Semi-Supervised 2D Human Pose Estimation Driven by Position Inconsistency Pseudo Label Correction Module

Linzhi Huang1,2 * Yulong Li2 Hongbo Tian1,2* Yue Yang2 Xiangang Li2
Weihong Deng1† Jieping Ye2

1Beijing University of Posts and Telecommunications, 2Beike
{huanglinzhi, tianhongbo, whdeng}@bupt.edu.cn
{liyulong008, yangyue092, lixiangang002, yejieping}@ke.com

Abstract

In this paper, we delve into semi-supervised 2D human pose estimation. The previous method ignored two problems: (i) When conducting interactive training between large model and lightweight model, the pseudo label of lightweight model will be used to guide large models. (ii) The negative impact of noise pseudo labels on training. Moreover, the labels used for 2D human pose estimation are relatively complex: keypoint category and keypoint position. To solve the problems mentioned above, we propose a semi-supervised 2D human pose estimation framework driven by a position inconsistency pseudo label correction module (SSPCM). We introduce an additional auxiliary teacher and use the pseudo labels generated by the two teacher model in different periods to calculate the inconsistency score and remove outliers. Then, the two teacher models are updated through interactive training, and the student model is updated using the pseudo labels generated by two teachers. To further improve the performance of the student model, we use the semi-supervised Cut-Occlude based on pseudo keypoint perception to generate more hard and effective samples. In addition, we also proposed a new indoor overhead fisheye human keypoint dataset WEPDTOF-Pose. Extensive experiments demonstrate that our method outperforms the previous best semi-supervised 2D human pose estimation method. We will release the code and dataset at https://github.com/hlz0606/SSPCM.

*This work was done when the authors were visiting Beike as interns.
†Corresponding author.

Figure 1. Performance comparison between our method SSPCM and SOTA method (DataDistill [33], DUAL [46]) on COCO [28] dataset. On the COCO dataset, using 1000, 5000, and 10000 labeled person instances, our method has increased 2.3mAP, 1.9mAP, and 1.1mAP compared with the previous method.

1. Introduction

2D human pose estimation (HPE) [4, 6, 24, 27, 30, 52] is a task to estimate all 2D keypoints of the human body from images. It is a fundamental task of action recognition [5,10,47], 3D human pose estimation [16,25,29,32,55], etc. In recent years, thanks to the development of deep learning [14, 20, 36, 39], 2D human pose estimation has made significant progress. However, the training of such a task is known to be data-hungry, where the labelling process is particularly costly and time-consuming. To solve this problem, semi-supervised 2D human pose estimation has become an important research direction. This direction focuses on how to use a small amount of labeled data and a large amount of unlabeled data to improve the performance of the model.
The current state-of-the-art semi-supervised 2D HPE model [46] is based on consistency learning. Xie et al. [46] find that by maximizing the similarity between different increments of the image directly, there would be a collapsing problem. The reason is that the decision boundary passes the high-density areas of the minor class, so more and more pixels are gradually wrongly classified as backgrounds. They proposed a simple way to solve this problem. For each unlabeled image, an easy augmentation $I_e$ and a hard augmentation $I_h$ are generated, and they are fed to the network to obtain two heatmap predictions. They use the accurate predictions on the easy augmentation to teach the network to learn about the corresponding hard augmentation. In addition, they also proposed a method that can replace EMA [12] to update the parameters of the teacher model, called Dual Network. The two models will take turns to act as teachers’ identities to generate pseudo labels, and take turns to act as students’ identities.

