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

Linzhi Huang[1, 2 *] Yulong Li[2] Hongbo Tian[1, 2*] Yue Yang[2] Xiangang Li[2]

Weihong Deng[1 †] Jieping Ye[2]

[1]Beijing University of Posts and Telecommunications, [2]Beike

{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.

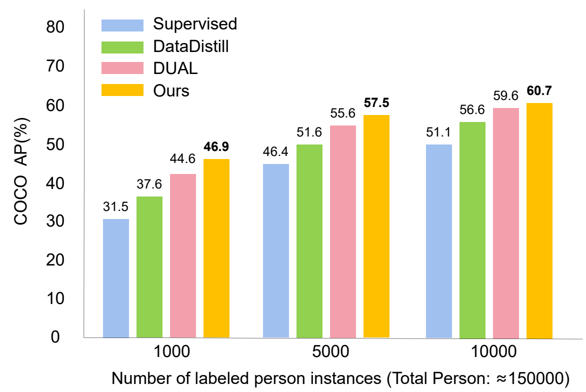

Figure 1. **Performance comparison** between our method SSPCM and SOTA method (DataDistill [33], DUAL [46]) on COCO [28] dataset. On the COCO dataset, using 1000, 5000, and 10000 labeled person instances, our method has increased 2.3mAP, 1.9mAP, and 1.1mAP compared with the previous method.

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.

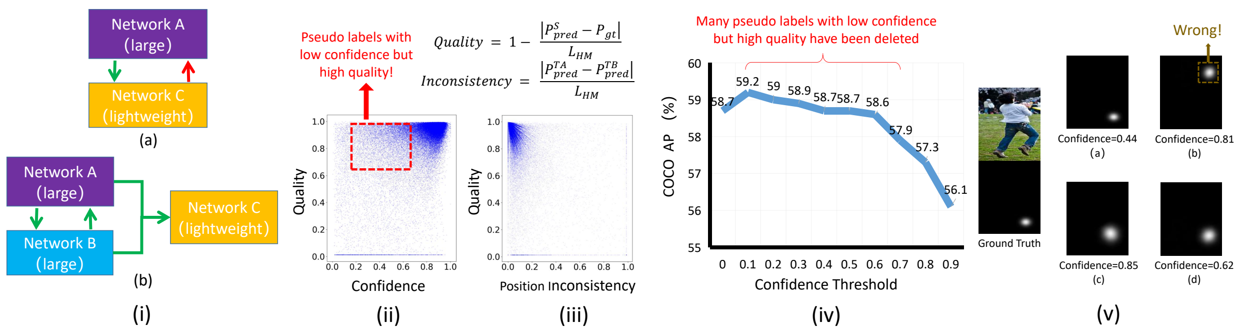

Figure 2. **Illustration of motivation.** (i): (a) Previous interactive training methods [46]. (b) Our architecture. The arrow indicates the transmission direction of the pseudo label. (ii) and (iii) are the statistics of teacher model predictions (pseudo labels) on COCO dataset. P_{pred} represents the keypoint coordinates predicted by the model (based on the heatmap). S represents student model. TA and TB represent 2 teacher models. P_{gt} represents the ground truth keypoint coordinates (based on the heatmap). L_{HM} represents the diagonal length of the heatmap. **(ii)**: The relationship between the quality of the pseudo labels and confidence. **(iii)**: The relationship between the quality of the pseudo labels and position inconsistency. **(iv)**: Results of the DUAL [46] at different confidence thresholds. **(v)**: One specific example to demonstrate that the confidence does not represent the localization quality. (a) and (b) are the output results of teacher model A in different epochs (heatmap of the right ankle). (c) and (d) are the output results of the additional auxiliary teacher model B.

