DetCo: Unsupervised Contrastive Learning for Object Detection

Enze Xie[1*] , Jian Ding[3*], Wenhai Wang[4], Xiaohang Zhan[5],
Hang Xu[2], Zhenguo Li[2], Ping Luo[1]
[1]The University of Hong Kong [2]Huawei Noah's Ark Lab
[3]Wuhan University [4]Nanjing University [5]Chinese University of Hong Kong

Abstract

Unsupervised contrastive learning achieves great success in learning image representations with CNN. Unlike most recent methods that focused on improving accuracy of image classification, we present a novel contrastive learning approach, named DetCo, which fully explores the contrasts between global image and local image patches to learn discriminative representations for object detection. DetCo has several appealing benefits. (1) It is carefully designed by investigating the weaknesses of current self-supervised methods, which discard important representations for object detection. (2) DetCo builds hierarchical intermediate contrastive losses between global image and local patches to improve object detection, while maintaining global representations for image recognition. Theoretical analysis shows that the local patches actually remove the contextual information of an image, improving the lower bound of mutual information for better contrastive learning. (3) Extensive experiments on PASCAL VOC, COCO and Cityscapes demonstrate that DetCo not only outperforms state-of-the-art methods on object detection, but also on segmentation, pose estimation, and 3D shape prediction, while it is still competitive on image classification. For example, on PASCAL VOC, DetCo-100ep achieves 57.4 mAP, which is on par with the result of MoCov2-800ep. Moreover, DetCo consistently outperforms supervised method by 1.6/1.2/1.0 AP on Mask RCNN-C4/FPN/RetinaNet with 1x schedule. Code will be released at github.com/xieenze/DetCo *and* github.com/open-mmlab/OpenSelfSup.

1. Introduction

Self-supervised learning of visual representation is an important problem in computer vision, facilitating many downstream tasks such as image classification, object detection, and semantic segmentation [22, 36, 41]. Previous methods [11, 14, 12, 2, 5, 19, 3, 18, 9, 10] focus on designing different pretext tasks. One of the most promising directions among them is contrastive learning [32], which transforms one image into multiple views, and minimize the distance between views from the same image and maximize the distance between views from different images.

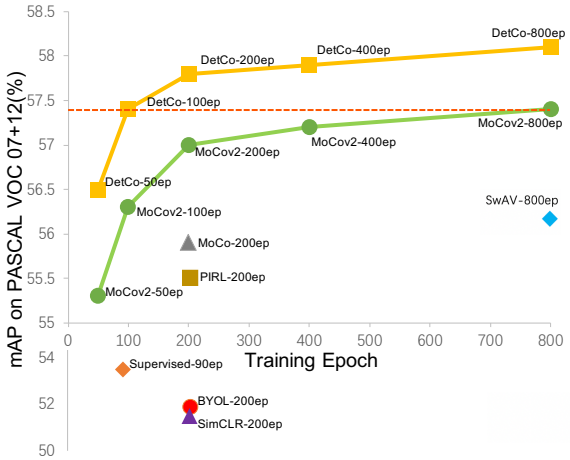


Figure 1. **Comparisons of mAP on PASCAL VOC [15] 07+12 for object detection**. For fair comparisons, all results are evaluated following the setting of MoCo [19]. For different pre-training epochs, we see that DetCo consistently outperforms MoCo v2 [5], which is a strong competitor on VOC compared to many other newly-proposed approaches such as BYOL [18], PIRL [30], and SwAV [3]. For example, DetCo-100ep already achieves similar mAP compared to MoCov2-800ep.

Recent trends in self-supervised contrastive learning mainly design different contrastive pretext tasks in order to bridge the performance gap between unsupervised and fully-supervised methods for image classification, whose accuracy is constantly refreshed by many unsupervised methods such as MoCo v1/v2 [19, 5], BYOL [18], and SwAV [3]. However, the pretext tasks designed in these works are suboptimal when transferring to object detection, because many differences between image classification and object detection are neglected in previous methods. For example, firstly, image classification is typically solved using

*equal contribution

the 1-of-K loss function, which assumes that each image only has one category. This is controversial with object detection where an image often has many objects of different categories. Secondly, object detection often needs to perform object classification and box regression on local image regions (patches), but image classification needs global image representation. Thirdly, recent advanced object detectors usually predict objects on multi-level features, while image classifiers typically learn high-level discriminative features.

As object detection plays an indispensable role in many computer vision tasks, this paper aims at *designing a self-supervised representation learning method, which is more powerful when transferring the learned representations to many detection-related downstream tasks* (e.g. *detection, segmentation, pose estimation, and 3D shape prediction) than the existing self-supervised approaches.*

To this end, we first investigate the inconsistency between the accuracy of image classification and the accuracy of object detection produced by the latest self-supervised methods. Then we present three potential practices in order to design a suitable pretext task for object detection. Finally, following these practices, we design DetCo, a detection-friendly contrastive pretext task that is able to train on large-scale unlabeled data.

Specifically, DetCo has two merits that are specially designed for improving object detection. First, a few hierarchical contrastive loss functions are applied to different stages of the backbone network. This is to ensure the discriminative capability of each stage, leading to better performance for object detection with multiple scales. This compensates for the discrepancy between object detection and image classification. We analyze this design choice in section 3.2.1 and evaluate its effectiveness with ablation studies in section 6.6. Second, we propose a novel contrastive learning method that combines the advantages of both global and local representations, by using global images and local image patches as input and establishing different contrastive losses between them. That is, we design cross contrasts between global and local information, which is not only beneficial in object detection, but also favorable in image classification, enabling us to surpass previous works, as shown in Figure 1. We also theoretically justify the essential to combine multiple global and local contrastive losses, showing that they are able to improve the lower bound of mutual information between two different views in contrastive learning, as shown in section 3.2.2 and section 3.2.3.

The main **contributions** of this work are three-fold. (1) We demonstrate the inconsistency of accuracies between image classification and object detection, when previous self-supervised learned representations are transferred to downstream tasks. We propose three potential practices to design suitable unsupervised pretext task for object detection. As far as we know, this is the first work that deeply studies this problem. (2) We propose a novel detection-friendly self-supervised method, DetCo, which is able to combine multiple global and local contrastive losses to improve contrastive learning to pre-train discriminative representations for object detection. Theoretical justification shows that DetCo is able to improve the lower bound of mutual information in contrastive learning. (3) Extensive experiments on PASCAL VOC [15], COCO [28] and Cityscapes [6] show that DetCo outperforms previous state-of-the-art methods when transferred to different downstream tasks such as object detection, segmentation, pose estimation, and 3D shape prediction.

2. Related Work

Existing unsupervised methods for representation learning can be roughly divided into two classes, generative and discriminative. Generative methods [11, 14, 12, 2] typically rely on auto-encoding of images [38, 23, 37] or adversarial learning [17], and operate directly in pixel space. Therefore, most of them are computationally expensive, and the pixel-level details required for image generation may not be necessary for learning high-level representations.

Among discriminative methods [9, 5], self-supervised contrastive learning [5, 19, 5, 3, 18] currently achieved state-of-the-art performance, arousing extensive attention from researchers. Unlike generative methods, contrastive learning avoids the computation-consuming generation step by pulling representations of different views of the same image (*i.e.*, positive pairs) close, and pushing representations of views from different images (*i.e.*, negative pairs) apart. Chen *et al*. [5] developed a simple framework, termed SimCLR, for contrastive learning of visual representations. It learns features by contrasting images after a composition of data augmentations. After that, He *et al*. [19] and Chen *et al*. [5] proposed MoCo and MoCo v2, using a moving average network (momentum encoder) to maintain consistent representations of negative pairs drawn from a memory bank. Recently, SwAV [3] introduced online clustering into contrastive learning, without requiring to compute pairwise comparisons. BYOL [18] avoided the use of negative pairs by bootstrapping the outputs of a network iteratively to serve as targets for an enhanced representation.

Moreover, earlier methods rely on all sorts of pretext tasks to learn visual representations. Relative patch prediction [9, 10], colorizing gray-scale images [40, 24], image inpainting [34], image jigsaw puzzle [31], image super-resolution [25], and geometric transformations [13, 16] have been proved to be useful for representation learning.

Nonetheless, most of the aforementioned methods are specifically designed for image classification while neglecting object detection. Our work focus on designing a better

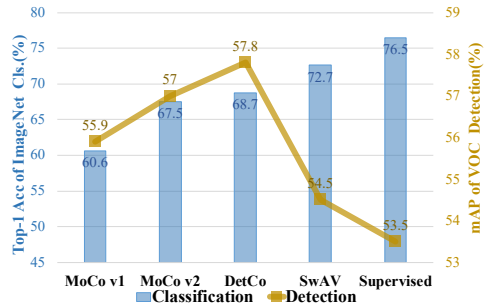


Figure 2. Performance of several self-supervised methods transferring to downstream tasks, ImageNet classification and PASCAL VOC detection. It shows the accuracy of **classification and detection are inconsistent and have low correlation**.

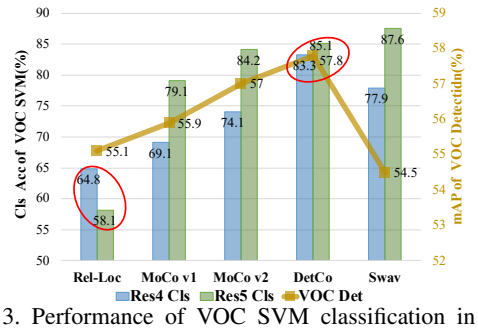


Figure 3. Performance of VOC SVM classification in `Res4`, `Res5` and detection. Although Relative-Loc is a non-contrastive method, it **keeps shallow layer feature discriminative and predicts position between local patches**, enabling competitive detection results.

pretext task friendly to object detection.

3. Methods

In this section, we first analyze the misalignment of accuracy between classification and detection with state-of-the-art unsupervised methods, and then point out three practices of designing a detection-friendly pretext task. Second, following the proposed practices, we propose DetCo shown in figure 4. It is composed of (1) a hierarchical intermediate contrastive loss that keeps features at multiple stages discriminative; (2) a cross global-and-local contrasts, *i.e.*, building the contrastive loss across the global image and local patch features by removing part of the contextual information to enhance the representation ability. After that, we give a theoretical proof that a cross global-and-local contrast can improve the lower bound of mutual information. Finally, we present the implementation details of DetCo, *e.g.*, the setting of important hyper-parameters.

