# Delving into Inter-Image Invariance for Unsupervised Visual Representations

**Jiahao Xie**[1], **Xiaohang Zhan**[2], **Ziwei Liu**[2], **Yew Soon Ong**[1,3], **Chen Change Loy**[1]
[1]Nanyang Technological University
[2]The Chinese University of Hong Kong
[3]AI3, A*STAR, Singapore
{jiahao003, asysong, ccloy}@ntu.edu.sg
{zx017, zwliu}@ie.cuhk.edu.hk

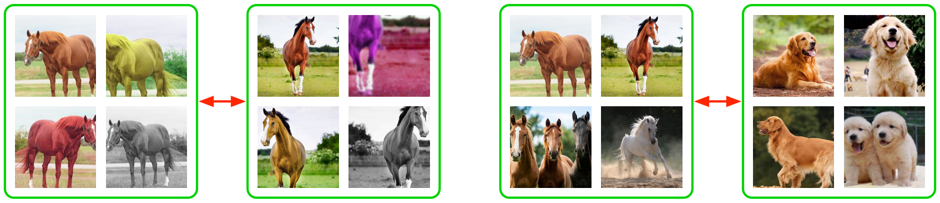
## Abstract

Contrastive learning has recently shown immense potential in unsupervised visual representation learning. Existing studies in this track mainly focus on intra-image invariance learning. The learning typically uses rich intra-image transformations to construct positive pairs and then maximizes agreement using a contrastive loss. The merits of inter-image invariance, conversely, remain much less explored. One major obstacle to exploit inter-image invariance is that it is unclear how to reliably construct inter-image positive pairs, and further derive effective supervision from them since there are no pair annotations available. In this work, we present a rigorous and comprehensive study on inter-image invariance learning from three main constituting components: pseudo-label maintenance, sampling strategy, and decision boundary design. Through carefully-designed comparisons and analysis, we propose a unified framework that supports the integration of unsupervised intra- and inter-image invariance learning. With all the obtained recipes, our final model, namely InterCLR, achieves state-of-the-art performance on standard benchmarks. Code and models will be available at `https://github.com/open-mmlab/OpenSelfSup`.

## 1 Introduction

Unsupervised representation learning witnesses substantial progress in recent years thanks to the emergence of self-supervised learning. Self-supervised learning methods can be broadly divided into four categories: recovery-based [1, 2, 3, 4, 5, 6], transformation prediction [7, 8, 9, 10], clustering-based [11, 12, 13, 14, 15, 16, 17], and contrastive learning [18, 19, 20, 21, 22, 23, 24, 25, 26]. Among the various paradigms, contrastive learning shows great potential and even surpasses supervised learning [24, 26]. A typical contrastive learning method applies rich transformations to an image and maximizes agreement between the original image and the transformed ones, or between the images in two transformations via a contrastive loss in the latent feature space. This process encourages the network to learn "intra-image" invariance (*i.e.*, instance discrimination [19]).

Some typical "intra-image" transformations, including random cropping, rotating, resizing and color distortion, are shown in Figure 1. Clearly, it is challenging to design convincing transformations to faithfully cover all the natural variances existing in natural images. Hence, it remains an open question whether the existing form of transformations can sufficiently lead to our ideal representations, which should be invariant to viewpoints, occlusions, poses, instance-level or subclass-level differences. Such variances naturally exist between pairs of instances belonging to the same semantic class. However, it is challenging to exploit such "inter-image" invariance in the context of unsupervised learning since no pair annotations are available. Clustering is a plausible solution to derive such pseudo-labels for contrastive learning. For instance, the recent study, LA [23], adopted off-the-shelf clustering to obtain pseudo-labels to constitute "inter-image" candidates. Nevertheless, the performance still



Intra-image invariance learning

Inter-image invariance learning

Figure 1: Intra-image invariance learning groups different augmented views of the same image together while separates different images apart. In contrast, inter-image invariance learning groups similar images together while separates dissimilar ones apart.

falls behind state-of-the-art intra-image based methods. We believe there exist details that might have been overlooked, if resolved, shall make the usefulness of inter-image invariance learning more pronounced than it currently does.

In this work, we present our method, InterCLR, an effective and unified framework for unsupervised intra- and inter-image invariance learning. We pay particular attention to the learning of inter-image invariance. Specifically, we perform a comprehensive empirical study on three major aspects:

**1) Pseudo-label maintenance**. Owing to expensive computational cost, offline clustering adopted in DeepCluster [14] and LA [23] can only be performed sparsely every several training epochs. Hence, it inevitably produces stale labels relative to the continuously updated network. To re-assign pseudo-labels continuously and instantly, we propose to use online clustering (*e.g.*, mini-batch $k$-means) in place of offline clustering. We integrate the label and centroid update steps into each training iteration. In this way, clustering and network update are simultaneously undertaken, yielding more reliable pseudo-labels.

**2) Sampling strategy**. It is common for supervised learning to adopt hard negative mining, *i.e.*, selecting the closest negative pairs and push them apart. However, in the scenario of unsupervised learning, hard negatives might well have wrong labels, *i.e.*, they may be actually positive pairs. On the other hand, if we choose easy negative pairs and push them apart, they will still be easy negatives next time, and might never be corrected, leading to a shortcut solution. Hence, the sampling strategy in unsupervised inter-image invariance learning is non-trivial.

**3) Decision boundary design**. Existing works [27, 28, 29, 30, 31] in supervised learning design large-margin loss functions to learn discriminative features. While in unsupervised learning, it is unsure whether pursuing discriminative features benefits since pseudo-labels are noisy. For example, if a positive pair of images are misclassified as a negative one, the large-margin optimization strategy will further push them apart. Then the situation will never be corrected. We explore decision margin designs for both the intra- and inter-image branches.