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

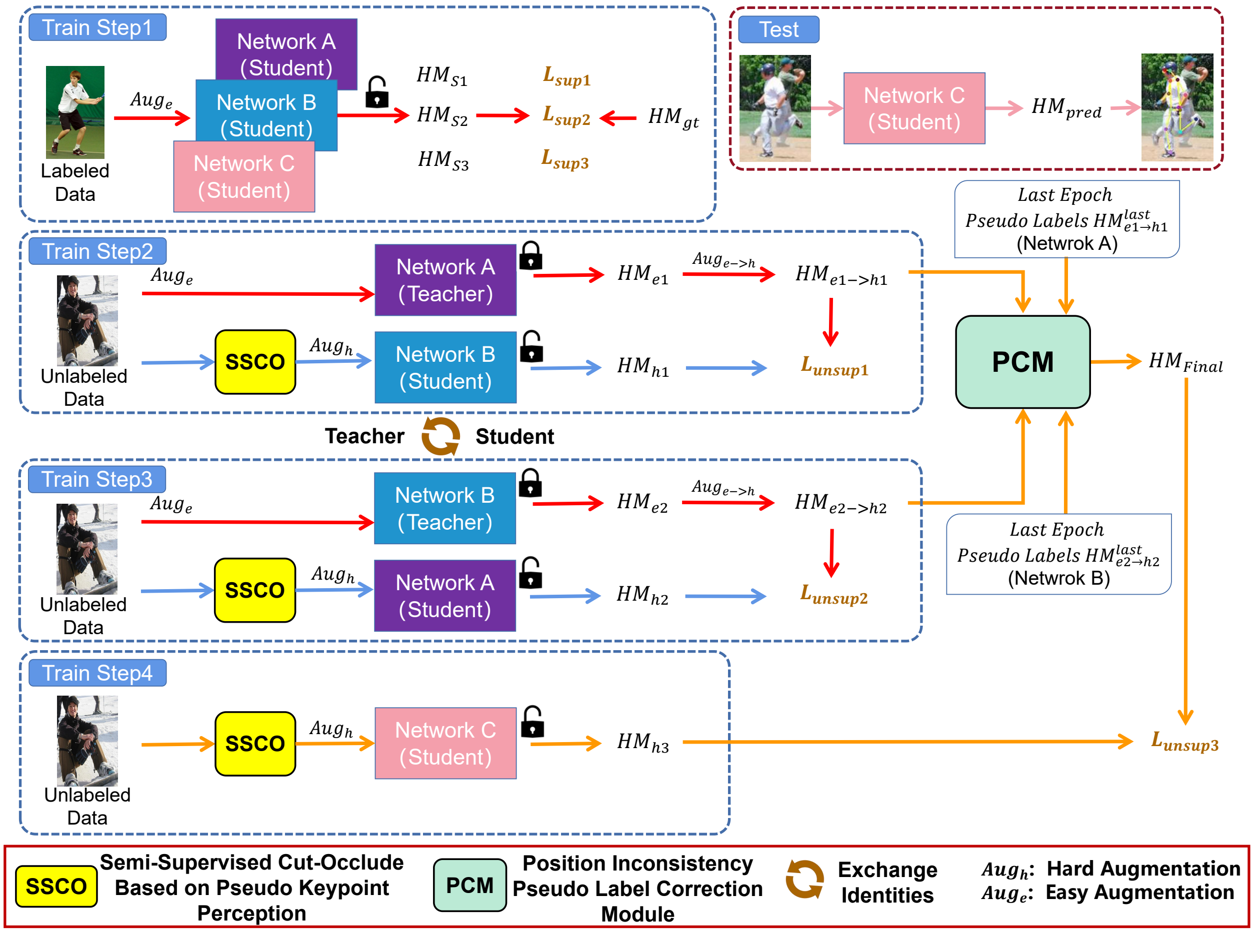

Figure 3. **Overall overview of our SSPCM.** In Train Step 1, we use labeled data to train $NetworkA$, $NetworkB$, and $NetworkC$, and update their parameters. In Train Step 2 and Train Step 3, we use unlabeled data and interactive training to update the parameters of $NetworkA$ and $NetworkB$. In Train Step 4, we use $NetworkA$, $NetworkB$, and PCM modules to guide $NetworkC$ training. SSCO is the semi-supervised Cut-Occlude based on pseudo keypoint perception. When testing, we only use $NetworkC$.