The previous methods [33, 46] can improve the accuracy of student models. However, the previous method has the following problems: 1) In the practical application of semi-supervised learning, large model are often used as teacher and lightweight model as student. Due to the inconsistent model structure, it is hard to use EMA to update the teacher model. When conducting interactive training between large model and lightweight model, the pseudo label of lightweight model will be used to guide large models, as shown in the (a) of Fig. 2 (i). Although this method can also improve the performance of the teacher model, it is suboptimal. 2) The noise labels will harm the model training, and the student model will overfit the noise labels (causing confirmation bias [2]). Some previous semi-supervised classification tasks [22, 37] use the confidence of classification to filter pseudo labels. There are a large number of high-quality pseudo labels in the low confidence region, as shown in Fig. 2 (ii). When the confidence threshold exceeds a certain value, the higher the confidence threshold, the lower the model performance, as shown in the Fig. 2 (iv). By observing (a) and (b) in Fig. 2 (v), we can find that (b) has a higher confidence, but it is a noise label deviating from the ground truth (outliers). Therefore, we choose to filter with a lower threshold. In addition, in recent studies [17, 23, 34, 42, 48], it has been found that the more inconsistent the prediction results of different models for the same object, the more likely the prediction results will be wrong. To solve above problems, we introduce an additional auxiliary teacher and use it to generate pseudo labels. Their parameters are updated through interactive training, which ensures the difference between the two models, as shown in the (b) of Fig. 2 (i). The structure of these two teacher models can be consistent with the student model, or they can be larger models. Two teacher models may output different results for the same image. Even in different training periods, the output results of the same model for the same image will be different. We post-process the $N$ pseudo labels output by the two teacher models in different periods to obtain $N$ prediction results (2D coordinates) of each keypoint. Then, we calculate the pixel distance between $N$ prediction results of each keypoint. We use pixel distance to characterize the degree of position inconsistency (inconsistency score). We visual the relationship between the quality of the pseudo labels and position inconsistency, as shown in Fig. 2 (iii). We select a group of pseudo labels (2 pseudo labels) with the smallest position inconsistency for ensemble to obtain the final corrected pseudo labels. In short, on the basis of filtering based on confidence, PCM...
module selects a set of pseudo labels with the least inconsistency to remove outliers. The correction of pseudo labels of PCM module is similar to ensemble learning, which can make pseudo labels smoother. It is worth mentioning that we only use student model when testing.

In addition, we also use the semi-supervised Cut-Occlude based on pseudo keypoint perception to generate more hard samples, as shown in Fig. 5. Specifically, we use the pseudo label of the teacher model to locate the center of each keypoint in the image. Then, based on this central position, we cut out the local limb image. We randomly paste the local limb image to the center of a keypoint in another image to simulate local occlusion.

Our contributions are as follows:

• We propose a semi-supervised 2D human pose estimation framework driven by a position inconsistency pseudo label correction module (SSPCM). Especially when the structure of teacher model and student model is inconsistent, it is a better solution.

• To further improve the performance of the student model, we propose the semi-supervised Cut-Occlude based on pseudo keypoint perception (SSCO) to generate more hard and effective samples.

• Extensive experiments on MPII [1], COCO [28], and AI-Challenger [43] have proved that our method outperforms the previous best semi-supervised 2D human pose estimation method, as shown in Fig. 1.

• We release a new 2D HPE dataset collected by indoor overhead fisheye camera based on the WEPDTOF [41] dataset, which is called WEPDTOF-Pose. We have conducted lots of experiments on WEPDTOF-Pose, CEPDOF [11] and BKFisheye datasets (after removing sensitive information).

2. Related Work

2D human pose estimation. 2D human pose estimation (HPE) [4,6,24,27,30,52] is one of the most important tasks in computer vision. Its purpose is to detect the keypoints of the human body from the image and predict the correct category. 2D HPE can generally be divided into two methods: top-down and bottom-up. The top-down method divides the whole task into two stages: human detection and keypoint detection. To be specific, we first use human detection to obtain human bbox, and then use human pose estimation to obtain the keypoints of each human. For example, HRNet [38] proposes a multi-scale feature fusion structure, which maintains a high-resolution representation and can achieve very good results on COCO [28] and other datasets. The bottom-up method is to first detect all the keypoints in the original image, and then assign these keypoints to the corresponding human body. For example, HigherHRNet [7] proposes to use high-resolution feature pyramids to obtain multi-scale information and uses association embedding [30] to group keypoints. However, 2D HPE needs to label the keypoints of each human body in the dataset, which is labor-intensive and expensive. Therefore, we propose a new semi-supervised 2D human pose estimation framework to mitigate this problem.