The main contributions of this work include a unified unsupervised framework for intra- and inter-image invariance learning, as well as in-depth studies in the design of inter-image invariance learning from different aspects. The merits of inter-image invariance learning are revealed through our state-of-the-art performance on standard unsupervised representation learning benchmarks.

## 2 Preliminaries

**Intra-image invariance learning.** A contrastive representation learning method typically learns a neural encoder $f_\theta(*)$ that maps training images $\mathbf{I} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ to compact features $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N\}$ with $\mathbf{v}_i = f_\theta(\mathbf{x}_i)$ in a $D$-dimensional L2-normalized embedding space, where the samples of a positive pair are pulled together and those of negative pairs are pushed apart. For intra-image invariance learning, the positive pair is usually formed with two different augmented views of the same image while the negative pairs are obtained from different images. To achieve this objective, a contrastive loss function is optimized with similarity measured by dot product. Here we consider an effective form of contrastive loss function, called InfoNCE [18], as follows:

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^{N} -\log \frac{\exp\left(\mathbf{v}_i \cdot \mathbf{v}_i^{+}/\tau\right)}{\exp\left(\mathbf{v}_i \cdot \mathbf{v}_i^{+}/\tau\right) + \sum_{\mathbf{v}_i^{-} \in \mathbf{V}_K} \exp\left(\mathbf{v}_i \cdot \mathbf{v}_i^{-}/\tau\right)}, \tag{1}$$

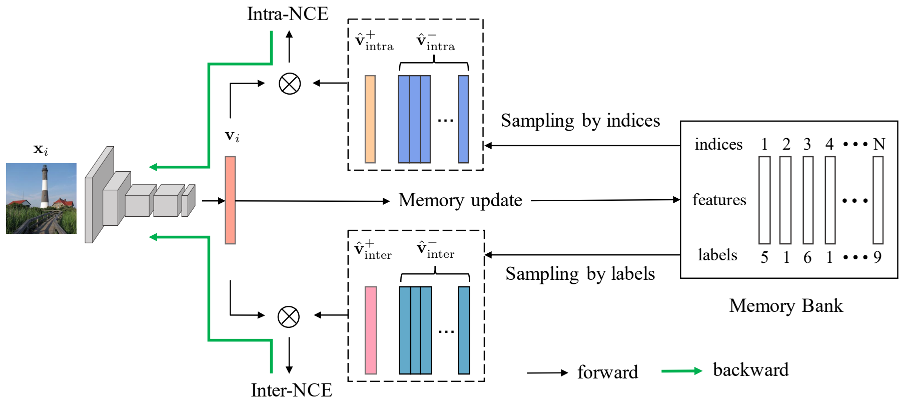


Figure 2: Overview of our unified inta- and inter-image invariance learning framework. For the intra-image component, positive and negative pairs are sampled by indices, while for the inter-image part, they are sampled by pseudo-labels. The memory bank including features and pseudo-labels is updated in each iteration. “Intra-NCE” and “Inter-NCE” constitute loss functions for the two branches respectively.

where $\tau$ is a temperature hyper-parameter, $\mathbf{v}_i^+$ is a positive sample for instance $i$, and $\mathbf{v}_i^- \in \mathbf{V}_K \subseteq \mathbf{V} \setminus \{\mathbf{v}_i\}$ denotes a set of $K$ negative samples randomly drawn from the training images excluding instance $i$.

**Memory bank.** Contrastive learning requires a large number of negative samples to learn good representations [18, 19]. However, the number of negatives is usually limited by the mini-batch size. To address this issue, one can use a memory bank to store running average features of all samples in the dataset computed in previous steps. Formally, let $\hat{\mathbf{V}} = \{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, ..., \hat{\mathbf{v}}_N\}$ denote the stored features in the memory bank, these features are updated by:

$$\hat{\mathbf{v}}_i \leftarrow (1-\omega)\,\hat{\mathbf{v}}_i + \omega\mathbf{v}_i, \tag{2}$$

where $\omega \in (0, 1]$ is a momentum coefficient. With a set of features $\hat{\mathbf{V}}$, we can then replace $\mathbf{V}$ with $\hat{\mathbf{V}}$ in Equation (1) without having to recompute all the features every time.

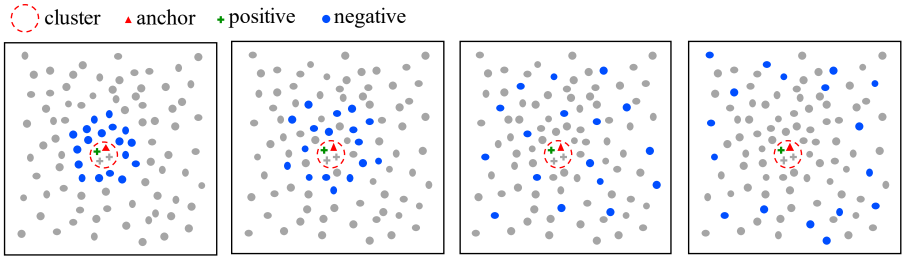
## 3 Methodology

Based on the aforementioned intra-image invariance learning, we describe how to extend the notion to leverage inter-image statistics for contrastive learning.

As shown in Figure 2, we introduce two invariance learning branches in our framework, one for intra-image and the other for inter-image. The intra-image branch draws contrastive pairs by indices following the conventional protocol. The inter-image counterpart constructs contrastive pairs with pseudo-labels obtained by online clustering - a positive sample for an input image is selected within the same cluster while the negative samples are obtained from other clusters. We use variants of InfoNCE described in Section 2 as our contrastive loss and perform back-propagation to update the networks. Within the inter-image branch, three components have non-trivial effects on learned representations and require specific designs, *i.e.*, 1) pseudo-label maintenance, 2) sampling strategy for contrastive pairs, and 3) decision boundary design for the loss function.