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 $D_l = \{(x_i^l, y_i^l)\}_{i=0}^{n_l}$ and a set of unlabeled data $D_u = \{(x_j^u)\}_{j=0}^{n_u}$, 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:

$$L_{all} = \sum_i L(x_i^l, y_i^l) + \gamma \cdot \sum_j L(x_j^u, y_j^u) \tag{1}$$

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, γ 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 $NetworkB$ (f_θ^B) on the basis of the original $NetworkA$ (f_θ^A) and $NetworkC$ (f_θ^C), where θ 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. $NetworkA$ (f_θ^A), $NetworkB$ (f_θ^B) and $NetworkC$ (f_θ^C) trains on labeled data and updates parameters. The supervision losses are as follows:

$$L_{sup} = \sum_{n\in N} \|HM_{gt}^n - HM_{s1}^n\|^2 + \|HM_{gt}^n - HM_{s2}^n\|^2 + \|HM_{gt}^n - HM_{s3}^n\|^2 \tag{2}$$

where HM_{gt}^n represents the ground truth of the nth im-

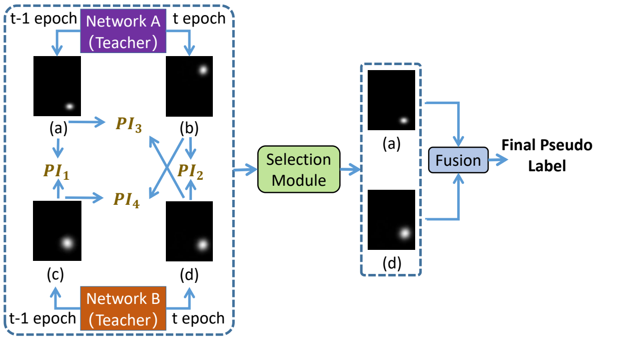

Figure 4. **Position inconsistency pseudo label correction module (PCM).** (a) Pseudo label of $NetworkA$ output in the last epoch. (b) Pseudo label of $NetworkA$ output in the current epoch. (c) Pseudo label of $NetworkB$ output in the last epoch. (d) Pseudo label of $NetworkB$ output in the current epoch. We post-process the pseudo labels to get the pseudo keypoint coordinates. Next, we calculate the position inconsistency PI between pseudo keypoints output by different models and transfer it to the Selection Module. Then, the Selection Module selects a group of pseudo labels with the smallest PI and performs pseudo labels fusion to obtain the final corrected pseudo labels.

age in the labeled data. HM^n_{s1}, HM^n_{s2} and HM^n_{s3} respectively represent the prediction results of $NetworkA$ (f^A_θ), $NetworkB$ (f^B_θ) and $NetworkC$ (f^C_θ) on the nth image in the labeled data.

Train Step 2. First, $NetworkA$ (f^A_θ) is used as the teacher model (with fixed parameters), and the additional $NetworkB$ (f^B_θ) is used as the student model (with parameter updates). Next, easy data augmentation Aug_e is performed on unlabeled data I_u, and the predicted pseudo labels HM_{e1} are obtained by inputting them into $NetworkA$ (f^A_θ). Then, input unlabeled data I_u into the SSCO module (detailed in Sec. 3.3) to get the hard sample with occlusion, and hard data augmentation Aug_h is performed on it. We input this hard sample into $NetworkB$ (f^B_θ) to get the prediction results HM_{h1}. Finally, we use $Aug_{e\to h}$ to map HM_{e1} to $HM_{e1\to h1}$ and calculate the consistency loss between $HM_{e1\to h1}$ and HM_{h1}:

$$L_{unsup1} = \sum_{n\in N} \|HM^n_{e1\to h1} - HM^n_{h1}\|^2 \tag{3}$$

where $HM^n_{e1\to h1}$ represents the pseudo label generated by $NetworkA$ (f^A_θ) for the nth image in the unlabeled data. HM^n_{h1} represents the prediction result of $NetworkB$ (f^B_θ) on the nth image output in unlabeled data. It is worth noting that when the model is used as a teacher model, the parameters are fixed. When the model is used as a student model, the parameters need to be updated.

