Self-supervised semi-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1476–1485 (2019)
52. Zhan, X., Xie, J., Liu, Z., Ong, Y.S., Loy, C.C.: Online deep clustering for unsupervised representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6688–6697 (2020)
53. Zhang, L., Qi, G.J., Wang, L., Luo, J.: Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2547–2555 (2019)
54. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision. pp. 649–666. Springer (2016)

55. Zheng, Y., Tang, S., Teng, G., Ge, Y., Liu, K., Qin, J., Qi, D., Chen, D.: Online pseudo label generation by hierarchical cluster dynamics for adaptive person re-identification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8371–8381 (2021)
56. Zhou, B., Khosla, A., Lapedriza, A., Torralba, A., Oliva, A.: Places: An image database for deep scene understanding. arXiv preprint arXiv:1610.02055 (2016)
57. Zhuang, C., Zhai, A.L., Yamins, D.: Local aggregation for unsupervised learning of visual embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6002–6012 (2019)