3.1. Inconsistency of Classification and Detection

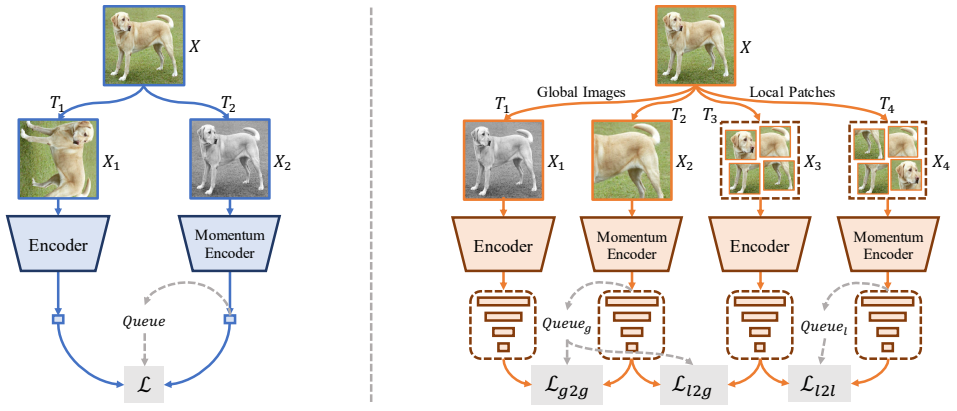
We analyze in detail the performance of recent self-supervised learning methods by transferring them to image classification and object detection. We are surprised to find that the performance of classification and detection is largely inconsistent. Specifically, we select a series of methods: supervised ResNet50 [22], Relative-Loc [9], MoCo v1 [19], MoCo v2 [5], and SwAV [3]. To ensure impartial comparisons, we follow the same fine-tuning setting from MoCo [19]. Detailed settings can be found in Appendix. Following [19, 4], we report the linear classification top-1 accuracy on ImageNet [8] for image classification, and report mAP on PASCAL VOC 07+12 [15] for object detection. Note that, the results of Relative-Loc are borrowed from OpenSelfSup [1] and the detection results of SwAV are produced by us, using the pre-trained weights from official code [2].

[1]https://github.com/open-mmlab/OpenSelfSup
[2]https://github.com/facebookresearch/swav

As shown in Figure 2, SwAV achieves the best linear classification top-1 accuracy 72.7%, which is 12.1% higher than MoCo v1 and 5.2% than MoCo v2. However, on the detection task, MoCo v2 achieves 57.0% mAP, while SwAV yields merely 54.5%, close to the supervised ResNet-50. Moreover, from Figure 3, we find that although the VOC classification performance of Relative-Loc [9] is much lower than other methods, the detection performance is competitive. These phenomena indicate that *for self-supervised pretext methods, the transferring performance on image classification has a low correlation with that on object detection.*

Why are the detection performance of these methods so different? MoCo v1 and v2 are contrastive learning methods, while SwAV is a clustering-based method, where samples are classified into 3000 cluster centers during training. Therefore, the training process of SwAV is similar to supervised classification methods to some extent. As a result, compared with contrastive learning methods, clustering-based methods are more friendly for the image classification task, and this is why SwAV has similar performance to the supervised ResNet50 on both image classification and object detection. Moreover, we consider that contrastive learning methods are better than clustering-/classification-based methods for object detection, because the latter assumes a prior knowledge that there is only one object in one given image, which is misaligned with the target of object detection. While contrastive learning methods do not require this prior knowledge, it discriminates image from a holistic perspective.

Why does the non-contrastive method Relative-Loc achieve competitive detection performance? It is an interesting phenomenon that Relative-Loc falls far below contrastive methods in classification while keeping competitive in detection, as shown in figure 3. We consider two potential reasons: (1) In Relative-Loc, not only the final features but also the features from the shallow stage have strong discrimination capability (see red circle in figure 3), which in-



(a) MoCo (b) DetCo

Figure 4. **The overall pipeline of DetCo compared with MoCo [19].** (a) is MoCo's framework, which only considers the high-level feature and learning contrast from a global perspective. (b) is our DetCo, which straightforwardly appends hierarchical intermediate contrast and two additional local patch views for input, building contrastive loss cross the global and local representation. Our DetCo improves the detection transferring ability by following the proposed three good practices. Note that "T" means image transforms and "L_{l2g}" means contrastive loss cross local and global features. "$Queue_{g/l}$" means different memory banks [39] for global/local features.

spires us to enhance the discrimination capability of features from different depths, thereby improving the detection performance. (2) Relative-Loc focuses on predicting relative positions between local patches, which benefits the detection task because local representation's capability is essential for object detection. This phenomenon inspires us to enhance the local representation in the contrastive learning framework.

What is the guideline for designing a detection-friendly pretext task? Based on the above analysis, we argue that designing a good detection-friendly pretext task is different from designing a classification-friendly one. Here, we summarize three good practices for detection-friendly pretext tasks. (1) Instance discrimination is better than classification or clustering to serve as a pretext task for object detection. (2) Pretext tasks should keep both low-level and high-level features discriminative for object detection. (3) Apart from global image features, local patch features are also essential for object detection. Especially, for practice (2), modern detectors often predict results on multi-level feature maps (*e.g.* Faster RCNN-FPN [26], RetinaNet [27]), and thus reliable multi-level feature maps are required, which is consistent with our practice (2). We follow the practice (1) to adopt MoCo v2 as our baseline model, and design DetCo in Section 3.2 by strictly following the practice (2) and (3). We conduct controlled experiments in Section 4 to verify the proposed practices.

3.2. DetCo Framework

Following the proposed practices, DetCo is designed by adding the intermediate multi-stage contrastive loss and cross local and global contrasts based on MoCo v2. The overall architecture of DetCo is illustrated in Figure 4. The loss function of DetCo can be defined as follows:

$$\mathcal{L}(\mathbf{I}_q, \mathbf{I}_k, \mathbf{P}_q, \mathbf{P}_k) = \sum_{i=1}^{4} w_i \cdot (\mathcal{L}^i_{g\leftrightarrow g} + \mathcal{L}^i_{l\leftrightarrow l} + \mathcal{L}^i_{g\leftrightarrow l}), \quad (1)$$

where $\mathbf{I}$ represents a global image and $\mathbf{P}$ represents the local patch set. Eqn. 1 is a multi-stage contrastive loss. In each stage, there are three cross local and global contrastive losses. We will describe the multi-stage contrastive loss $\sum_{i=1}^{4} w_i \cdot \mathcal{L}^i$ in Section 3.2.1, and the cross local and global contrasts $\mathcal{L}^i_{g\leftrightarrow g} + \mathcal{L}^i_{l\leftrightarrow l} + \mathcal{L}^i_{g\leftrightarrow l}$ in Section 3.2.2.

3.2.1 Intermediate Contrastive Loss

The practice (2) requires both low-level and high-level features to keep strong instance discrimination ability. To verify the effectiveness of the practice (2), we make an intuitive modification to the original MoCo. Specifically, we feed one image to a standard backbone ResNet-50, and it outputs features from different stages, termed `Res2`, `Res3`, `Res4`, `Res5`. MoCo only uses `Res5`, but we use all levels of features to calculate contrastive losses, ensuring that each stage of the backbone produces discriminative representations.

Given an image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, it is first transformed to two global views $\mathbf{I}_q$ and $\mathbf{I}_k$ with two transformations randomly drawn from a set of transformations on global views, termed $\mathcal{T}_g$. We aim at training a $\texttt{encoder}_\texttt{q}$ together with a $\texttt{encoder}_\texttt{k}$ with the same architecture, where $\texttt{encoder}_\texttt{k}$ update weights using a momentum update strategy [19]. The $\texttt{encoder}_\texttt{q}$ contains a backbone and four global MLP heads to extract features from four levels. We feed $\mathbf{I}_q$ to the backbone $b^\theta_q(\cdot)$, with parameters θ that extracts features $\{f_2, f_3, f_4, f_5\} = b^\theta_q(\mathbf{I}_q)$, where

f_i means the feature from the i-th stage. After obtaining the multi-level features, we append four global MLP heads $\{mlp_q^2(\cdot), mlp_q^3(\cdot), mlp_q^4(\cdot), mlp_q^5(\cdot)\}$ whose weights are non-shared. As a result, we obtain four global representations $\{q_2^g, q_3^g, q_4^g, q_5^g\} = \texttt{encoder}_\texttt{q}(\mathbf{I}_\texttt{q})$. Likewise, we can easily get $\{k_2^g, k_3^g, k_4^g, k_5^g\} = \texttt{encoder}_\texttt{k}(\mathbf{I}_\texttt{k})$.

MoCo uses InfoNCE to calculate contrasitive loss, formulated as:

$$\mathcal{L}_{g\leftrightarrow g}(\mathbf{I}_q, \mathbf{I}_k) = -\log \frac{\exp(q^g \cdot k_+^g/\tau)}{\sum_{i=0}^{K} \exp(q^g \cdot k_i^g/\tau)}, \quad (2)$$

where τ is a temperature hyper-parameter [39].

Our loss function is similar to MoCo, except that we extend it to multi-level contrastive losses for multi-stage features, formulated as:

$$Loss = \sum_{i=1}^{4} w_i \cdot \mathcal{L}_{g\leftrightarrow g}^i, \quad (3)$$

where w is the loss weight, and i indicates the current stage. Inspired by the loss weight setting in PSPNet [41], we set the loss weight of shallow layers to be smaller than deep layers, and obtain the optimal setting by grid search. In addition, we build an individual memory bank $queue_i$ for each layer. In the appendix, we provide the pseudo-code of intermediate contrastive loss.

3.2.2 Cross Global and Local Contrast

Following the practice (3), we aim at enhancing the local patch representation of DetCo. We first transform the input image into 9 local patches using jigsaw augmentation, the augmentation detail is shown in section 6.5. In this way, the contextual information of the global image is reduced. These patches pass through the encoder, and then we can get 9 local feature representation. After that, we combine these features into one feature representation, and build a cross global-and-local contrastive loss.

Given an image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, first it is transformed to two local patch set $\mathbf{P}_q$ and $\mathbf{P}_k$ by two transformations selected from a local transformation set, termed $\mathcal{T}_l$. There are 9 patches $\{p_1, p_2, ..., p_9\}$ in each local patch set. We feed the local patch set to backbone and get 9 features $F_p = \{f_{p1}, f_{p2}, ..., f_{p9}\}$ at each stage. Taking a stage as an example, we build a local patch MLP head $mlp_{local}(\cdot)$, which does not share weights with $mlp_{global}(\cdot)$ in section 3.2.1. Then, F_p is concatenated and fed to the local patch MLP head to get final representation q^l. Likewise, we can use the same approach to get k^l.