Train Step 3. This step is similar to Train Step 2, except that we need to exchange the identities of $NetworkA$ (f^A_θ) and $NetworkB$ (f^B_θ), $NetworkB$ (f^B_θ) as the teacher model (with fixed parameters), and $NetworkA$ (f^A_θ) as the student model (with updated parameters). The consistency loss is as follows:

$$L_{unsup2} = \sum_{n\in N} \|HM^n_{e2\to h2} - HM^n_{h2}\|^2 \tag{4}$$

where $HM^n_{e2\to h2}$ represents the pseudo label generated by $NetworkB$ (f^B_θ) for the nth image in the unlabeled data. HM^n_{h2} represents the prediction result of $NetworkA$ (f^A_θ) on the nth image output in unlabeled data.

Train Step 4. We take $NetworkA$ (f^A_θ) and $NetworkB$ (f^B_θ) as teacher models (with fixed parameters) and $NetworkC$ (f^C_θ) as student models (with updated parameters). Next, we input the pseudo label $HM^n_{e1\to h1}$ and $HM^n_{e2\to h2}$ of the same image output by $NetworkA$ (f^A_θ) and $NetworkB$ (f^B_θ) in Train Step 3 and Train Step 4 into the PCM module. In addition, we also input the pseudo label $HM^{last,n}_{e1\to h1}$ and $HM^{last,n}_{e2\to h2}$ generated by $NetworkA$ (f^A_θ) and $NetworkB$ (f^B_θ) on this image in the last epoch into the PCM module. The PCM module is shown in Fig. 4, where $HM^n_{e1\to h1}$ corresponds to Fig. 4 (b), $HM^n_{e2\to h2}$ corresponds to Fig. 4 (d), $HM^{last,n}_{e1\to h1}$ corresponds to Fig. 4 (a), and $HM^{last,n}_{e2\to h2}$ 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:

$$PI = \frac{\|argmax(HM^A_{i,k}) - argmax(HM^B_{j,k})\|}{L_{HM}} \tag{5}$$

where $HM^A_{i,k}$ represents the pseudo label of the Kth keypoint output by $NetworkA$ (f^A_θ) in the ith epoch, and $HM^B_{j,k}$ represents the pseudo label of the Kth keypoint output by $NetworkB$ (f^B_θ) in the jth 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:

$$HM_{Final} = 0.5 \cdot (HM_{min1} + HM_{min2}) \tag{6}$$

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

$$L_{unsup3} = \sum_{n\in N} \|HM^n_{Final} - HM^n_{h3}\|^2 \tag{7}$$

The final loss function is as follows:

$$L_{Final} = L_{sup} + \beta \cdot (L_{unsup1} + L_{unsup2} + L_{unsup3}) \tag{8}$$

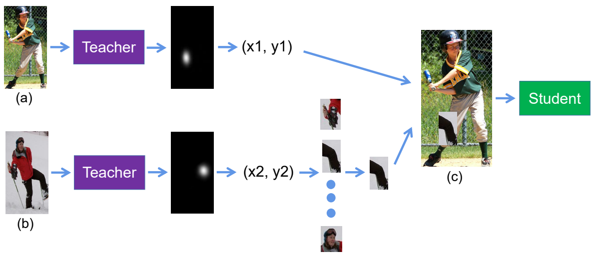

Figure 5. Semi-supervised Cut-Occlude based on pseudo keypoint perception (SSCO). (a) and (b) are two images in one batch. (c) is a hard sample with occlusion.

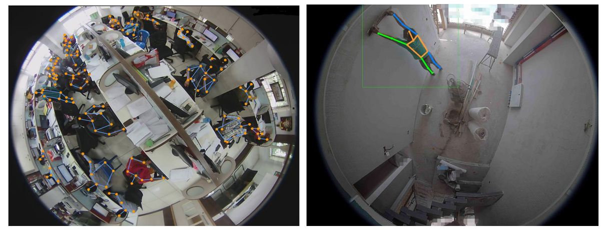

Figure 6. **Left**: One image in the WEPDTOF-Pose dataset. **Right**: One image in the BKFisheye dataset.

where β 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 $(x1, y1)$ 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 $(x2, y2)$, 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 $(x1, y1)$ 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.