### 3.1 Maintaining pseudo-labels

To avoid stale labels from offline clustering, we adopt mini-batch $k$-means to integrate label update into the network update iterations, thus updating the pseudo-labels on-the-fly. Formally, we first initialize all the features, labels and centroids via a global clustering process, *e.g.*, $k$-means. Next, in a mini-batch stochastic gradient descent iteration, the forward batch features are used to update the corresponding stored features in the memory bank with Equation (2). Meanwhile, the label of each involved sample is updated by finding its current nearest centroid following $\min_{\mathbf{y}_i \in \{0,1\}^k,\ \text{s.t.}\ \mathbf{y}_i^{\top}\mathbf{1}=1} \|\hat{\mathbf{v}}_i - \mathbf{C}\mathbf{y}_i\|_2^2$, where $k$ is the number of clusters, $\mathbf{C} \in \mathbb{R}^{d\times k}$ is a recorded



Hard Negative Semi-hard Negative Random Negative Semi-easy Negative

Figure 3: Illustration of different negative sampling strategies in the embedding space. Given an anchor sample (red triangle), the positive sample candidates (crosses) are those points within the cluster represented by the dashed red circle while the negative sample candidates (dots) are the points beyond this cluster. The positive sample (green cross) is drawn randomly from the cluster while the negative samples (blue dots) are drawn with different sampling strategies.

centroid matrix with each column representing a temporary cluster centroid that evolves during training, $\mathbf{y}_i$ is a $k$-dimensional one-hot vector indicating the label assignment for instance $i$. Finally, the recorded centroid matrix is updated by averaging all the features belonging to their respective clusters currently [32]. In this way, labels are updated instantly along with the features.

### 3.2 Sampling contrastive pairs

We define samples sharing the same label with the input image $\mathbf{x}_i$ in the memory bank as positive sample candidates $\mathcal{S}_i^p$, while others as negative sample candidates $\mathcal{S}_i^n$. For positive sampling, we randomly draw one sample from $\mathcal{S}_i^p$ and use it to form a positive pair with $\mathbf{v}_i$. For negative sampling, we sample $K$ negatives from $\mathcal{S}_i^n$. Based on the intuitions in Section 1, we design and compare four sampling strategies for negative samples: hard, semi-hard, random, and semi-easy. Specifically, as shown in a schematic illustration Figure 3, for "hard negative" sampling, we sample $K$ nearest neighbors of $\mathbf{v}_i$ from $\mathcal{S}_i^n$ using cosine distance criterion. For "semi-hard negative" sampling, we first create a relatively larger nearest neighbor pool, *i.e.*, top 10% nearest neighbors from $\mathcal{S}_i^n$, then we randomly draw $K$ samples from this pool. For "random negative" sampling, we simply draw $K$ negative samples at random from $\mathcal{S}_i^n$. For "semi-easy negative" sampling, similar to the "semi-hard negative" strategy, we first sample a pool with top 10% farthest neighbors from $\mathcal{S}_i^n$, then we randomly draw $K$ samples from this pool. We do not include an "easy negative" strategy that chooses the top $K$ easiest negatives. As mentioned before, the easiest samples are more prone to a shortcut solution.

### 3.3 Designing decision boundary

Considering the contrastive loss in Equation (1), since features in the embedding space are L2-normalized, we replace $\mathbf{v}_i \cdot \mathbf{v}_j$ with $\cos(\theta_{\mathbf{v}_i,\mathbf{v}_j})$. For simplicity of analysis, we consider the case where there is only one negative sample, *i.e.*, a binary classification scenario. The contrastive loss thus results in a zero-margin decision boundary given by:

$$\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^+}) = \cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^-}). \tag{3}$$

To allow the decision margins to be more stringent or looser, we first introduce a cosine decision margin $m$ such that the decision boundary becomes:

$$C_+ : \cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^+}) - m \geq \cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^-}), \quad C_- : \cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^-}) - m \geq \cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^+}). \tag{4}$$

As shown in Figure 4, $m > 0$ indicates a more stringent decision boundary that encourages the discriminative ability of the representations, while $m < 0$ stands for a looser decision boundary. Then, we define a margin contrastive loss (MarginNCE) as:

$$\mathcal{L}_{\text{MarginNCE}} = \sum_{i=1}^{N} -\log \frac{\exp\left(\left(\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^+}) - m\right)/\tau\right)}{\exp\left(\left(\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^+}) - m\right)/\tau\right) + \sum_{\mathbf{v}_i^- \in \mathbf{V}_K} \exp\left(\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^-})/\tau\right)}. \tag{5}$$

We make a hypothesis that for the intra-image MarginNCE loss ($\mathcal{L}_{\text{Intra-MarginNCE}}$), the margin should be positive, since the labels derived from image indices are always correct; while for the inter-image

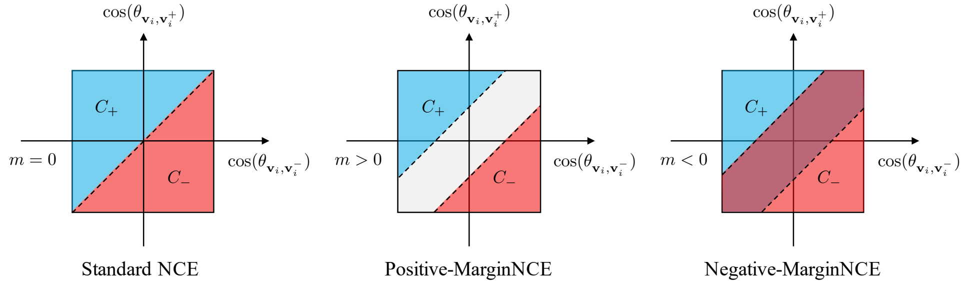

$\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^+})$
$C_+$
$m = 0$
$C_-$
$\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^-})$
Standard NCE
$\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^+})$
$C_+$
$m > 0$
$C_-$
$\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^-})$
Positive-MarginNCE
$\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^+})$
$C_+$
$m < 0$
$C_-$
$\cos(\theta_{\mathbf{v}_i,\mathbf{v}_i^-})$
Negative-MarginNCE

Figure 4: Comparison of different decision margins between standard NCE and MarginNCE under one negative sample case. The dashed line represents the decision boundary and the gray area (shown as wine red when $C_+$ and $C_-$ overlap) shows the decision margins.