The cross contrastive loss has two parts: the global↔local contrastive loss and the local↔local contrastive loss. The global↔local contrastive loss can be written as:

$$\mathcal{L}_{g\leftrightarrow l}(\mathbf{P}_q, \mathbf{I}_k) = -\log \frac{\exp(q^l \cdot k_+^g/\tau)}{\sum_{i=0}^{K} \exp(q^l \cdot k_i^g/\tau)}. \quad (4)$$

Similarly, the local↔local contrastive loss can be formulated as:

$$\mathcal{L}_{l\leftrightarrow l}(\mathbf{P}_q, \mathbf{P}_k) = -\log \frac{\exp(q^l \cdot k_+^l/\tau)}{\sum_{i=0}^{K} \exp(q^l \cdot k_i^l/\tau)}. \quad (5)$$

By reducing the contextual information of global image and building cross global-and-local contrast, each local patches of the image now are aware of instance discrimination. As a result, both the detection and classification performance boost up. We will give a theory explanation from the perspective of Mutual Information Optimization in Section 3.2.3.

3.2.3 Improving Lower Bound of Mutual Information

This section analyzes that DetCo can improve the lower bound (LB) of the mutual information (MI) between two views, leading to better contrastive learning. Oord *et al.* [33] demonstrated that contrastive learning is to maximize the lower bound of mutual information, equivalent to minimize the InfoNCE loss function,

$$\mathbf{MI} \geqslant \log(K) - \mathcal{L}_{NCE} \triangleq \texttt{Lower Bound (LB)},$$

where $\mathcal{L}_{NCE}$ is defined in Eqn.(2) and K is the number of negative samples. In section 3.2.2, we introduce the cross global-and-local contrasts. Here we can show that the $\mathbf{MI}$ lower bound between a global image view $\mathbf{I}_1$ and a set of local patches $\mathbf{P}_2$ is larger than that between two global image views $\mathbf{I}_1$ and $\mathbf{I}_2$, denoted as $\texttt{LB}^{g\leftrightarrow l} > \texttt{LB}^{g\leftrightarrow g}$. We have

$$\begin{aligned} \texttt{LB}^{g\leftrightarrow l} - \texttt{LB}^{g\leftrightarrow g} &= (\log(K) - \mathcal{L}_{NCE}^{g\leftrightarrow l}) - (\log(K) - \mathcal{L}_{NCE}^{g\leftrightarrow g}) \\ &= \mathcal{L}_{NCE}^{g\leftrightarrow g}(\mathbf{I}_1, \mathbf{I}_2) - \mathcal{L}_{NCE}^{g\leftrightarrow l}(\mathbf{I}_1, \mathbf{P}_2) > 0. \end{aligned} \quad (6)$$

Intuitively, Eqn.(6) is established because the information of a complete image often contains two parts, content information and contextual information, denoted as $\texttt{Info}_I = (\texttt{Info}_{content}, \texttt{Info}_{context})$. And a set of randomly-jittered local patches $\mathbf{P}$ would lose the contextual information compared to the global image $\mathbf{I}$, making the similarity between positive global↔local pairs tend to be smaller than positive global↔global pairs (*i.e.* the similarity between two global image views is often larger than the similarity between a set of random local patches and a global image), as demonstrated in Appendix. That is, we have $\mathcal{L}_{NCE}^{g\leftrightarrow g}(\mathbf{I}_1, \mathbf{I}_2) > \mathcal{L}_{NCE}^{g\leftrightarrow l}(\mathbf{I}_1, \mathbf{P}_2)$.

With $\texttt{LB}^{g\leftrightarrow l} > \texttt{LB}^{g\leftrightarrow g}$, we see that the global↔local contrastive loss improves the lower bound of mutual information compared to the contrast between two global image views. Similarly, we can also verify that the local↔local contrast has the same benefit.

3.3. Implementation Details

Here we introduce the basic details. See Appendix for more details .

Image augmentations. The global image augmentation is the same as MoCo v2 [5]. First, a random region is cropped with at least 20% of the image and resized to 224×224 with a random horizontal flip, followed by a random color jittering related to brightness, contrast, saturation, hue and grayscale. Gaussian Blur is also used for augmentation. The local patch augmentation follows PIRL [30]. First, a random region is cropped with at least 60% of the image and resized to 255×255, followed by random flip, color jitter and blur, sharing the same parameters with global augmentation. Then we divide the image into 3×3 grids; each grid is 85×85. A random crop is applied on each patch to get 64×64 to avoid continuity between patches. Finally, we obtain nine patches.

Training Details. We use OpenSelfSup [3] as the codebase. Batch size is 256 with 8 V100 GPUs for every experiment. We use standard ResNet-50 [22] for all experiments. For the pretext task, most training hyper-parameters are the same as MoCo v2. We pre-train 100 epochs on ImageNet for the ablation study, . Unless other specified, we pre-train 200 epochs on ImageNet for fair comparison with other methods. We try Auto Augmentation strategy [7] in DetCo pre-train and it improves the performance on downstream COCO detection.

4. Experiments

4.1. Ablation Study

Experiment Settings. We conduct all the controlled experiments by training 100 epochs. We adopt MoCo v2 as our strong baseline. More ablation studies about hyper-parameters are shown in Appendix. In table 1 and 2, *"HIC"* means **H**ierarchical **I**ntermediate **C**ontrastive loss, and *"CGLC"* means **C**ross **G**lobal and **L**ocal **C**ontrasts.

Effectiveness of hierarchical intermediate contrastive loss. As shown in Table 1 (a) and (b), when adding the hierarchical intermediate contrastive loss on MoCo v2, the performance of classification *drop* but detection *increase*. This perfectly matches the analysis and practice (2) in Section 3.1: For classification, only the last feature needs to keep discriminative. However, for detection, it is better to keep features at multiple stages discriminative. We also evaluate the VOC SVM classification accuracy at four stages: `Res2`, `Res3`, `Res4`, `Res5` to demonstrate the enhancement of the intermediate feature. As shown in Table 2 (a) and (b), the discrimination ability of shallow features vastly improves compared with baseline.

Effectiveness of cross global and local contrasts. As shown in Table 1 (b) and (c), when adding cross global and local contrasts, the performance of both classification and detection boosts up and surpasses MoCo v2 baseline. This improvement mainly benefits from the local view that removes the context information and improves the lower bound of mutual information. Also, the cross global and local contrasts keep each local patch learn an independent representation of the whole image, which is consistent with object detection. From table 2 (b) and (c), the cross contrasts further improve the representation ability of all the stages.

[3] https://github.com/open-mmlab/OpenSelfSup

	+HIC	+CGLC	Top1	Top5	mAP
(a)	×	×	64.3	85.6	56.3
(b)	✓	×	63.2 ↓	84.9 ↓	57.0 ↑
(c)	✓	✓	66.6 ↑	87.2 ↑	57.4 ↑

Table 1. **Ablation: hierarchical intermediate contrastive loss (HIC) and cross global and local contrasts(CGLC)**. The results are evaluated on ImageNet linear classification and PASCAL VOC07+12 detection.

	+HIC	+CGLC	Res2	Res3	Res4	Res5
(a)	×	×	47.1	58.2	70.9	82.1
(b)	✓	×	50.9 ↑	67.1 ↑	78.7 ↑	81.8 ↓
(c)	✓	✓	51.6 ↑	69.7 ↑	82.5 ↑	84.3 ↑

Table 2. **Ablation: hierarchical intermediate contrastive loss (HIC) and cross global and local contrasts(CGLC)**. Accuracy of feature in different stages are evaluated by PASCAL VOC07 SVM classification.

4.2. Transfer Results on General Object Detection

Setup. We choose three representative detectors: Faster RCNN-C4 [36], Faster RCNN-FPN [26] and RetinaNet [27]. The first two detectors are two-stage and RetinaNet is one stage detector. Our training settings strictly follow MoCo [19], including using "SyncBN" [35] in backbone and FPN. We report object detection results on PASCAL VOC and COCO dataset.

PASCAL VOC. As shown in Table 6, MoCo v2 is a strong baseline compared with other unsupervised methods. However, with only 100 epoch pre-training, DetCo achieves almost the same performance as MoCo v2-800ep (800 epoch pre-training). Moreover, DetCo-800ep establishes the new state-of-the-art at 58.2 mAP and 65.0 AP_{75}, 6.2% improvement in AP_{75} compared with supervised counterpart.

COCO with 1× and 2× schedule. Table 4 shows the Mask RCNN [21] results on standard 1× schedule, DetCo also outperforms MoCo v2 and other methods in all metrics. The results of 2× schedule is in Appendix. The column 2-3 of Table 5 shows the results of one stage detector RetinaNet. DetCo pretrain is also better than ImageNet supervised methods and MoCo v2 in 1× and 2× schedule. For instance, DetCo is 1.3% higher than MoCov2 on AP_{50}, 1× schedule.

COCO with fewer training iterations. COCO dataset is much larger than PASCAL VOC in the data scale. Even training from scratch [20] can get a satisfactory result. To verify the effectiveness of unsupervised pre-training, we conduct experiments on extremely stringent conditions: only train detectors with 12k iterations($\approx$ 1/7× *vs.* 90k-

Method	Mask R-CNN R50-C4 COCO 12k						Mask R-CNN R50-FPN COCO 12k					
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Rand Init	7.9	16.4	6.9	7.6	14.8	7.2	10.7	20.7	9.9	10.3	19.3	9.6
Supervised	27.1	46.8	27.6	24.7	43.6	25.3	28.4	48.3	29.5	26.4	45.2	25.7
InsDis[39]	25.8(-1.3)	43.2(-3.6)	27.0(-0.6)	23.7(-1.0)	40.4(-3.2)	24.5(-0.8)	24.2(-4.2)	41.5(-6.8)	25.1(-4.4)	22.8(-3.6)	38.9(-6.3)	23.7(-2.0)
PIRL[30]	25.5(-1.6)	42.6(-4.2)	26.8(-0.8)	23.2(-1.5)	39.9(-3.7)	23.9(-1.4)	23.7(-4.7)	40.4(-7.9)	24.4(-5.1)	22.1(-4.3)	37.9(-7.3)	22.7(-3.0)
SwAV[3]	16.5(-10.6)	35.2(-11.6)	13.5(-14.1)	16.1(-8.6)	32.0(-11.6)	14.6(-10.7)	25.5(-2.9)	46.2(-2.1)	25.4(-4.1)	24.8(-1.6)	43.5(-1.7)	25.3(-0.4)
MoCo[19]	26.9(-0.2)	44.5(-2.3)	28.2(+0.6)	24.6(-0.1)	41.8(-1.8)	25.6(+0.3)	25.6(-2.8)	43.4(-4.9)	26.6(-2.9)	23.9(-2.5)	40.8(-4.4)	24.8(-0.9)
MoCov2[5]	27.6(+0.5)	45.3(-1.5)	28.9(+1.3)	25.1(+0.4)	42.6(-1.0)	26.3(+1.0)	26.6(-1.8)	44.9(-3.4)	27.7(-1.8)	24.8(-1.6)	42.1(-3.1)	25.7(0.0)
DetCo	**29.3**(+2.2)	**48.4**(+1.6)	**30.3**(+2.7)	**26.5**(+1.8)	**45.3**(+1.7)	**27.3**(+2.0)	27.9(-0.5)	46.9(-1.4)	29.3(-0.2)	26.0(-0.4)	44.2(-1.0)	**26.9**(+1.2)
DetCo+AA	**29.8**(+2.7)	**49.1**(+2.3)	**31.4**(+3.8)	**26.9**(+2.2)	**46.0**(+2.4)	**27.9**(+2.6)	**29.6**(+1.2)	**49.4**(+1.1)	**31.0**(+1.5)	**27.6**(+1.2)	**46.6**(+1.4)	**28.7**(+3.0)

Table 3. **Object detection and instance segmentation fine-tuned on COCO**. All methods are pretrained 200 epochs on ImageNet. **Green** means increase and **gray** means decrease. Our method converges faster than other unsupervised methods under a limited number of iterations. "AA" means we use Auto Augmentation in pre-training.