Semi supervised learning (SSL). Semi-supervised learning uses a small amount of labeled data and a large amount of unlabeled data to train the model. The current semi-supervised methods are mainly divided into SSL based on the pseudo label [22,33,45,49] and SSL based on consistency [3,21,35,37,40]. SSL based on pseudo labels generates pseudo labels for unlabeled data through pretrained models and uses these pseudo labels to further optimize the model. Consistency-based SSL enables multiple images to be obtained by different data augmentation to the same image and encourages the model to make similar predictions about them. For example, FixMatch [37] uses the model to generate pseudo labels for weakly augmented unlabeled images. Only when the model produces a prediction with high confidence will the pseudo label be retained. Then, when a strongly augmented version of the same image is input, the model is trained to predict the pseudo labels. We mainly focus on SSL based on consistency, because it has superior accuracy in the public benchmark.

Semi-supervised 2D human pose estimation. The goal of semi-supervised 2D human pose estimation is to optimize the performance of the human pose estimator using a small amount of labeled data and a large amount of unlabeled data. Xie et al. [46] find that by maximizing the similarity between different increments of the image directly, there would be a collapsing problem. They propose a Dual [46] network to solve this problem. First, the input image is augmented into a pair of hard and easy data, and the easy augmentation data is transferred to the teacher model and the hard augmentation data is transferred to the student model to keep the output of the two models consistent. In addition, they also update the parameters by letting the two models take turns playing the roles of teachers and students, which is better than using EMA [12] directly. However, they ignore the negative impact of noise pseudo labels on training. Therefore, we propose a new semi-supervised training framework and a new data augmentation method.

3. Method

In this section, we first give the definition of the semi-supervised 2D human pose estimation task (see Sec. 3.1). Then, in Sec. 3.2, we introduced a semi-supervised 2D human pose estimation framework based on the position inconsistency pseudo label correction module. Finally, we introduced the semi-supervised Cut-Occlude based on pseudo
keypoint perception in Sec. 3.3.

3.1. Problem Definition

In semi-supervised 2D human pose estimation (SSHPE), we obtained a set of labeled data \( D_l = \{(x_i^l, y_i^l)\}_i^{n_l} \) and a set of unlabeled data \( D_u = \{(x_i^u)\}_i^{n_u} \), where \( x \) and \( y \) represent images and ground truth labels, \( n_l \) represents the number of labeled data, and \( n_u \) represents the number of unlabeled data. The goal of SSHPE is to train 2D human pose estimators on labeled and unlabeled data. The loss function is as follows:

\[
L_{all} = \sum_i L(x_i^l, y_i^l) + \gamma \cdot \sum_j L(x_j^u, y_j^u)
\]

where \( x_i^l \) represents labeled data, \( y_i^l \) represents ground truth label, \( x_j^u \) represents unlabeled data, \( y_j^u \) represents pseudo label generated by teacher model, \( \gamma \) represents weight of unsupervised learning, and \( L \) represents loss.

3.2. Overview of SSPCM

Fig. 3 shows the overall framework of our SSPCM. We will introduce SSPCM in detail in this section. As described in Sec. 1, we introduced an auxiliary Network\( B (f_B^\theta) \) on the basis of the original Network\( A (f_A^\theta) \) and Network\( C (f_C^\theta) \), where \( \theta \) represents network parameters. The three models have the same network structure, but their parameters are independent. In training, the training process for each batch of data can be divided into 4 stages, as shown in Fig. 3. Next, we will introduce these 4 steps in detail. The PCM module will be introduced in Train Step 4.

**Train Step 1.** Network\( A (f_A^\theta) \), Network\( B (f_B^\theta) \) and Network\( C (f_C^\theta) \) trains on labeled data and updates parameters. The supervision losses are as follows:

\[
L_{sup} = \sum_{n \in N} \|HM_{gt}^n - HM_{A}^n\|^2 + \|HM_{gt}^n - HM_{B}^n\|^2 + \|HM_{gt}^n - HM_{C}^n\|^2
\]

where \( HM_{gt}^n \) represents the ground truth of the \( n \)th im-
NetworkA except that we need to exchange the identities of parameters. When the model is used as a student model, the additional hard data augmentation (input unlabeled data) is used as the student model (with updated parameters), and NetworkB (f_θ^B) as the teacher model (with fixed parameters), and NetworkA (f_θ^A) as the student model (with updated parameters). The consistency loss is as follows:

$$L_{unsup2} = \sum_{n \in N} \|HM_{e2-h2}^n - HM_{h2}^n\|^2$$  \hspace{1cm} (4)

where $HM_{e2-h2}^n$ represents the pseudo label generated by NetworkB ($f_θ^B$) for the nth image in the unlabeled data. $HM_{h2}^n$ represents the prediction result of NetworkA ($f_θ^A$) on the nth image output in unlabeled data.

