A. Additional Implementation Details

A.1. Pseudo Code for MaskCo Training Loop

We provide the pseudo code for MaskCo training loop in Algorithm 1.

A.2. Additional Training Details for Pre-training

Negative Sampling Strategy: In the pre-training stage, our default model uses $M \times N = 16 \times 31 = 496$ negative samples from the same GPU, where $M = 16$ is the number of negative boxes per image, and $N = 31$ is the number of images per GPU minus one (excluding the same image). The MaskCo(+in) model uses additional intra-image negatives, which is additional 16 negative samples from the same image so that the number of total negative samples become 512.

Pre-training on Multiple Datasets: In all pre-training datasets, including ImageNet, CC, and COCO, we use exactly the same training hyper-parameters that is tuned on the ImageNet.

A.3. Training Details for Downstream Tasks

ImageNet Linear Classification: We adopt the ImageNet Linear Classification protocol used in [6]. The features from different residual layers, from conv1, conv2, until conv5, are extracted, and additional pooling and linear layers are added on top of the extracted features. Only the additional linear layer is trainable. We call this evaluation protocol as multi-layer linear evaluation. There is another linear evaluation protocol used in MoCo [8] in which only the global average pooling features were used to train the linear classifier. We call this evaluation protocol as last-layer linear evaluation. We observe that the last-layer protocol is biased towards the ID-based methods, and SSL methods based on other pretext tasks usually do not produce the best result at the final layer. Since we intend to develop a new pretext task, we choose the multi-layer protocol to evaluate the capabilities of different pre-trained layers. Moreover, the multi-layer protocol was also used by a lot of previous works [9, 1, 5]. We use exactly the same training configurations as in [1], and no hyper-parameter tuning is performed. The initial learning rate is set to 0.1 with momentum 0.9 and weight decay 1e-4. The total fine-tuning epoch is set to 90, and the learning rate is decayed by 10 at epochs 30 and 60.

Pascal VOC Object Detection: We use the exact configurations as in [2]. The Faster RCNN with ResNet-50-R4 backbone is trained on trainval07+12 set and evaluated on test2007 set. We train for 24K iterations using SGD optimizer with batch size 16 (2 per GPU). We use the base learning rate 0.02, perform warmup for 100 iterations, and divided it by 10 at iterations 18K and 22K.

COCO Object Detection and Instance Segmentation: We use the exact configurations as in [3], which is the standard 2x schedule in [10].

B. Additional Experimental Results

The complete results of ImageNet Linear Classification: Our main paper only reports the ImageNet linear classification results of conv4 and conv5 in Table 3. For completeness, we list the complete results from all residual layers in Table 1 of this supplementary document.

More Visualization results of MPH: We also present additional visualization results of MPH in Figure 1.

Ablation on COCO Pre-Training: We report the ablation results of the mask strategy when our model is pre-trained on the COCO dataset in Table 2. The trend is almost identical to what we find in ImageNet pre-training.

---

Algorithm 1: Pseudocode for MaskCo training loop.

```python
# net_q: encoder for query image, including the backbone and MPH
# net_k: momentum encoder for key image
# head_q: projection head for query image
# head_k: projection head for key image
# m: momentum
# t: temperature
# x: input image

# generate two views, masked box, and key boxes
x_q, x_k, masked_box, key_boxes = transform(x)
q = head_q(roi_align(net_q(x_q), masked_box)) # queries: Nx1xC
k = head_k(roi_align(net_k(x_k), key_boxes)) # keys: NxKxC
k = k.detach() # no gradient to keys
pos_k = k[:, 0:1]
neg_k_inter = sample_inter(k) # inter-image negatives: NxK1xC
neg_k_intra = sample_intra(k) # intra-image negatives: NxK2xC
neg_k = cat([neg_k_inter, neg_k_intra], dim=1)

l_pos = bmm(q, pos_k.transpose(1, 2)) # positive logits: Nx1
l_neg = bmm(q, neg_k.transpose(1, 2)) # negative logits: Nx(K1+K2)
logits = cat([l_pos, l_neg], dim=1)

# MoCo contrastive loss (Positive labels at index 0).
labels = zeros(N)
loss = CrossEntropyLoss(logits/t, labels)

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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet conv1</th>
<th>conv2</th>
<th>conv3</th>
<th>conv4</th>
<th>conv5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand Init</td>
<td>11.4</td>
<td>16.2</td>
<td>13.5</td>
<td>9.1</td>
<td>6.5</td>
</tr>
<tr>
<td>Supervised</td>
<td>15.2</td>
<td>34.0</td>
<td>47.9</td>
<td>67.6</td>
<td>76.2</td>
</tr>
<tr>
<td>Relative-Pos [4]</td>
<td>14.8</td>
<td>31.3</td>
<td>45.8</td>
<td>49.3</td>
<td>40.2</td>
</tr>
<tr>
<td>Rotation-Pred [5]</td>
<td>12.9</td>
<td>34.3</td>
<td>44.9</td>
<td>55.0</td>
<td>49.1</td>
</tr>
<tr>
<td>NPID [11]</td>
<td>14.3</td>
<td>31.2</td>
<td>40.7</td>
<td>54.5</td>
<td>56.6</td>
</tr>
<tr>
<td>MoCo v2 [3]</td>
<td>14.7</td>
<td>32.8</td>
<td>45.0</td>
<td>61.6</td>
<td>66.7</td>
</tr>
<tr>
<td>SimCLR [2]</td>
<td>17.1</td>
<td>31.4</td>
<td>41.4</td>
<td>54.4</td>
<td>61.6</td>
</tr>
<tr>
<td>BYOL [7]</td>
<td>15.5</td>
<td>34.5</td>
<td>47.2</td>
<td>62.8</td>
<td>71.6</td>
</tr>
<tr>
<td>MaskCo</td>
<td>15.4</td>
<td>33.6</td>
<td>45.8</td>
<td>59.6</td>
<td>65.1</td>
</tr>
</tbody>
</table>

Table 1. The complete ImageNet linear classification results of Table 3 of the main paper.

<table>
<thead>
<tr>
<th>Mask</th>
<th>Pascal VOC AP</th>
<th>AP50</th>
<th>AP75</th>
<th>ImageNet conv4</th>
<th>conv5</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>56.2</td>
<td>81.4</td>
<td>62.6</td>
<td>55.3</td>
<td>38.5</td>
</tr>
<tr>
<td>✓</td>
<td>54.1</td>
<td>79.5</td>
<td>59.5</td>
<td>51.8</td>
<td>51.1</td>
</tr>
</tbody>
</table>

Table 2. Ablation studies on the mask strategy when our model is pre-trained on the COCO dataset. MPH is not used in these experiments.

References