Method	Mask R-CNN R50-C4 COCO 90k						Mask R-CNN R50-FPN COCO 90k					
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Rand Init	26.4	44.0	27.8	29.3	46.9	30.8	31.0	49.5	33.2	28.5	46.8	30.4
Supervised	38.2	58.2	41.2	33.3	54.7	35.2	38.9	59.6	42.7	35.4	56.5	38.1
InsDis[39]	37.7(-0.5)	57.0(-1.2)	40.9(-0.3)	33.0(-0.3)	54.1(-0.6)	35.2(0.0)	37.4(-1.5)	57.6(-2.0)	40.6(-2.1)	34.1(-1.3)	54.6(-1.9)	36.4(-1.7)
PIRL[30]	37.4(-0.8)	56.5(-1.7)	40.2(-1.0)	32.7(-0.6)	53.4(-1.3)	34.7(-0.5)	37.5(-1.4)	57.6(-2.0)	41.0(-1.7)	34.0(-1.4)	54.6(-1.9)	36.2(-1.9)
SwAV[3]	32.9(-5.3)	54.3(-3.9)	34.5(-6.7)	29.5(-3.8)	50.4(-4.3)	30.4(-4.8)	38.5(-0.4)	60.4(+0.8)	41.4(-1.3)	35.4(0.0)	57.0(+0.5)	37.7(-0.4)
MoCo[19]	38.5(+0.3)	58.3(+0.1)	41.6(+0.4)	33.6(+0.3)	54.8(+0.1)	35.6(+0.4)	38.5(-0.4)	58.9(-0.7)	42.0(-0.7)	35.1(-0.3)	55.9(-0.6)	37.7(-0.4)
MoCov2[5]	38.9(+0.7)	58.4(+0.2)	42.0(+0.8)	34.2(+0.9)	55.2(+0.5)	36.5(+1.3)	38.9(0.0)	59.4(-0.2)	42.4(-0.3)	35.5(+0.1)	56.5(0.0)	38.1(0.0)
DetCo	**39.4**(+1.2)	**59.2**(+1.0)	**42.3**(+1.1)	**34.4**(+1.1)	**55.7**(+1.0)	**36.6**(+1.4)	**39.5**(+0.6)	60.3(+0.7)	**43.1**(+0.4)	**35.9**(+0.5)	56.9(+0.4)	**38.6**(+0.5)
DetCo+AA	**39.8**(+1.6)	**59.7**(+1.5)	**43.0**(+1.8)	**34.7**(+1.4)	**56.3**(+1.6)	**36.7**(+1.5)	**40.1**(+1.2)	**61.0**(+1.4)	**43.9**(+1.2)	**36.4**(+1.0)	**58.0**(+1.5)	**38.9**(+0.8)

Table 4. **Object detection and instance segmentation fine-tuned on COCO**. All methods are pretrained 200 epochs on ImageNet. Our DetCo is state-of-the-art, surpassing MoCov2 and the supervised method in all metrics. "AA" means we use Auto Augmentation in pre-training.

Method	RetinaNet R50 12k			RetinaNet R50 90k			RetinaNet R50 180k			Keypoint RCNN R50 180k		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}
Rand Init	4.0	7.9	3.5	24.5	39.0	25.7	32.2	49.4	34.2	65.9	86.5	71.7
Supervised	24.3	40.7	25.1	37.4	56.5	39.7	38.9	58.5	41.5	65.8	86.9	71.9
InsDis[39]	19.0(-5.3)	32.0(-8.7)	19.6(-5.5)	35.5(-1.9)	54.1(-2.4)	38.2(-1.5)	38.0(-0.9)	57.4(-1.1)	40.5(-1.0)	66.5(+0.7)	87.1(+0.2)	72.6(+0.7)
PIRL[30]	19.0(-5.3)	31.7(-9.0)	19.8(-5.3)	35.7(-1.7)	54.2(-2.3)	38.4(-1.3)	38.5(-0.4)	57.6(-0.9)	41.2(-0.3)	66.5(+0.7)	87.5(+0.6)	72.1(+0.2)
SwAV[3]	19.7(-4.6)	34.7(-6.0)	19.5(-5.6)	35.2(-2.2)	54.9(-1.6)	37.5(-2.2)	38.6(-0.3)	58.8(+0.3)	41.1(-0.4)	66.0(+0.2)	86.9(0.0)	71.5(-0.4)
MoCo[19]	20.2(-4.1)	33.9(-6.8)	20.8(-4.3)	36.3(-1.1)	55.0(-1.5)	39.0(-0.7)	38.7(-0.2)	57.9(-0.6)	41.5(0.0)	66.8(+1.0)	87.4(+0.5)	72.5(+0.6)
MoCov2[5]	22.2(-2.1)	36.9(-3.8)	23.0(-2.1)	37.2(-0.2)	56.2(-0.3)	39.6(-0.1)	39.3(+0.4)	58.9(+0.4)	42.1(+0.6)	66.8(+1.0)	87.3(+0.4)	73.1(+1.2)
DetCo	23.6(-0.7)	38.7(-2.0)	24.6(-0.5)	**38.0**(+0.6)	**57.4**(+0.9)	**40.7**(+1.0)	**39.8**(+0.9)	**59.5**(+1.0)	**42.4**(+0.9)	**67.2**(+1.4)	**87.5**(+0.6)	**73.4**(+1.5)
DetCo+AA	**25.3**(+1.0)	**41.6**(+0.9)	**26.5**(+1.4)	**38.4**(+1.0)	**57.8**(+1.3)	**41.2**(+1.5)	**39.7**(+0.8)	**59.3**(+0.8)	**42.6**(+1.1)	-	-	-

Table 5. **One-stage object detection and keypoint detection fine-tuned on COCO**. All methods are pretrained 200 epochs on ImageNet. DetCo outperforms all unsupervised counterparts. "AA" means we use Auto Augmentation in pre-training.

$1\times$ schedule). The 12k iterations make detectors heavily under-trained and far from converge, as shown in Table 3 and Table 5 column 1. Under this setting, for Mask RCNN-C4, DetCo exceeds MoCo v2 by 3.1% in AP^{bb}_{50} and outperforms supervised methods in all metrics, which indicates DetCo can fasten the training converge. For Mask RCNN-FPN and RetinaNet, DetCo also has significant advantages over MoCo v2, and has the closest performance compared with supervised counterpart.

Discussion. On the one hand, when the dataset scale of the downstream task is small (*e.g.* PASCAL VOC), DetCo pre-training has significant advantages compared with the supervised method. Nonetheless, if the dataset scale is very large (*e.g.* COCO), our DetCo is also better than the supervised method, but the advantage is narrowed than the small-scale dataset. On the other hand, when the computational resource is limited, DetCo can fasten training converge compared with other unsupervised methods, and it is on par with supervised methods.

4.3. Transfer Results on More Detection Tasks

Multi-Person Pose Estimation. The last column of Table 5 shows the results of COCO keypoint detection results using Mask RCNN. DetCo also surpasses other methods in all metrics, *e.g.* 1.4% AP^{kp} and 1.5% AP^{kp}_{75} higher than supervised counterpart.

Segmentation for Autonomous Driving. Cityscapes is a dataset for autonomous driving in the urban street. We follow MoCo to evaluate on instance segmentation with Mask RCNN and semantic segmentation with FCN-16s [29]. The results are shown in Table 7. On instance segmentation, DetCo outperforms supervised counterpart by 3.6% on AP^{mk}_{50}. For the semantic segmentation task, which is also a dense prediction task, DetCo is also 1.9% higher than supervised and 0.8% higher than MoCo v2.

3D Human Shape Prediction Estimating 3D shape from a single 2D image is challenging, so we evaluate DetCo on DensePose [1] task. As shown in Table 9, DetCo substantially outperforms ImageNet supervised method and MoCo v2 in all metrics, especially 1.4% on AP_{50}.

Method	Epoch	AP	AP_{50}	AP_{75}
Rand Init	-	33.8	60.2	33.1
Supervised	90	53.5	81.3	58.8
InsDis [39]	200	55.2(+1.7)	80.9(-0.4)	61.2(+2.4)
PIRL [30]	200	55.5(+2.0)	81.0(-0.3)	61.3(+2.5)
SwAV [3]	800	56.1(+2.6)	82.6(+1.3)	62.7(+3.9)
MoCo [19]	200	55.9(+2.4)	81.5(+0.2)	62.6(+3.8)
MoCov2 [5]	200	57.0(+3.5)	82.4(+1.1)	63.6(+4.8)
MoCov2 [5]	800	57.4(+3.9)	82.5(+1.2)	64.0(+5.2)
DetCo	100	57.4(+3.9)	82.5(+1.2)	63.9(+5.1)
	200	57.8(+4.3)	82.6(+1.3)	64.2(+5.4)
	800	**58.2**(+4.7)	**82.7**(+1.4)	**65.0**(+6.2)

Table 6. **Object Detection finetuned on PASCAL VOC07+12 using Faster RCNN-C4.** DetCo-100ep is on par with previous state-of-the-art, and DetCo-800ep achieves the best performance.