**Train Step 4.** We take NetworkA ($f_θ^A$) and NetworkB ($f_θ^B$) as teacher models (with fixed parameters) and NetworkC ($f_θ^C$) as student models (with updated parameters). Next, we input the pseudo label $HM_{e1-h1}^n$ and $HM_{e2-h2}^n$ of the same image output by NetworkA ($f_θ^A$) and NetworkB ($f_θ^B$) in Train Step 3 and Train Step 4 into the PCM module. In addition, we also input the pseudo label $HM_{last,n}$ and $HM_{e2-h2}$ generated by NetworkA ($f_θ^A$) and NetworkB ($f_θ^B$) on this image in the last epoch into the PCM module. The PCM module is shown in Fig. 4, where $HM_{e1-h1}^n$ corresponds to Fig. 4 (b), $HM_{e2-h2}$ corresponds to Fig. 4 (d), $HM_{last,n}$ corresponds to Fig. 4 (a), and $HM_{e2-h2}$ corresponds to Fig. 4 (c). Since the output results of the same model in two epochs may be similar, we only calculate the position inconsistency between different models. We first post-process the generated pseudo label $HM$ to obtain pseudo keypoint coordinates. Then, we calculate the pixel distance between different pseudo keypoints. We normalize it with the diagonal length of the heatmap to obtain the position inconsistency:

$$PI = \frac{\|\text{argmax}(HM_{i,k}^A) - \text{argmax}(HM_{i,k}^B)\|}{L_{HM}}$$  \hspace{1cm} (5)

where $HM_{i,k}^A$ represents the pseudo label of the Kth keypoint output by NetworkA ($f_θ^A$) in the i-th epoch, and $HM_{i,k}^B$ represents the pseudo label of the Kth keypoint output by NetworkB ($f_θ^B$) in the i-th epoch. $L_{HM}$ represents the diagonal length of the heatmap. We select a group of pseudo labels $HM_{min1}$ and $HM_{min2}$ with the smallest position inconsistency ($PI$), and conduct pseudo label fusion to obtain the corrected pseudo labels:

$$HM_{Final} = 0.5 \cdot (HM_{min1} + HM_{min2})$$  \hspace{1cm} (6)

We use the same operation as in Train Step 2 to obtain hard samples with occlusion, and pass them into the NetworkC ($f_θ^C$) to get the prediction results $HM_{h3}^n$. The consistency loss is as follows:

$$L_{unsup3} = \sum_{n \in N} \|HM_{e1-h1}^n - HM_{h3}^n\|^2$$  \hspace{1cm} (7)

The final loss function is as follows:

$$L_{Final} = L_{sup} + \beta \cdot (L_{unsup1} + L_{unsup2} + L_{unsup3})$$  \hspace{1cm} (8)
where $\beta$ represents weight of unsupervised learning.

**Test.** NetworkA and NetworkB are used to guide NetworkC training. When testing, we will only use NetworkC. Therefore, our method does not increase the number of parameters or calculations of the model.

### 3.3. Semi-Supervised Cut-Occlude Based on Pseudo Keypoint Perception

One of the main difficulties in 2D HPE is occlusion. We use the semi-supervised Cut-Occlude based on pseudo keypoint perception to provide more hard and effective samples for student models. Let’s take two images in one batch as an example, as shown in Fig. 5. First, we input image (a) into the teacher model to get pseudo labels and obtain the coordinates of each pseudo keypoint through post-processing. Next, we extract $N$ pseudo keypoint coordinates $(x_1, y_1)$ from them (assuming that $N$ is 1), and we take this coordinate as the center of the position to be pasted. Then, we input image (b) into the teacher model, and we also get $N$ pseudo keypoint coordinates $(x_2, y_2)$, which are taken as the central coordinates of the local limb image. We use this coordinate to clip a local limb image. After we get the local limb image, we will paste it to the position $(x_1, y_1)$ in the image (a), as shown in Fig. 5 (c). Finally, we input it into the student model to get the prediction results.