MarginNCE loss ($\mathcal{L}_{\text{Inter-MarginNCE}}$), the margin should be negative, since the pseudo-labels are evolving during training and are not accurate enough. The final loss consists of these two MarginNCE loss functions:

$$\mathcal{L}_{\text{Intra-Inter-MarginNCE}} = \lambda \mathcal{L}_{\text{Intra-MarginNCE}} + (1 - \lambda)\, \mathcal{L}_{\text{Inter-MarginNCE}}, \tag{6}$$

where $\lambda$ is a hyper-parameter to balance the Intra-MarginNCE and Inter-MarginNCE loss. We study $m$ for both $\mathcal{L}_{\text{Intra-MarginNCE}}$ and $\mathcal{L}_{\text{Inter-MarginNCE}}$ in Section 5. A study on $\lambda$ is provided in the appendix.

## 4 Related work

**Contrastive-based representation learning.** Contrastive-based methods learn invariant features by contrasting positive samples against negative ones. A positive pair is usually formed with two augmented views of the same image, while negative ones are formed with different images. Typically, the positive and negative samples can be obtained either within a batch or from a memory bank. In batch-wise methods [18, 21, 22, 33, 34, 26], positive and negative samples are drawn from the current mini-batch with the same encoder and the encoder is updated end-to-end with back-propagation. For methods based on memory bank [19, 20, 23, 25], positive and negative samples are drawn from a memory bank that stores features of all samples computed in previous steps. Recently, He *et al.* [24] propose Momentum Contrast (MoCo), a mechanism for building large and consistent dictionaries for contrastive learning using a slowly progressing encoder. As opposed to our work, most of the aforementioned approaches only explore intra-image statistics for contrastive learning. Although LA [23] is the first attempt to leverage inter-image statistics for contrastive learning, it mainly focuses on designing sampling metric while leaving other important aspects unexplored. We show that pseudo-label maintenance, sampling strategy and decision boundary design have to be collectively considered for good results.

**Clustering-based representation learning.** Earlier attempts have shown great potential of joint clustering and feature learning, but the studies are limited to small datasets [12, 13, 35, 36, 37, 38]. DeepCluster [14] (DC) scales up the learning to millions of images through alternating between deep feature clustering and CNN parameters update. Although DC uses clustering during representation learning, it differs from our work conceptually in two aspects. First, DC optimizes the cross-entropy loss between predictions and pseudo-labels obtained by cluster assignments. Such optimization requires an additional parametric classifier. Second, DC adopts offline global clustering that unavoidably permutes label assignments randomly in different epochs. As a result, the classifier has to be frequently reinitialized after each label reassignment, which leads to training instability. In contrast, we optimize a non-parametric classifier at instance level and integrate the label update procedure into each training iteration with online clustering.

## 5 Experiments

### 5.1 Results on standard benchmarks

Following common practice in self-supervised learning [39, 40], we evaluate the quality of the learned representations by transferring to several standard downstream tasks. To ensure fair and

Table 1: Image classification evaluation. We report top-1 center-crop accuracy of fully-connected classifiers for ImageNet and Places205, and mAP of linear SVMs for VOC07 and VOC07$_{\text{lowshot}}$. We show the parameter counts of the feature extractors. Numbers with $^\dagger$: we use the officially released pre-trained model for MoCo, and re-implement SimCLR. All other numbers are taken from the papers as cited. $^*$: SimCLR requires a large batch size of 4096 allocated on 128 TPUs. CMC and AMDIM use FastAutoAugment [41] that is supervised by ImageNet labels. Methods in the last section require larger architectures.

| Method | Arch. | #Params (M) | #Epochs | Transfer Dataset | | | |
|---|---|---|---|---|---|---|---|
| | | | | ImageNet | Places205 | VOC07 | VOC07$_{\text{lowshot}}$ |
| Supervised [40] | R50 | 24 | - | 75.5 | 52.5 | 88.0 | 75.4 |
| Random [40] | R50 | 24 | - | 13.7 | 16.6 | 9.6 | 9.0 |
| Colorization [40] | R50 | 24 | 28 | 39.6 | 37.5 | 55.6 | 33.3 |
| Jigsaw [40] | R50 | 24 | 90 | 45.7 | 41.2 | 64.5 | 39.2 |
| NPID [19] | R50 | 24 | 200 | 54.0 | 45.5 | - | - |
| Rotation [25] | R50 | 24 | - | 48.9 | 41.4 | 63.9 | - |
| BigBiGAN [42] | R50 | 24 | - | 56.6 | - | - | - |
| LA [23] | R50 | 24 | 200 | 58.8 | 49.1 | - | - |
| MoCo [24] | R50 | 24 | 200 | 60.6 | 50.2$^\dagger$ | 79.3$^\dagger$ | 57.9$^\dagger$ |
| SimCLR [26] | R50 | 24 | 200 | 61.9 | 51.6$^\dagger$ | 79.0$^\dagger$ | 58.4$^\dagger$ |
| InterCLR (ours) | R50 | 24 | 200 | **65.5** | **52.2** | **82.6** | **65.5** |
| SeLa [16] | R50 | 24 | 400 | 61.5 | - | - | - |
| ODC [17] | R50 | 24 | 440 | 57.6 | 49.3 | 78.2 | 57.1 |
| PIRL [25] | R50 | 24 | 800 | 63.6 | 49.8 | 81.1 | - |
| SimCLR [26] | R50 | 24 | 1000 | 69.3$^*$ | - | - | - |
| InterCLR (ours) | R50 | 24 | 1000 | **69.6** | **53.4** | **85.7** | **70.0** |
| CPC [18] | R101 | 28 | - | 48.7 | - | - | - |
| CPC v2 [33] | R170$_{\text{wider}}$ | 303 | $\sim$200 | 65.9 | - | - | - |
| CMC [20] | R50$_{\text{L+ab}}$ | 47 | 280 | 64.1$^*$ | - | - | - |
| AMDIM [34] | Custom | 626 | 150 | 68.1$^*$ | 55.0$^*$ | - | - |