Methods	Instance Seg.		Semantic Seg.
	AP^{mk}	AP^{mk}_{50}	mIOU
Rand Init	25.4	51.1	65.3
supervised	32.9	59.6	74.6
InsDis [39]	33.0 (+0.1)	60.1 (+0.5)	73.3 (-1.3)
PIRL [30]	33.9 (+1.0)	61.7 (+2.1)	74.6 (0.0)
SwAV [3]	33.9 (+1.0)	62.4 (+2.8)	73.0 (-1.6)
MoCo [19]	32.3 (-0.6)	59.3 (-0.3)	75.3 (+0.7)
MoCov2 [5]	33.9 (+1.0)	60.8 (+1.2)	75.7 (+1.1)
DetCo	**34.7** (+1.8)	**63.2** (+3.6)	**76.5** (+1.9)

Table 7. **DetCo *vs.* supervised and other unsupervised methods on Cityscapes dataset.** All methods are pretrained 200 epochs on ImageNet. We evaluate instance segmentation and semantic segmentation tasks.

Figure 5. **Attention maps generated by DetCo and MoCov2 [5].** DetCo can activate more accurate object regions in the heatmap than MoCov2. More visualization results are in Appendix.

4.4. Transfer Results on Classification

We follow the standard settings: ImageNet linear classification and VOC SVM classification. The training epoch and learning rate is same as MoCo. Table 8 shows the results, our DetCo also outperforms its strong baseline MoCo v2 by 1.1% in Top-1 Accuracy. It is also competitive on VOC SVM classification accuracy compared with state-of-the-art counterparts.

Discussion. We report classification accuracy *only to verify the robustness of DetCo, because the first thing we pursue is the transferring detection ability*. As analyzed above, SwAV, the strongest method on classification, performs not good on object detection. It also meets our purpose that detection needs different pretext task design with classification, and the design of DetCo is more friendly for detection.

Method	Epoch	ImageNet		VOC07
		Top1	Top5	Acc
Jigsaw [31]	-	44.6	-	64.5
Rotation [16]	-	55.4	-	63.9
InsDis [39]	200	56.5	-	76.6
LocalAgg [42]	200	58.8	-	-
PIRL [30]	800	63.6	-	81.1
SimCLR [4]	1000	69.3	89.0	-
BYOL [18]	1000	74.3	91.6	-
SwAV [3]	200	72.7	-	87.6
MoCo [19]	200	60.6	-	79.2
MoCov2 [5]	200	67.5	-	84.1
DetCo	200	68.6	88.5	85.1

Table 8. **Comparison of ImageNet Linear Classification and VOC SVM Classification.** Although DetCo is designed for detection, it is also robust and competitive on classification task, and it substantially exceeds MoCov2 baseline by 1.5%.

Method	Epoch	AP^{dp}	AP^{dp}_{50}	AP^{dp}_{75}
Rand Init	-	40.8	78.6	37.3
Supervised	90	50.8	86.3	52.6
MoCo [19]	200	49.6(-1.2)	85.9(-0.4)	50.5(-2.1)
MoCo v2 [5]	200	50.9(+0.1)	87.2(+0.9)	52.9(+0.3)
DetCo	200	**51.3**(+0.5)	**87.7**(+1.4)	**53.3**(+0.7)

Table 9. **DetCo *vs.* other methods on Dense Pose task.** It also performs best on monocular 3D human shape prediction.

4.5. Visualization

Figure 5 visualizes the attention map of DetCo and MoCo v2. We can see when there is more than one object in the image, DetCo successfully locates all the objects, while MoCo v2 fails to activate some objects. Moreover, in the last column, the attention map of DetCo is more accurate than MoCo v2 on the boundary. It reflects from the side that the localization capability of DetCo is stronger than MoCo v2.

5. Conclusion

In this paper, we focus on designing a good pretext task for object detection. First, we detailly analyze a series of self-supervised methods and conclude that the performance inconsistency transferring to the classification and detection task. Second, we propose three good practices to design a detection-friendly self-supervised learning framework. Third, follow the proposed practices, we propose DetCo, with hierarchical intermediate contrastive loss and cross global and local contrast. It achieves state-of-the-art performance on a series of detection-related tasks. We believe that there is no single best unsupervised pretext task for different downstream tasks and we will put in more effort to explore that in the future.

Acknowledgement. We thank Huawei to support >200 GPUs and Yaojun Liu for insightful discussion.

6. Appendix

In appendix, we first show that the results of DetCo on Semi-Supervised Object Detection in Section 6.1 and more downstream tasks in Section 6.2. Second, in Section 6.3, we show the visualization results of DetCo and MoCo v2. Third, we give a proof of lower bound improvement in Section 6.4. Fourth, we show more implementation details in Section 6.5. Finally, we analysis more ablation studies of DetCo in Section 6.6.

6.1. Semi-Supervised Object Detection

To verify the effectiveness of self-supervised learning on small scale dataset, we randomly sample 1%, 2%, 5%, 10% data to fine-tune the Mask RCNN C4 / FPN and RetinaNet. For all the settings, we fine-tune the detectors with 12k iterations to avoid overfitting. Other settings are the same as COCO $1\times$ and $2\times$ schedule. The results for Mask RCNN with 1% and 2% data are shown in Table 10. The results for Mask RCNN with 5% and 10% data are shown in Table 11. The results for RetinaNet with 1%, 2%, 5%, 10% are shown in Table 12. From Table 10 and 12, we find that with only 1% and 2% data, all other unsupervised methods have lower results than supervised counterparts. However, DetCo performs better than all supervised / unsupervised methods. Moreover, for 5% and 10% training data, DetCo also outperforms all the counterparts with a large margin. These results shows that the feature representation pre-trained from self-supervised learning method is beneficial for semi-supervised object detection.

6.2. More Experiment Results

COCO with $2\times$ schedule. Table 13 shows the results of Mask RCNN R50 C4 / FPN on COCO with $2\times$ schedule. DetCo achieves state-of-the-art performance on both object detection and instance segmentation. For example, for Mask RCNN-C4, DetCo is *1.6%* better than supervised method on AP^{bb}_{75}, *0.6%* better than MoCo v2 on AP^{bb}_{50}.

LVIS Instance Segmentation. We use LVIS v1.0 for training and evaluation. MoCo [19] adopted LVIS v0.5, but it is outdated and can not be downloaded from the official website. So we fine-tune and compare all the methods using LVIS v1.0 dataset. The training schedule of LVIS is 180k iterations, the same as MoCo. Other settings also keep the same with MoCo. The results are shown in Table 14. DetCo also outperforms MoCo v2 and supervised methods in both detection and instance segmentation.

6.3. More Visualized Results

6.3.1 Visualization of Attention Map

Implementation Details. We visualize the attention map of `Res5` on the ImageNet dataset, which is 1/32 resolution of the input image size. To get relatively clear attention map, we enlarge the input size from $224\times224\times3$ to $448\times448\times3$. The shape of output tensor `Res5` is $14\times14\times2048$. We calculate the mean of tensor `Res5` in the channel dimension and normalize the value to 0-1. Then we get the attention map, which shape is $14\times14\times1$. We further upsample the attention map to input image's size using bilinear interpolation and project the attention map on to the image to get the visualized results.

Visualization Results. As shown in Figure 6, it is surprising to see that both MoCo v2 and DetCo can generate relatively high-quality attention map that focuses on the foreground objects. It demonstrates that contrastive learning-based self-supervised representation methods can potentially solve saliency object detection or object localization in an unsupervised manner. *Moreover*, the attention map of DetCo is much better than MoCo v2 mainly in two aspects: (1) more accurate boundary localization. (2) more object discovery. We analyze that it is mainly due to introducing the global-to-local contrasts into DetCo, forcing each local patch aware of instance discrimination. To optimize global-to-local contrastive loss, each local patch needs to distinguish the foreground feature; that is why DetCo can output a more accurate attention map. However, MoCo v2 uses the whole image to extract features, so it only needs to activate the most discriminative area.

6.3.2 Visualization of Image Retrieval

Implementation Details. We visualize the image retrieval results on the ImageNet validation dataset. First, we extract the final-layer feature of all the images using the features learned by DetCo. Then we use global average pooling(GAP) on the extracted feature, followed by a $L2$ normalization. The shape of each feature vector is $1\times1\times2048$. For retrieval, we randomly select several images as query images, then directly find K nearest images in the feature space. K is set to 9 in this paper.

Visualization Results. Figure 7 shows the nearest-neighbor retrieval results. We find that DetCo can successfully group images according to their categories in most cases in an unsupervised learning manner.

Method	Mask R-CNN R50-FPN COCO 1% Data						Mask R-CNN R50-FPN COCO 2% Data					
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Rand Init	2.5	5.8	1.7	2.3	4.9	1.8	4.5	10.3	3.2	4.3	9.3	3.5
Supervised	10.0	19.9	9.2	9.7	18.3	9.2	13.7	26.6	12.8	13.0	24.2	12.6
MoCo[19]	9.1(-0.9)	17.3(-2.6)	8.6(-0.6)	8.6(-1.1)	16.1(-2.2)	8.3(-0.9)	13.0(-0.7)	24.1(-2.5)	12.6(-0.2)	12.3(-0.7)	22.4(-1.8)	12.2(-0.4)
MoCo v2[5]	9.9(-0.1)	18.7(-1.2)	9.5(+0.3)	9.5(-0.2)	17.2(-1.1)	9.2(0.0)	13.8(+0.1)	25.3(-1.3)	13.4(+0.6)	12.9(-0.1)	23.3(-0.9)	12.7(+0.1)
DetCo	**10.7**(+0.7)	**20.2**(+0.3)	**10.4**(+1.2)	**10.2**(+0.5)	**19.0**(+0.7)	**9.9**(+0.7)	**14.3**(+0.6)	**26.3**(-0.3)	**13.9**(+1.1)	**13.5**(+0.5)	**24.6**(+0.4)	**13.1**(+0.5)
DetCo+AA	**12.4**(+2.4)	**23.5**(+3.6)	**11.8**(+2.6)	**12.1**(+2.4)	**21.9**(+3.6)	**12.0**(+2.8)	**16.0**(+2.3)	**29.6**(+3.0)	**15.6**(+2.8)	**15.3**(+2.3)	**27.4**(+2.2)	**15.1**(+2.5)

Table 10. **Semi-Supervised two-stage Detection fine-tuned on COCO 1% and 2% data**. All methods are pretrained 200 epochs on ImageNet. **Green** means increase and **gray** means decrease. DetCo is better than supervised / unsupervised counterparts in all metrics.