### 4. Experiments

#### 4.1. Datasets

**MPII** [1] and **AI-Challenger** [43]. The MPII dataset contains 25k images and 40k person instances with 16 keypoints. The AI-Challenger dataset has 210k images and 370K person instances with 14 keypoints. We use MPII as the labeled set, AI-Challenger as the unlabeled set. The metric of PCKh@0.5 [1] is reported.

**COCO** [28]. COCO dataset has 4 subsets: TRAIN, VAL, TEST−DEV and TEST−CHALLENGE. In addition, there are 123K wild unlabeled images (WILD). We randomly selected 1K, 5K and 10K labeled data from TRAIN. In some experiments, unlabeled data came from the remaining images of TRAIN. In other experiments, we used the entire TRAIN as the labeled dataset and WILD as the unlabeled dataset. The metric of mAP (Average AP over 10 OKS thresholds) [28] is reported.

**CEPDOF** [11]. This dataset is an indoor dataset collected by an overhead fisheye camera. It only contains bbox labels for human detection, without keypoint labels. We will experiment with this dataset as unlabeled data. Since the dataset is video data, and the repeatability between adjacent frames is high, we conducted 10 times down-sampling of the original dataset, and filtered person instances whose height or width is less than 50 pixels. Finally, there are 11878 person instances.

**WEPDTOF-Pose.** This dataset is a new human body keypoint dataset based on the WEPDTOF [41] dataset. We will release it soon. It is an indoor dataset collected by an indoor overhead fisheye camera. Since the WEPDTOF is video dataset, and the repeatability between adjacent frames is high, we conducted 10 times down-sampling of the original dataset, and filtered person instances whose height or width is less than 50 pixels. Then, we annotate the processed images, and there are 14 keypoints in total: right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right hip, right knee, right ankle, left hip, left knee, left ankle, head, and lower neck, as shown in Fig. 6 (Left). It consists of WEPDTOF-Pose TRAIN (4688 person instances) and WEPDTOF-Pose TEST (1179 person instances). The full amount of WEPDTOF-Pose TRAIN is used as labeled data, and the CEPDOF [11] dataset is used as unlabeled data for experiments. The metric of mAP [28] is reported. More details see supplementary material.

**BKFisheye.** A dataset of a real site scene (after removing sensitive information) consists of BKFisheye TRAIN (7330 person instances), BKFisheye TEST (2655 person instances), and BKFisheye UNLABEL (46923 person instances). This dataset doesn’t contain personal identity or other personal privacy information. We have mosaic the faces in the images. The annotation method is consistent with WEPDTOF-Pose, as shown in Fig. 6 (Right). The metric of mAP [28] is reported.
4.2. Implementation Details

Consistent with the previous work [46], we use SimpleBaseline to estimate the heatmap and ResNet [13] and HRNet [38] as backbones. The input image size is set to 256x192. COCO [28] dataset training is conducted on 4 A100 GPUs, and the batch size is 32. We use the Adam optimizer [19] to train these models. The initial learning rate is 1e-3, which decreases to 1e-4 and 1e-5 at 70 epochs and 180 epochs, respectively, with a total of 200 epochs. When testing, do not flip horizontally.

4.3. Comparison with SOTA Methods

Consistent with the previous work [46], we first use the ResNet18 [13] model to conduct experiments on the COCO [28] dataset. We used 1K, 5K, and 10K labeled data for the experiment, as shown in Table. 1. The results of supervised training using only labeled data are the worst. Our method outperforms the best semi-supervised 2D human pose estimation method in 1K, 5K, and 10K settings, and improves 2.3 mAP, 1.9 mAP, and 1.1 mAP respectively.

We evaluate the effect of using different networks in Table. 2. We use ResNet50 as the Teacher model and ResNet18 as the Student model. We find that the HRNet18 model has significantly improved performance. This is mainly because ResNet50 can provide more accurate pseudo label for ResNet18 which notably boosts its performance.

Data augmentation. Our data augmentation settings are consistent with previous work [46]. Easy data augmentation: random rotation (−30°−30°), random scale (0.75−1.25). Hard data augmentation: random rotation (−60°−60°), random scale (0.75−1.25). The random rotation range used by the fisheye dataset is (−180°−180°).