direct comparisons with previous methods, we train using a batch size of 256 for 200 epochs. We also report results of a larger batch size (4,096) and longer epochs (1,000) in Table 1 and Table 2 to compare with the large-batch long-epoch results of SimCLR [26]. We provide more details of our experimental settings in the appendix.

**Image classification with linear models.** Following [40, 25], we freeze all the backbone parameters and train classifiers on representations from different depths of the network. For ImageNet [43] and Places205 [44], we train a 1,000-way and 205-way fully-connected classifier respectively. For VOC07 [45], we train linear SVMs following the setting from [40, 25]. Table 1 shows the results for the best-performing layer of each method. InterCLR outperforms previous self-supervised learners that are pre-trained within 200 epochs using a standard ResNet-50 backbone, setting a new state of the art in this fair competition on all three datasets. With 1,000-epoch pre-training, InterCLR again outperforms corresponding SimCLR's result, showing that InterCLR can also benefit from longer training epochs as SimCLR.

**Low-shot image classification.** Next, we explore the quality of learned representations when there are few training examples per category by transferring to the low-shot VOC07 classification task. Specifically, we vary the number of labeled examples in each class and train linear SVMs on the frozen backbone following the same setup in [40]. We report mAP across five independent samples for each low-shot value evaluated on the test split of VOC07. Table 1 shows the final mAP results of different methods obtained with the averages of all low-shot values and all independent runs. InterCLR outperforms the runner-up by a large margin. Figure 5 shows the per-shot results pre-trained within 200 epochs. InterCLR shows consistent improvement over previous methods for each shot, especially on extremely low shots.

**Semi-supervised learning.** We perform semi-supervised learning experiments on ImageNet following the experimental setup of [19, 25]. We randomly select 1% and 10% subsets of the labeled ImageNet training data in a class-balanced way. Then, we fine-tune our models on these two subsets. We report top-5 accuracy on the official ImageNet validation split. As shown in Table 2, InterCLR

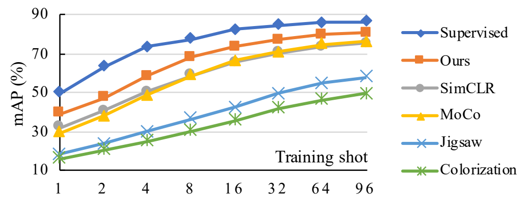


Figure 5: Low-shot image classification on VOC07 with linear SVMs trained on the features from the best-performing layer for ResNet-50. We show the average performance for each shot across five runs. Results for MoCo are from the released pre-trained model. Results for SimCLR are re-implemented by us.

| Method | Backbone | #Epochs | Label fraction | |
|---|---|---|---|---|
| | | | 1% | 10% |
| | | | Top-5 accuracy | |
| *Methods using semi-supervised learning:* | | | | |
| Pseudo-label [46] | R50v2 | - | 51.6 | 82.4 |
| VAT+Entropy Min. [47, 48] | R50v2 | - | 47.0 | 83.4 |
| $S^4$L Exemplar [49] | R50v2 | - | 47.0 | 83.7 |
| $S^4$L Rotation [49] | R50v2 | - | 53.4 | 83.8 |
| *Methods using self-supervised learning only:* | | | | |
| NPID [19] | R50 | 200 | 39.2 | 77.4 |
| Jigsaw [40] | R50 | 90 | 45.3 | 79.3 |
| MoCo [24] | R50 | 200 | $61.3^{\dagger}$ | $84.0^{\dagger}$ |
| SimCLR [26] | R50 | 200 | $64.5^{\dagger}$ | $82.6^{\dagger}$ |
| InterCLR (ours) | R50 | 200 | **66.3** | **84.5** |
| PIRL [25] | R50 | 800 | 57.2 | 83.8 |
| SimCLR [26] | R50 | 1000 | 75.5 | 87.8 |
| InterCLR (ours) | R50 | 1000 | **78.6** | **88.8** |

Table 2: Semi-supervised learning on ImageNet. We report top-5 center-crop accuracy on the ImageNet validation set of self-supervised models that are fine-tuned with 1% and 10% of the labeled ImageNet training data. Numbers with $^{\dagger}$: we use the officially released pre-trained model for MoCo, and re-implement SimCLR. All other numbers are taken from the corresponding papers.

surpasses both self-supervised learners and semi-supervised learners with both 1% and 10% of the labels under either 200 or 1,000 pre-training epochs.