Method	Mask R-CNN R50-FPN COCO 5% Data						Mask R-CNN R50-FPN COCO 10% Data					
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Rand Init	9.2	18.6	7.9	8.8	16.9	8.1	10.1	20.2	9.1	9.8	18.6	9.2
Supervised	19.9	37.0	19.3	18.6	33.7	18.4	23.8	42.8	23.9	22.2	39.6	22.3
MoCo[19]	19.6(-0.3)	35.1(-1.9)	20.0(+0.7)	18.3(-0.3)	32.3(-1.4)	18.6(+0.2)	23.3(-0.5)	40.7(-2.1)	23.9(0.0)	21.9(-0.3)	38.0(-1.6)	22.4(+0.1)
MoCo v2[5]	20.6(+0.7)	36.6(-0.4)	21.0(+1.7)	19.1(+0.5)	33.7(0.0)	19.2(+0.8)	24.1(+0.3)	42.0(-0.8)	24.8(+0.9)	22.5(+0.3)	39.1(-0.5)	23.3(+1.0)
DetCo	**21.4**(+1.5)	**38.1**(+1.1)	**21.6**(+2.3)	**19.9**(+1.3)	**35.1**(+1.4)	**19.8**(+1.4)	**25.3**(+1.5)	**43.9**(+1.1)	**26.0**(+2.1)	**23.6**(+1.4)	**40.8**(+1.2)	**24.0**(+1.7)
DetCo+AA	**21.9**(+2.0)	**39.1**(+2.1)	**22.2**(+2.9)	**20.4**(+1.8)	**36.1**(+2.4)	**20.6**(+2.2)	**26.0**(+2.2)	**45.2**(+2.4)	**27.0**(+3.1)	**24.3**(+2.1)	**42.0**(+2.4)	**25.0**(+2.7)

Table 11. **Semi-Supervised two-stage Detection fine-tuned on COCO 5% and 10% data**. All methods are pretrained 200 epochs on ImageNet. DetCo is better than supervised / unsupervised counterparts in all metrics.

Method	RetinaNet R50 COCO 1% Data			RetinaNet R50 COCO 2% Data			RetinaNet R50 COCO 5% Data			RetinaNet R50 COCO 10% Data		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP50	AP75
Rand Init	1.4	3.5	1.0	2.5	5.6	2.0	3.6	7.4	3.0	3.7	7.5	3.2
Supervised	8.2	16.2	7.2	11.2	21.7	10.3	16.5	30.3	15.9	19.6	34.5	19.7
MoCo[19]	7.0(-1.2)	13.5(-2.7)	6.5(-0.7)	10.3(-0.9)	19.2(-2.5)	9.7(-0.6)	15.0(-1.5)	27.0(-3.3)	14.9(-1.0)	18.2(-1.4)	31.6(-2.9)	18.4(-1.3)
MoCo v2[5]	8.4(+0.2)	15.8(-0.4)	8.0(+0.8)	12.0(+0.8)	21.8(+0.1)	11.5(+1.2)	16.8(+0.3)	29.6(-0.7)	16.8(+0.9)	20.0(+0.4)	34.3(-0.2)	20.2(+0.5)
DetCo	**8.8**(+0.6)	**16.7**(+0.5)	**8.2**(+1.0)	**13.0**(+1.8)	**24.0**(+2.3)	**12.5**(+2.2)	**17.9**(+1.4)	**31.7**(+1.4)	**17.7**(+1.8)	**20.8**(+1.2)	**35.6**(+1.1)	**21.3**(+1.6)
DetCo+AA	**9.9**(+1.7)	**19.3**(+3.1)	**9.1**(+1.9)	**13.5**(+2.3)	**25.1**(+3.4)	**12.7**(+2.4)	**18.7**(+2.2)	**32.9**(+2.6)	**18.7**(+2.8)	**21.9**(+2.3)	**37.6**(+3.1)	**22.3**(+2.6)

Table 12. **Semi-Supervised one-stage Detection fine-tuned on COCO 1% and 2% data**. All methods are pretrained 200 epochs on ImageNet. DetCo is better than supervised / unsupervised counterparts in all metrics.

Method	Mask R-CNN R50-C4 COCO 180k						Mask R-CNN R50-FPN COCO 180k					
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Rand Init	35.6	54.6	38.2	31.4	51.5	33.5	36.7	56.7	40.0	33.7	53.8	35.9
Supervised	40.0	59.9	43.1	34.7	56.5	36.9	40.6	61.3	44.4	36.8	58.1	39.5
MoCo[19]	40.7(+0.7)	60.5(+0.6)	44.1(+1.0)	35.4(+0.7)	57.3(+0.8)	37.6(+0.7)	40.8(+0.2)	61.6(+0.3)	44.7(+0.3)	36.9(+0.1)	58.4(+0.3)	39.7(+0.2)
MoCov2[5]	41.0(+1.0)	60.6(+0.7)	44.5(+1.4)	35.6(+0.9)	57.2(+0.7)	38.0(+1.1)	40.9(+0.3)	61.5(+0.2)	44.7(+0.3)	37.0(+0.2)	58.7(+0.6)	39.8(+0.3)
DetCo	**41.4**(+1.4)	**61.2**(+1.3)	**44.7**(+1.6)	**35.8**(+1.1)	**57.8**(+1.3)	**38.3**(+1.4)	**41.5**(+0.9)	**62.1**(+0.8)	**45.6**(+1.2)	**37.6**(+0.8)	**59.2**(+1.1)	**40.5**(+1.0)
DetCo+AA	**41.3**(+1.3)	**61.2**(+1.3)	**45.0**(+1.9)	**35.8**(+1.1)	**57.9**(+1.4)	**38.2**(+1.3)	**41.5**(+0.9)	**62.5**(+1.2)	**45.6**(+1.2)	**37.7**(+0.9)	**59.5**(+1.4)	**40.5**(+1.0)

Table 13. **Object detection and instance segmentation fine-tuned on COCO**. All methods are pretrained 200 epochs on ImageNet. Our DetCo is state-of-the-art, surpassing MoCov2 and the supervised method in all metrics.

Method	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Rand Init	19.6	31.8	20.9	19.0	29.8	20.1
Supervised	22.7	36.8	24.1	22.2	34.6	23.5
MoCo[19]	23.1	37.4	24.5	22.5	35.1	23.8
MoCo v2[5]	23.2	37.4	24.7	22.8	35.1	24.4
DetCo	**23.5**	**37.7**	**24.8**	**23.0**	**35.5**	**24.5**

Table 14. **DetCo *vs.* supervised and other unsupervised methods on LVIS v1.0 dataset.** All methods are pretrained 200 epochs on ImageNet. We evaluate object detection and instance segmentation tasks.

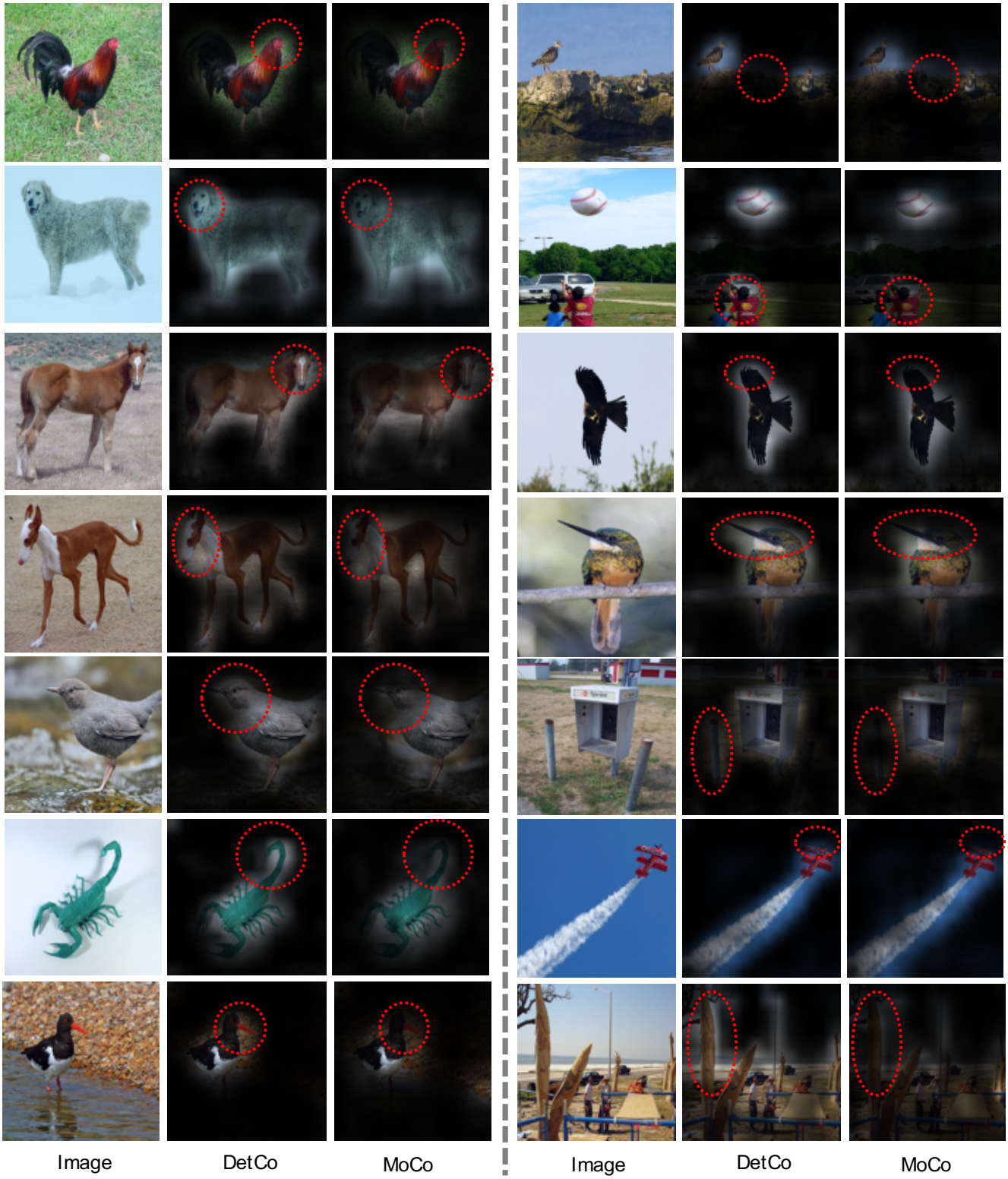


Figure 6. **Attention maps generated by DetCo and MoCov2.** DetCo can activate more object regions in the heatmap than MoCov2, and the attention map of DetCo is more accurate than MoCo v2 in object boundary. *Zoom in for better visualized results.*