4.4. Implementation Details

Consistent with the previous work [46], we use SimpleBaseline to estimate the heatmap and ResNet [13] and HRNet [38] as backbones. The input image size is set to 256x192. COCO [28] dataset training is conducted on 4 A100 GPUs, and the batch size is 32. We use the Adam optimizer [19] to train these models. The initial learning rate is 1e-3, which decreases to 1e-4 and 1e-5 at 70 epochs and 180 epochs, respectively, with a total of 200 epochs. When testing, do not flip horizontally.

4.3. Comparison with SOTA Methods

Consistent with the previous work [46], we first use the ResNet18 [13] model to conduct experiments on the COCO [28] dataset. We used 1K, 5K, and 10K labeled data for the experiment, as shown in Table. 1. The results of supervised training using only labeled data are the worst. Our method outperforms the best semi-supervised 2D human pose estimation method in 1K, 5K, and 10K settings, and improves 2.3 mAP, 1.9 mAP, and 1.1 mAP respectively.

We evaluate the effect of using different networks in Table. 2. We use ResNet50 as the Teacher model and ResNet18 as the Student model. We find that the HRNet18 model has significantly improved performance. This is mainly because ResNet50 can provide more accurate pseudo label for ResNet18 which notably boosts its performance. As shown in the Table. 2, we can use large models as teachers to improve the performance of lightweight models, and our method goes beyond the previous semi-supervised method.

Consistent with the previous work [46], we used the complete COCO [28] TRAIN as the labeled dataset and WILD as the unlabeled data for experiments, as shown in Table. 3. It can be seen from Table. 3 that our method is always better than the best method used in different mod-
As shown in Table. 7, we carried out the hyper-Rand Augment [8] and JC [46], our SSCO shows better performance, compared with Cutout [9], Mixup [53], CutMix [50], the best method DUAL [46], which improved 1.4 mAP. In addition, we added the SSCO module to the PCM module, 1.2 mAP can be improved. We also reported the previous methods [15, 26, 38, 44, 46, 51]. In this work, we propose a new semi-supervised 2D human pose estimation method. We first introduce our proposed semi-supervised 2D human pose estimation framework driven by the position inconsistency pseudo label correction module. Then, we introduce the semi-supervised Cut-Occlude based on pseudo keypoint perception. We have conducted lots of experiments on WEPDTOF-Pose and BKFiشبه. We use the complete WEPDTOF-Pose TRAIN as the labeled dataset, and 11878 person instances in CEPDOF [11] as the unlabeled dataset for experiments, as shown in Table. 8 (Top). Then, We use the complete WEPDTOF-Pose TRAIN as the labeled dataset, and BKFiشبه UNLABEL as the unlabeled dataset for experiments, as shown in Table. 8 (Middle). In addition, we also conducted the same experiment on the BKFiشبه dataset (labeled training set is BKFiشبه TRAIN, unlabeled training set is BKFiشبه UNLABEL, and the test set is BKFiشبه TEST), as shown in Table. 8 (Down).

5. Conclusion

In this work, we propose a new semi-supervised 2D human pose estimation method. We first introduce our proposed semi-supervised 2D human pose estimation framework driven by the position inconsistency pseudo label correction module. Then, we introduce the semi-supervised Cut-Occlude based on pseudo keypoint perception. We have carried out a lot of experiments on datasets of different scenarios, which proved the effectiveness of our method. In addition, we released our code and new dataset, hoping to stimulate more people to study in this field.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Train</th>
<th>Test</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [44]</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>51.7</td>
</tr>
<tr>
<td>+Dual [46]</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>58.7</td>
</tr>
<tr>
<td>+Dual [46] +SSCO (Ours)</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.1</td>
</tr>
<tr>
<td>+PCM (Ours)</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>59.6</td>
</tr>
<tr>
<td>+PCM (Ours) +Cutout [9]</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.0</td>
</tr>
<tr>
<td>+PCM (Ours) +Mixup [53]</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>58.2</td>
</tr>
<tr>
<td>+PCM (Ours) +CutMix [50]</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.5</td>
</tr>
<tr>
<td>+PCM (Ours) +RandAugment [8]</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>59.9</td>
</tr>
<tr>
<td>+PCM (Ours) +JC [46]</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.0</td>
</tr>
<tr>
<td>+PCM (Ours) +SSCO (Ours)</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.7</td>
</tr>
<tr>
<td>+PCM (Ours) +SSCO (Ours)</td>
<td>ResNet18</td>
<td>COCO 5K</td>
<td>COCO</td>
<td>57.5</td>
</tr>
<tr>
<td>+PCM (Ours) +JC [46]</td>
<td>ResNet18</td>
<td>COCO 5K</td>
<td>COCO</td>
<td>44.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>N</th>
<th>Backbone</th>
<th>Train</th>
<th>Test</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM +SSCO</td>
<td>0</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>59.6</td>
</tr>
<tr>
<td>PCM +SSCO</td>
<td>1</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.2</td>
</tr>
<tr>
<td>PCM +SSCO</td>
<td>2</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.7</td>
</tr>
<tr>
<td>PCM +SSCO</td>
<td>3</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.4</td>
</tr>
<tr>
<td>PCM +SSCO</td>
<td>4</td>
<td>ResNet18</td>
<td>COCO 10K</td>
<td>COCO</td>
<td>60.5</td>
</tr>
</tbody>
</table>