**Object detection.** Following [24], we use Detectron2 [50] to train the Faster-RCNN [51] object detection model with a R50-C4 backbone [52], with BatchNorm tuned. We fine-tune all layers on the trainval split of VOC07+12, and evaluate on the test split of VOC07. We use the same setup for both supervised and self-supervised models. As shown in Table 3, InterCLR surpasses both MoCo and the supervised pre-training counterpart within 200 pre-training epochs.

| Method | Architecture | VOC07+12 |
|---|---|---|
| Random [24] | R50-C4 | 60.2 |
| Supervised [24] | R50-C4 | 81.3 |
| MoCo [24] | R50-C4 | 81.5 (+0.2) |
| InterCLR (ours) | R50-C4 | **81.8 (+0.5)** |

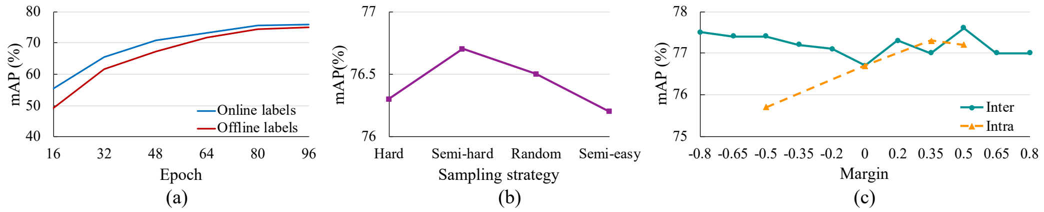
Table 3: Object detection fine-tuned on VOC07+12 using Faster-RCNN. We report $AP_{50}$, the default metric for VOC object detection, averaged across five independent runs. Numbers are taken from the papers as cited. Numbers in the brackets denote the gains to the supervised ImageNet pre-training conterpart.

### 5.2 Study and analysis

We give an in-depth analysis of our design for inter-image invariance learning in this subsection. To perform a large amount of experiments needed for the analysis, we adjust the experimental setting to train each model for 100 epochs with 4,096 negative samples while keeping the other hyper-parameters unchanged. We set $\lambda = 0.5$ in Equation (6) when analyzing the proposed three main features. Unless otherwise specified, we use the VOC07 classification benchmark in Section 5.1 to measure the quality of learned representations.

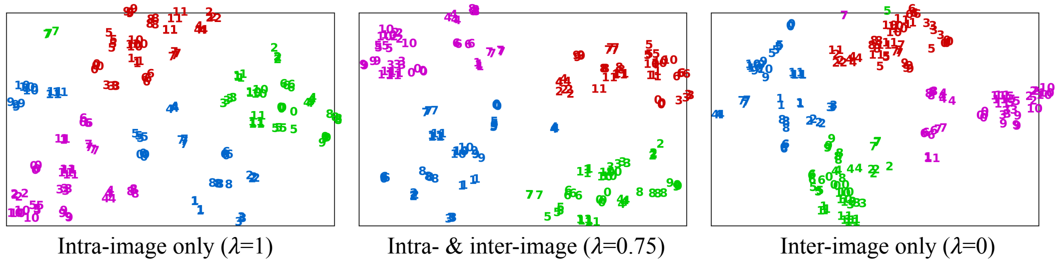
**Online labels are better than offline labels.** We compare our proposed online pseudo-label maintenance against commonly used offline clustering methods. Figure 6(a) shows the results using these two mechanisms. We observe better performance and faster convergence of online labels over offline labels during the training process, suggesting the superiority of maintaining pseudo-labels online.

**Semi-hard negative sampling wins out.** We then study the importance of negative sampling strategies. Figure 6(b) shows the comparison of four negative sampling strategies proposed in



(a) (b) (c)

Figure 6: (a) Comparison between online labels and offline labels. (b) Comparison of different sampling strategies. (c) Effect of decision margin for intra- and inter-image branches.



Intra-image only ($\lambda$=1) Intra- & inter-image ($\lambda$=0.75) Inter-image only ($\lambda$=0)

Figure 7: Feature space visualization. The features are embedded via t-SNE [53] into 2D space. The color indicates ImageNet original classes, and the number indicates different images in each class. Points with the same color and number are the same image in different augmentations.

Section 3.2. Interestingly, we find that semi-hard negative sampling achieves the best performance, while hard negative sampling is even worse than the naïve random sampling strategy. The observation reveals that hard negative mining is not the best choice in the unsupervised learning scenario. In the "Semi-hard" setting, randomly sampled negatives within a relatively larger nearest neighbor pool are more reliable and unbiased.

**Decision margins: positive for "Intra" and negative for "Inter".** We study the impact of decision boundary using different cosine margins for the proposed MarginNCE loss. Specifically, we perform a set of margin experiments for each branch by setting the margin of the other branch as $0$. Figure 6(c) shows the effect of different decision margins. For the intra-image branch, using a positive margin (the best performance is observed when $m = 0.35$) improves the performance upon zero-margin decision boundary, while a negative one degrades the performance. Hence, it is necessary to pursue discriminative features in intra-image invariance learning. However, for the inter-image branch, we find that using both positive and negative margins improves the quality of learned representations. Specifically, when increasing the absolute value of margin, we observe a consistent improvement of performance for negative margins. In contrast, there is a fluctuation in performance improvement for positive margins. Therefore, it is beneficial and low-risk to design a less stringent decision boundary in inter-image invariance learning.

**Analysis on intra- and inter-image invariance.** As shown in Figure 7, the "intra-image only" model merely groups the same image in different augmentations together; however, different images are separated even though in the same class. The "inter-image only" method shortens the distance between images in the same class; however, outliers emerge. The "intra- & inter-image" method well inherits the advantages from above two methods, resulting in a more separable feature space.

## 6 Conclusion

In this work, we have proposed a unified framework, InterCLR, for unsupervised intra- and inter-image invariance learning. With this framework, we delve deep into inter-image invariance learning from different perspectives and show the effect of different design choices, *w.r.t.* pseudo-label maintenance, sampling strategy, and decision boundary design. By combining our observations, we substantially improve over previous contrastive learning methods and achieve state-of-the-art performance on multiple standard benchmarks. Our results demonstrate that leveraging inter-image invariance is critical to unleash the potential of contrastive learning paradigm and push forward unsupervised visual representation learning.

## References

[1] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[2] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[3] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, pages 577–593, 2016.

[4] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[5] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[6] Xiaohang Zhan, Xingang Pan, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised learning via conditional motion propagation. In *CVPR*, 2019.

[7] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, pages 766–774, 2014.

[8] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, pages 4463–4471, 2017.

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[10] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *CVPR*, pages 2547–2555, 2019.