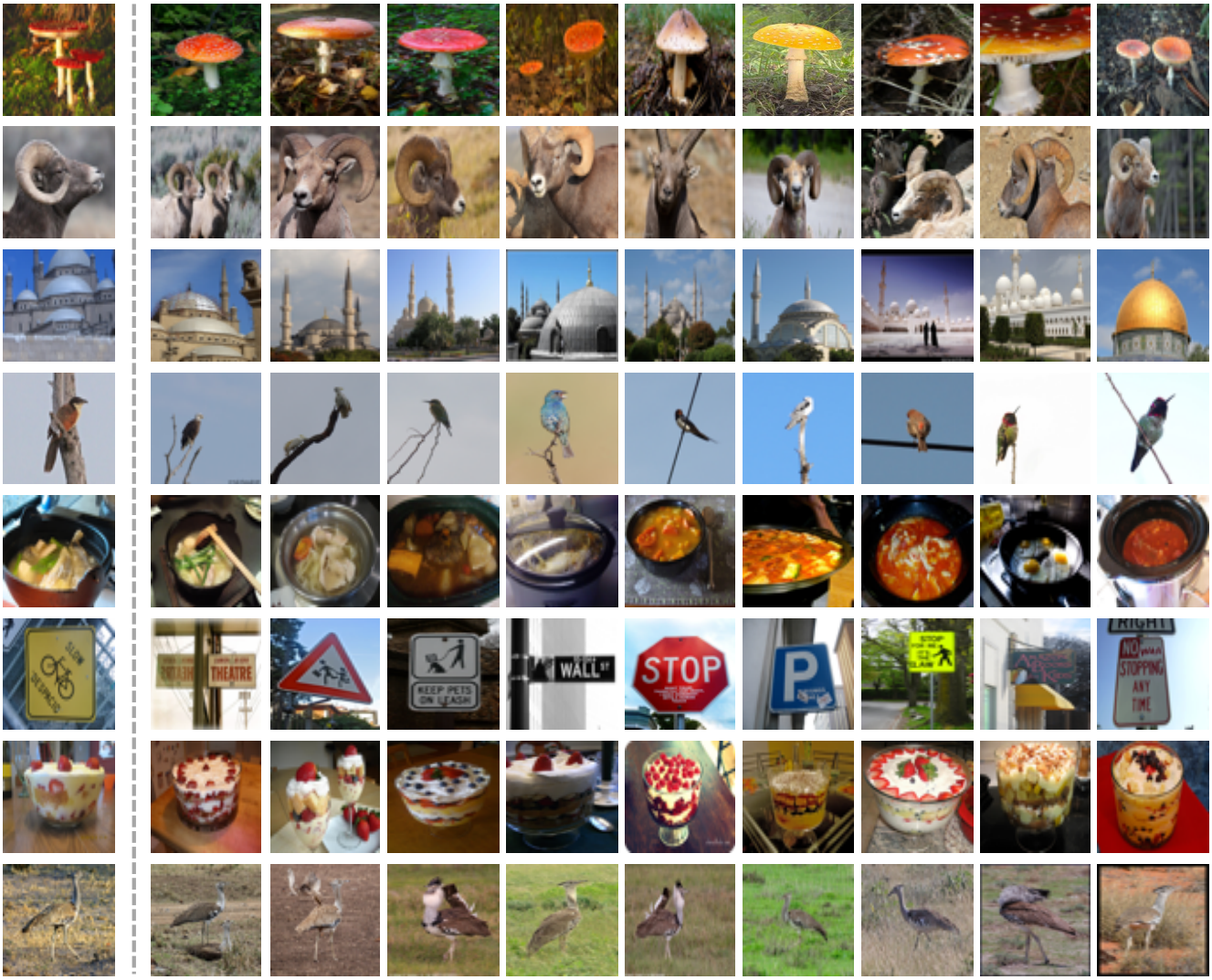


Figure 7. **Retrieval results of DetCo on ImageNet.** The left column are queries from the validation set, while the right columns show 9 nearest neighbors retrieved from the validation set.

6.4. Improving Lower Bound of Mutual Information with Patch Representation

By adding global-to-local contrasts, we define the DetCo loss in Eqn. 1 in the main paper. The additional global↔local contrastive loss improves the lower bound of mutual information compared with global↔global contrastive loss.

As already shown in Section 3.2.3,

$$\mathtt{LB}^{g\leftrightarrow l} - \mathtt{LB}^{g\leftrightarrow g} = \mathcal{L}_{NCE}^{g\leftrightarrow g}(\mathbf{I}_1, \mathbf{I}_2) - \mathcal{L}_{NCE}^{g\leftrightarrow l}(\mathbf{P}_1, \mathbf{I}_2), \quad (7)$$

where $\mathcal{L}_{NCE}^{g\leftrightarrow g}(\mathbf{I}_1, \mathbf{I}_2)$ and $\mathcal{L}_{NCE}^{g\leftrightarrow l}(\mathbf{P}_1, \mathbf{I}_2)$ are defined in Eqn. 2 and 4 in main paper. Here we define exponential cosine similarity $\mathtt{Sim} = \exp(q\cdot k_+/\tau)$ for simplicity, so the InfoNCE of global↔global and global↔local can be re-writen as:

$$\mathcal{L}_{NCE}^{g\leftrightarrow g}(\mathbf{I}_q, \mathbf{I}_k) = -\log \frac{\mathtt{Sim}_P^{g\leftrightarrow g}}{\mathtt{Sim}_P^{g\leftrightarrow g} + \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow g}}, \quad (8)$$

and

$$\mathcal{L}_{NCE}^{g\leftrightarrow l}(\mathbf{P}_q, \mathbf{I}_k) = -\log \frac{\mathtt{Sim}_P^{g\leftrightarrow l}}{\mathtt{Sim}_P^{g\leftrightarrow l} + \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow l}}, \quad (9)$$

where $\mathtt{Sim}_P$ means similarity between positive pairs and $\mathtt{Sim}_N$ means similarity between negative pairs.

So $\mathtt{LB}^{g\leftrightarrow l} - \mathtt{LB}^{g\leftrightarrow g}$ is translated to Eqn. 8 − Eqn.9,

$$\begin{aligned}
\mathtt{LB}^{g\leftrightarrow l} - \mathtt{LB}^{g\leftrightarrow g} &= -\log \frac{\mathtt{Sim}_P^{g\leftrightarrow l}}{\mathtt{Sim}_P^{g\leftrightarrow l} + \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow l}} \\
&\quad - (-\log \frac{\mathtt{Sim}_P^{g\leftrightarrow g}}{\mathtt{Sim}_P^{g\leftrightarrow g} + \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow g}}) \\
&= \log \frac{\mathtt{Sim}_P^{g\leftrightarrow g} \cdot (\mathtt{Sim}_P^{g\leftrightarrow l} + \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow l})}{\mathtt{Sim}_P^{g\leftrightarrow l} \cdot (\mathtt{Sim}_P^{g\leftrightarrow g} + \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow g})}
\end{aligned} \quad (10)$$

Intuitively, if we want to get $\mathtt{LB}^{g\leftrightarrow l} > \mathtt{LB}^{g\leftrightarrow g}$, we need to prove numerator (denote as A) > denominator(denote as B) in Eqn. 10,

$$\begin{aligned}
\mathtt{LB}^{g\leftrightarrow l} - \mathtt{LB}^{g\leftrightarrow g} &\approx \mathtt{A} - \mathtt{B} \\
&= \mathtt{Sim}_P^{g\leftrightarrow g} \cdot (\mathtt{Sim}_P^{g\leftrightarrow l} + \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow l}) \\
&\quad - \mathtt{Sim}_P^{g\leftrightarrow l} \cdot (\mathtt{Sim}_P^{g\leftrightarrow g} + \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow g}) \\
&= \mathtt{Sim}_P^{g\leftrightarrow g} \cdot \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow l} - \mathtt{Sim}_P^{g\leftrightarrow l} \cdot \sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow g} \\
&= P^g \cdot N^l - P^l \cdot N^g,
\end{aligned} \quad (11)$$

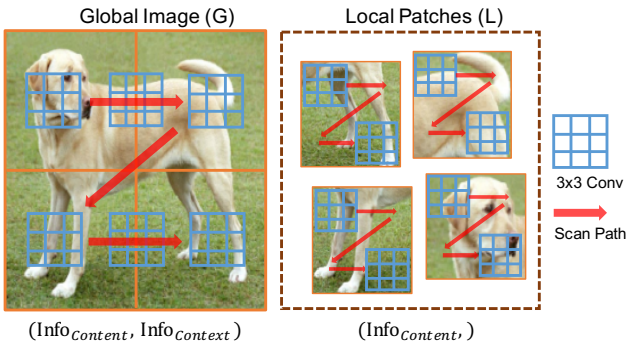


Figure 8. **Illustration of the information of global image and local patches extracted by CNN.** For global image, both content information and the context information is extracted by CNN. For local patches, only the content information is extracted by CNN.

where we use P^g, N^l, P^l, N^g to denote $\mathtt{Sim}_P^{g\leftrightarrow g}$, $\sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow l}$, $\mathtt{Sim}_P^{g\leftrightarrow l}$, $\sum_{i=1}^{K} \mathtt{Sim}_N^{g\leftrightarrow g}$ for simplicity. In summary, if we can prove $P^g \cdot N^l > P^l \cdot N^g$, then we can conclude that $\mathtt{LB}^{g\leftrightarrow l} > \mathtt{LB}^{g\leftrightarrow g}$.

Here we define $\Delta P = P^g - P^l$ and $\Delta N = N^g - N^l$, where ΔP and ΔN denotes the difference between global and local similarity. If we bring $\Delta P, \Delta N$ into Eqn. 11, we can get:

$$\begin{aligned}
\mathtt{LB}^{g\leftrightarrow l} - \mathtt{LB}^{g\leftrightarrow g} &= P^g \cdot N^l - P^l \cdot N^g \\
&= P^g \cdot (N^g - \Delta N) - (P^g - \Delta P) \cdot N^g \\
&= \Delta P \cdot N^g - \Delta N \cdot P^g
\end{aligned} \quad (12)$$

Here we can naturally know the $P^g, N^g \in (1, e)$. Then we give an empirically assumption that $\Delta P > 0$ and $\Delta N \to 0$. We verify the assumption is established in both theoretical analysis and experimental support. For experimental support, we collect the statistical data of 32000 samples, as shown in Figure 9, the experimental results match our assumption. The left figure of Figure 9 shows that two positive pair's similarity distributions $\Delta P = 0.1$. The right figure shows that two negative pair's similarity distributions $\Delta N \to 0$. We will discuss the theoretical analysis in the next paragraph. *Combine with Eqn. 12 and Figure 9*, we can easily conclude that $\Delta P \cdot N^g - \Delta N \cdot P^g > 0$, that is $\mathtt{LB}^{g\leftrightarrow l} > \mathtt{LB}^{g\leftrightarrow g}$.