Table 7. Hyper-parameter analysis of SSCO module. N represents the number of local limbs used.

Table 6. Ablation Study. DUAL [46] and JC [46] are the previous SOTA methods. PCM and SSCO are our methods.

Table 8. Comparison to the SOTA methods on the datasets collected by indoor overhead fisheye camera. WEPDTOF-Pose TEST and BKFiشبه TEST are used as the test set. WEPDTOF-Pose TRAIN and BKFiشبه TRAIN are used as the labeled set. CEPDOF [11] and BKFiشبه UNLABEL are used as the unlabeled set. Backbone is ResNet18 [13].

<table>
<thead>
<tr>
<th>Methods</th>
<th>Labeled Dataset</th>
<th>Unlabeled Dataset</th>
<th>AP</th>
<th>AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised [14]</td>
<td>WEPDTOF-Pose</td>
<td>—</td>
<td>49.5</td>
<td>53.4</td>
</tr>
<tr>
<td>Cons [46]</td>
<td>WEPDTOF-Pose</td>
<td>CEPDOF</td>
<td>54.6</td>
<td>58.1</td>
</tr>
<tr>
<td>Dual [46]</td>
<td>WEPDTOF-Pose</td>
<td>CEPDOF</td>
<td>55.1</td>
<td>59.0</td>
</tr>
<tr>
<td>Ours</td>
<td>WEPDTOF-Pose</td>
<td>CEPDOF</td>
<td>55.6</td>
<td>60.0</td>
</tr>
<tr>
<td>Supervised [14]</td>
<td>WEPDTOF-Pose</td>
<td>—</td>
<td>49.5</td>
<td>53.4</td>
</tr>
<tr>
<td>Cons [46]</td>
<td>WEPDTOF-Pose</td>
<td>BKFiشبه UNLABEL</td>
<td>57.2</td>
<td>61.4</td>
</tr>
<tr>
<td>Dual [46]</td>
<td>WEPDTOF-Pose</td>
<td>BKFiشبه UNLABEL</td>
<td>57.2</td>
<td>61.5</td>
</tr>
<tr>
<td>Ours</td>
<td>BKFiشبه</td>
<td>BKFiشبه UNLABEL</td>
<td>59.1</td>
<td>63.7</td>
</tr>
<tr>
<td>Supervised [14]</td>
<td>BKFiشبه</td>
<td>—</td>
<td>65.2</td>
<td>70.4</td>
</tr>
<tr>
<td>Cons [46]</td>
<td>BKFiشبه</td>
<td>BKFiشبه UNLABEL</td>
<td>68.2</td>
<td>72.7</td>
</tr>
<tr>
<td>Dual [46]</td>
<td>BKFiشبه</td>
<td>BKFiشبه UNLABEL</td>
<td>68.4</td>
<td>73.0</td>
</tr>
<tr>
<td>Ours</td>
<td>BKFiشبه</td>
<td>BKFiشبه UNLABEL</td>
<td>68.7</td>
<td>73.7</td>
</tr>
</tbody>
</table>
References


[40] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Advances in neural information processing systems, pages 1195–1204, 2017. 3


[50] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 6022–6031, 2019. 8