[11] Chen Huang, Chen Change Loy, and Xiaoou Tang. Unsupervised learning of discriminative attributes and visual representations. In *CVPR*, 2016.

[12] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016.

[13] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016.

[14] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[15] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, pages 2959–2968, 2019.

[16] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[17] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *CVPR*, pages 6688–6697, 2020.

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

[19] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018.

[20] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[21] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[22] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, pages 6210–6219, 2019.

[23] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pages 6002–6012, 2019.

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[25] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991*, 2019.

[26] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[27] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, pages 212–220, 2017.

[28] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[29] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, pages 5265–5274, 2018.

[30] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 2018.

[31] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019.

[32] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

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

[34] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, pages 15509–15519, 2019.

[35] Renjie Liao, Alex Schwing, Richard Zemel, and Raquel Urtasun. Learning deep parsimonious representations. In *NeurIPS*, pages 5076–5084, 2016.

[36] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, pages 517–526, 2017.

[37] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *ICCV*, pages 5879–5887, 2017.

[38] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, pages 9865–9874, 2019.

[39] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[40] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, pages 6391–6400, 2019.

[41] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *NeurIPS*, pages 6662–6672, 2019.

[42] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *NeurIPS*, pages 10541–10551, 2019.

[43] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[44] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, pages 487–495, 2014.

[45] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[46] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, 2013.

[47] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, pages 529–536, 2005.

[48] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI*, 41(8):1979–1993, 2018.

[49] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485, 2019.

[50] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.

[51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[52] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[53] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9:2579–2605, 2008.

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[55] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[56] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[57] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

# Appendix

## A Implementation details

**Network architecture.** We use ResNet-50 [54] as our base network. It maps input images into 2,048-dimensional features. Following [26], we add a non-linear projection head (a 2-layer MLP with a 2,048-dimensional hidden layer and ReLU) to project high-dimensional features into a 128-D L2-normalized embedding space. Note that the non-linear head only influences the unsupervised pre-training stage. It is removed when transferring to downstream tasks.

**Data augmentation.** We use ImageNet [43] training set that contains 1.28M images without labels for pre-training. The data augmentation setting is similar to [26], where we extend the data augmentation in [19] by including Gaussian blur. However, we do not use the same heavy color distortion as introduced in [26]. Instead, we only apply a color jittering with a saturation factor in $[0, 2]$, and a hue factor in $[-0.5, 0.5]$.

**Training details for Section 5.1.** For InterCLR with 200 epochs, we train our models using SGD with a momentum of 0.9, a weight decay of $10^{-4}$, and a batch size of 256. We use the cosine learning rate decay schedule [55] with an initial learning rate of $0.03$ and the final learning rate of $3 \times 10^{-5}$. We set the temperature parameter $\tau = 0.1$, the number of negative samples $K = 16,384$, and the momentum coefficient $\omega = 0.5$. We find over-clustering to be beneficial and set the number of clusters as 10,000, which is 10 times of the annotated number of ImageNet classes. Regarding the studies, we use online pseudo-label maintenance, semi-hard negative sampling, and cosine margin $m = -0.5$ for $\mathcal{L}_{\text{Inter-MarginNCE}}$. We set the trade-off hyper-parameter $\lambda = 0.75$ in the final loss (Equation (6) of the main paper). For the experiment with 1,000 epochs, we adopt most of the training hyper-parameters from SimCLR [26]. We use LARS optimizer [56] with a weight decay of $10^{-6}$, and a batch size of 4,096. We use a learning rate of 4.8 ($= 0.3 \times \text{BatchSize}/256$) with linear warmup for the first 10 epochs. After warmup, we use the cosine learning rate decay schedule [55] with the final learning rate of $4.8 \times 10^{-3}$. Other hyper-parameters are exactly the same as InterCLR with 200 epochs.

**Training details for Section 5.2.** We train each model for 100 epochs with 4,096 negative samples while keeping the other hyper-parameters of InterCLR with 200 epochs in Section 5.1 unchanged. We set $\lambda = 0.5$ in Equation (6) and perform a set of experiments progressively when studying the three main aspects. Specifically, for pseudo-label maintenance study, we use random negative sampling and zero-margin decision boundary. For sampling strategy study, we use online pseudo-label maintenance and zero-margin decision boundary. For decision boundary study, we use online pseudo-label maintenance and semi-hard negative sampling.

## B Transfer learning details

**Image classification with linear models.** For ImageNet and Places205, we train linear models on the frozen representations using SGD with a momentum of 0.9. For ImageNet, we train for 100 epochs, with a batch size of 256 and a weight decay of $10^{-4}$. The learning rate is initialized as 0.01, decayed by a factor of 10 after every 30 epochs. For Places205, we train for 28 epochs, with a batch size of 256 and a weight decay of $10^{-4}$. The learning rate is initialized as 0.01, dropped by a factor of 10 at three equally spaced intervals. Other hyper-parameters are set following [40]. We report top-1 center-crop accuracy on the official validation split of ImageNet and Places205. For VOC07, we use the conv5 features (average pooled to around 9,000 dimensions) of ResNet-50. We follow the same setup in [40, 25] and train linear SVMs on the frozen representations using LIBLINEAR package [57]. We train on the trainval split of VOC07 and report mAP on the test split.

**Low-shot image classification.** We use the conv5 features (average pooled to around 9,000 dimensions) of ResNet-50 and train linear SVMs on the frozen representations following the same procedure in [40]. We train on the trainval split of VOC07 and report mAP on the test split for both the final results and the per-shot results in the main paper.