Discussion. In the main paper Section 3.2.3, we briefly define the information of one image $\mathtt{Info}_I = (\mathtt{Info}_{content}, \mathtt{Info}_{context})$ and $\mathtt{Info}_P$ is reduced because a set of randomly-jittered local patches $\mathbf{P}$ would lose the contextual information compared to the global image I. As shown in Figure 8, the left figure demonstrates that the scan path of a 3×3 CNN kernel on a global image. It clearly extracts both the content information of the image and the contextual information between patches. Meanwhile, the

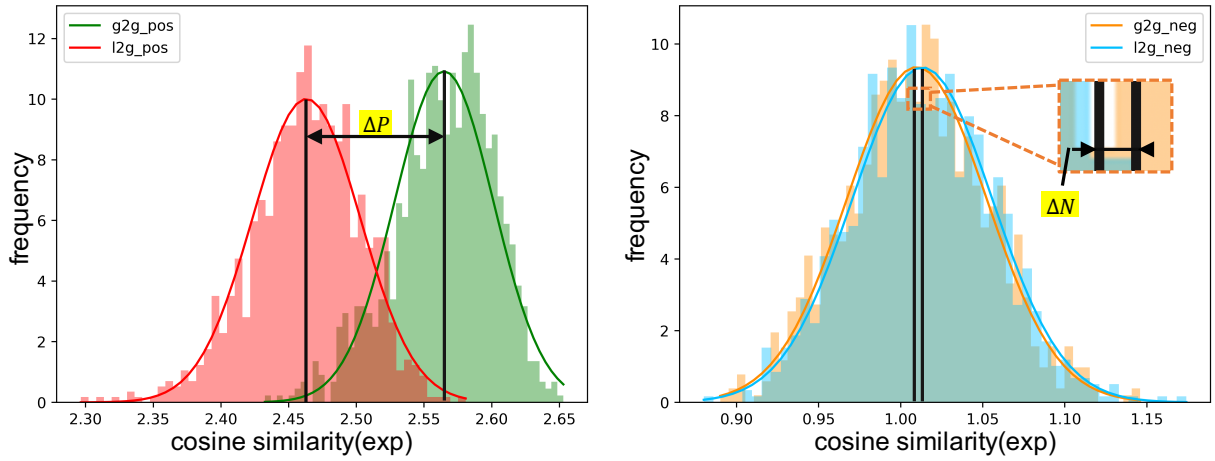


Figure 9. **The statistical distribution of exponential cosine similarity between positive pairs and negative pairs.** The left figure shows the similarity between positive pairs for global↔global and local↔glocal contrasts. The right figure shows the similarity between negative pairs.

right figure shows the scan path of a 3×3 CNN kernel on a series of individual local patches. The contextual information would lose by transforming one image to several random-jittered local patches because CNN fails to extract the information cross patches. As a result, the information of local patches is lower than that of a global image.

Intuitively, in a contrastive learning framework, the similarity between two global images would be largely higher than the similarity between a global image and a set of random local patches *for positive samples*, because the local patches lose the contextual information. However, *for negative samples*, we assume that the similarity of two global images and local patches only has a minor gap because, for CNN, it is much easy to distinguish two different images regardless of contextual information. We conduct experiments to obtain the statistics results of the similarity between positive / negative samples for global↔global and global↔local contrastive loss. In detail, we calculate the cosine similarity of 1000 batches; in each batch, there are 32 positive pairs; we calculate the mean similarity of each batch and show the results in Figure 9.

In summary, the Figure 8 and 9 give the theoretical explanation and experimental support that the global↔local contrastive loss improves the lower bound of mutual information, leading to better feature representation.

6.5. More Implementation Details

First, we provide a pseudo code for the DetCo training loop in Pytorch style, as shown in Algorithm 1. We use Apex [4] for mixed-precision training to speed up the training process. Most of our training hyperparameters are directly taken from MoCo [19]. The loss weight for intermediate contrastive loss is 1,0.7,0.4,0.1 for `Res5`, `Res4`, `Res3`, `Res2` as default. The learning rate for pre-training is 0.06 with the cosine decay schedule. For each intermediate layer and global-to-local contrast, we use a 2-layer multi-layer perceptron(MLP) head that projects the feature to a 128-D space. The design of MLP is the same as MoCo v2, except for the input channel. We also build an individual memory bank for each head to store the negative samples.

Downstream tasks. *(1) DensePose.* In the main paper, we report the Densepose results. For the Densepose task, MoCo used Detectron2 to evaluate Densepose. However, we find the Detectron2 updated recently, and the performance is higher than MoCo. So we re-finetuned all the methods use the latest Detectron2 code. We fine-tune Densepose RCNN with 26k iterations for all methods and report the results of "densepose_gps". *(2) RetinaNet.* We use ResNet50 as the backbone. We follow the setting of MoCo on Mask RCNN, adding extra normalization layers in both backbone and fpn. *(3) other method.* For SwAV [3], we download the pre-trained weights from the official code [5]. For a fair comparison, we choose the SwAV with batch size 256, and the training epochs are 200. The corresponding ImageNet linear classification is 72.7% in Top-1 accuracy. Then we fine-tuned the weights on PASCAL VOC, increasing the learning rate from 0.01 to 0.1, and set the `warmup_factor`=0.333 for 1000 iterations. The above settings the same with the SwAV [3] paper.

[4] https://github.com/NVIDIA/apex

[5] https://github.com/facebookresearch/swav

Algorithm 1 Pseudocode of Intermediate Contrastive Loss.

```
# net_q: encoder for query
# net_q = {backbone_q, head_q_list=[head 2-5]},

# net_k: same structure as net_q, encoder for key
# queue_list: a list of queues 2-5 of K keys (CxK)
# m: momentum
# t: temperature
# x: input image
# w: loss weight for 2-5
# loss_nce: contrastive loss

net_q.params = net_k.params # initialize
x_q = aug(x) # a randomly augmented version
x_k = aug(x) # another randomly augmented version

list_q = backbone_q.forward(x_q) # list of queries
list_k = backbone_k.forward(x_k) # list of keys

total_loss = 0 # weight-sum of loss from 4 stages
for i in range(4): # loop through 4 stages
   # feature, queue and weight of current stage
   q = list_q[i] # queries: NxC
   k = list_k[i] # keys: NxC
   queue = queue_list[i] # dictionary of K keys (CxK)
   weight = w[i]

   # forward mlp head
   q = head_q_list[i].forward(q)
   k = head_k_list[i].forward(k).detach()

   # calculate loss
   loss = loss_nce(q, k, queue, t)
   total_loss = total_loss+loss*weight

   # update the queue
   dequeue_and_enqueue(queue, k)

# SGD update: query network
total_loss.backward()
update(net_q.params)

# momentum update: key network
net_k.params = m*net_k.params+(1-m)*net_q.params
```

`2-5`: 4 different stages of backone, termed res2, res3, res4, res5.

6.6. More Ablation Studies

Weight of Hierarchical Intermediate Loss. We find that the transfer detection performance is highly sensitive to the loss weight hyper-parameter for hierarchical intermediate loss, so we set different weights for `Res2`, `Res3`, `Res4`, `Res5`. The result is shown in Table 15. As shown in Table 15 (a), if we set the loss weight equals (0,0,0,1), our method degenerates to MoCo. In Table 15 (b)(c), we find that directly adding hierarchical intermediate loss with inappropriate weights leads to negative results. We find that the shallow layers and the deep layers are in a competitive relationship. In other words, if we set loss weight (1,1,1,1) equally, the discriminative ability of deep layers are large negatively influenced by shallow layers. Here we revisit PSPNet [41], which also use the shallow feature as the auxiliary loss. In PSPNet, the loss weights of the shallow and deep feature are (0.4,1). In DetCo, we set the loss weight to (0.1,0.4,0.7,1.0), as shown in Table 15 (d)(e), and we find this loss weight improves the transfer detection performance. We argue that making the shallow layer's weight equal to deep layers is too aggressive to optimize. Moreover, if we normalize the weight, making the sum of loss weight equals to 1, the accuracy also drops.

	weight	AP	AP_{50}	AP_{75}
(a)	(0,0,0,1)	56.3	81.8	62.1
(b)	(1,1,1,1)+Norm	55.1	80.4	60.4
(c)	(1,1,1,1)	55.8	81.6	62.2
(d)	(0.1,0.4,0.7,1)+Norm	56.5	82.2	62.7
(e)	(0.1,0.4,0.7,1)	57.0	82.2	63.1

Table 15. Ablation study of intermediate loss weight, under 100 epoch pre-training. "Norm" means normalized the loss weights, making the sum of weights equals 1.0.

	share queue	AP	AP_{50}	AP_{75}
(a)	✓	56.2	81.8	62.3
(b)	×	57.0	82.2	63.1

Table 16. Ablation study of share or not share queue, under 100 epoch pre-training.

Memory Banks for Hierarchical Intermediate Loss. Original MoCo only utilizes the final feature to calculate contrastive loss, so MoCo only uses one memory bank (queue) to store negative samples. However, here we utilize `Res2`, `Res3`, `Res4`, `Res5` to calculate the multi-level contrastive loss. An intuitive idea is we also use one memory bank to store negative samples from four stages. In this way, one memory bank is shared for both high-level and low-level features. However, we find sharing memory bank leads to performance drop. So we build an individual memory bank for the feature from each level. Under this setting, the performance improves. The results are shown in Table 16. We consider if sharing memory bank cross levels, the positive samples of each stage need to discriminate negative samples from all levels, which is challenging to optimize. Suppose each layer owns an individual memory bank. In that case, each stage's positive samples only need to discriminate negative samples from its corresponding layer, making the network converge easy.

References

[1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 7

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1, 2

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 1, 2, 3, 7, 8, 14

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 3, 8

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2, 3, 6, 7, 8, 10

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR*, 2020. 6

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[9] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 1, 2, 3

[10] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017. 1, 2

[11] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 1, 2

[12] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pages 10542–10552, 2019. 1, 2

[13] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014. 2

[14] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 1, 2

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1, 2, 3

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2, 8

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 1, 2, 8

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2, 3, 4, 6, 7, 8, 9, 10, 14

[20] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision*, pages 4918–4927, 2019. 6

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3, 6

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[24] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 2

[25] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4, 6

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4, 6

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 7

[30] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 1, 6, 7, 8

[31] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 2, 8

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[34] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[35] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6181–6189, 2018. 6

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 6

[37] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *International Conference on Machine Learning*, volume 2, 2014. 2

[38] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[39] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 4, 5, 7, 8

[40] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2

[41] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1, 5, 15

[42] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012, 2019. 8