**Semi-supervised learning.** We fine-tune ResNet-50 models using SGD with a momentum of 0.9, and a batch size of 256. We use the 1% and 10% ImageNet subsets specified in the official code release of SimCLR. For both 1% and 10% labeled data, we fine-tune for 20 epochs, with the initial learning rate of backbone set as 0.01 and that of linear classifier as 1. The learning rate is decayed by

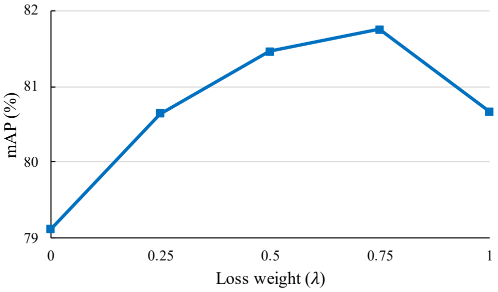


Figure 8: Effect of the trade-off hyper-parameter $\lambda$ on the quality of learned representations. We report mAP of linear SVMs on the VOC07 classification benchmark.

a factor of 5 at 12 and 16 epochs. We use a weight decay of $5 \times 10^{-4}$ for 1% fine-tuning and $10^{-4}$ for 10% fine-tuning.

**Object detection.** We use a batch size of 2 images per GPU, a total of 8 GPUs and fine-tune ResNet-50 models for 24k iterations ($\sim$23 epochs). The learning rate is initialized as 0.02 with a linear warmup for 100 iterations, and decayed by a factor of 10 at 18k and 22k iterations. The image scale is $[480, 800]$ pixels during training and 800 at inference. All hyper-parameters are exactly the same as [24].

## C Ablation on loss weight

The final loss (Equation (6) of the main paper) contains a trade-off hyper-parameter $\lambda$ that controls the weight between two MarginNCE losses. For $\lambda = 1$, our framework degenerates to a typical form of intra-image invariance learning. For $\lambda = 0$, only the inter-image invariance learning branch is retained. When conducting ablation on $\lambda$, we train for 200 epochs with 4,096 negative samples. Besides, we use online pseudo-label maintenance, random negative sampling and zero-margin decision boundary. Other hyper-parameters used are the same as 200-epoch InterCLR in Section 5.1. Figure 8 shows the effect of $\lambda$. InterCLR benefits from the combination of two kinds of image invariance learning, with the best performance obtained by setting $\lambda = 0.75$. Table 4 provides additional transferring results when setting $\lambda = 1$ (intra-image invariance only) and $\lambda = 0.75$ (intra- & inter-image invariance), demonstrating the superiority of our proposed techniques that introduce inter-image invariance learning into intra-image invariance learning.

Table 4: Comparison of transfer learning performance of introducing inter-image invariance learning with intra-image invariance learning baseline.

| | Places205 | VOC07 | VOC07$_{\text{lowshot}}$ | VOC07+12 (detection) |
|---|---|---|---|---|
| Intra ($\lambda = 1$) | 50.3 | 80.7 | 59.8 | 79.7 |
| Intra & Inter ($\lambda = 0.75$) | 51.2 | 81.8 | 63.9 | 80.4 |

## D KNN visualization

Figure 9 shows nearest-neighbor retrieval results using the features learned by InterCLR. The upper three rows show the successful cases where all top 10 results are correctly classified in the same categories as the queries regardless of backgrounds and viewpoints. The lower three rows show the failure cases where none of the top 10 results are in the same categories as the queries. Since no manual annotations are available, it is an extremely hard problem to distinguish between subcategories when the appearances are very similar. However, even in the failure cases, we find that InterCLR can group images according to other contexts present in the queries, *e.g.*, "dog lying down", "dog on the grass" and "bird on the branch", demonstrating the potential of unsupervised learning to discover new semantics beyond human annotations.

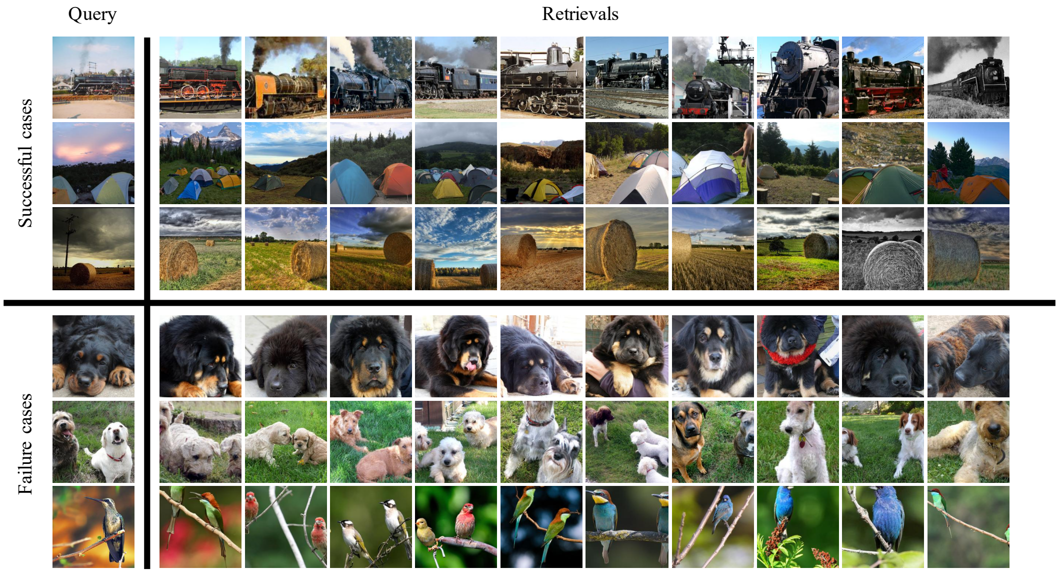


Figure 9: Retrieval results of some example queries on ImageNet. The left-most column are queries from the validation set, while the right columns show 10 nearest neighbors retrieved from the training set. The upper half shows the successful cases, while the lower half shows the failure cases. In the failure cases, even though the retrieved images look visually similar to the queries, they belong to different sub-categories according to the ImageNet manual annotations.