Table 1. AP of different methods on COCO [28] when different numbers of labels are used. Backbone is ResNet18 [13].

Methods	1K	5K	10K	All
Supervised [44]	31.5	46.4	51.1	67.1
PseudoPose [46]	37.2	50.9	56.0	—
DataDistill [33]	37.6	51.6	56.6	—
Cons [46]	42.1	52.3	57.3	—
Dual [46]	44.6	55.6	59.6	—
Ours	**46.9**	**57.5**	**60.7**	—

Table 2. The effects of using different network structures for the two models ($Teacher$ and $Student$) on COCO [28]. We report the AP of the **student model**.

Methods	Teacher	Student	1K	5K	10K
Supervised [44]	—	ResNet18	31.5	46.4	51.1
Supervised [44]	—	ResNet50	34.4	50.3	56.3
DUAL [46]	ResNet18	ResNet18	44.6	55.6	59.6
DUAL [46]	ResNet50	ResNet50	48.7	61.2	65.0
DUAL [46]	ResNet50	ResNet18	47.2	57.2	60.4
Ours	ResNet18	ResNet18	46.9	57.5	60.7
Ours	ResNet50	ResNet50	49.4	61.6	65.4
Ours	ResNet50	ResNet18	**48.3**	**58.9**	**61.9**

Table 3. Results on the COCO [28] VAL set when all images from the $TRAIN$ set are used as the labeled set and all images from the $WILD$ set are used as the unlabeled set.

Method	Network	AP	Ap .5	AR	AR .5
Supervised [44]	ResNet50	70.9	91.4	74.2	92.3
Dual [46]	ResNet50	73.9	92.5	77.0	93.5
Ours	**ResNet50**	**74.2**	**92.7**	**77.2**	**93.8**
Supervised [44]	ResNet101	72.5	92.5	75.6	93.1
Dual [46]	ResNet101	75.3	93.6	78.2	94.1
Ours	**ResNet101**	**75.5**	**93.8**	**78.4**	**94.2**
Supervised [44]	ResNet152	73.2	92.5	76.3	93.2
Dual [46]	ResNet152	75.5	93.6	78.5	94.3
Ours	**ResNet152**	**75.7**	**93.7**	**78.6**	**94.5**
Supervised [38]	HRNetW48	77.2	93.5	79.9	94.1
Dual [46]	HRNetW48	79.2	94.6	81.7	95.1
Ours	**HRNetW48**	**79.4**	**94.8**	**81.9**	**95.2**

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 90 epochs, respectively, with a total of 100 epochs. When using the complete COCO [28] dataset, the initial learning rate is 1e-3, and it drops to 1e-4 and 1e-5 at 300 epochs and 350 epochs, respectively, with a total of 400 epochs. The fisheye dataset is trained on 1 A100 GPU, and the batch size is 32. We use the Adam optimizer to train these models. The initial learning rate is 1e-3, which decreases to 1e-4 and 1e-5 at 140 epochs and 180 epochs, respectively, with a total of 200 epochs. When testing, do not flip horizontally.

Table 4. Comparison to the SOTA methods on the COCO [28] $TEST-DEV$ dataset. The COCO $TRAIN$ set is the labeled set and COCO $WILD$ set is the unlabeled set. The person detection results are provided by SimpleBaseline (with flipping strategy).

Method	Network	Input Size	AP	AR
SB [44]	ResNet50	256 × 192	70.2	75.8
HRNet [38]	HRNetW48	384 × 288	75.5	80.5
MSPN [26]	ResNet50	384 × 288	76.1	81.6
DARK [51]	HRNetW48	384 × 288	76.2	81.1
UDP [15]	HRNetW48	384 × 288	76.5	81.6
DUAL [46] (+DARK)	HRNetW48	384 × 288	77.2	82.2
Ours (+DARK)	**HRNetW48**	**384 × 288**	**77.5**	**82.4**

Table 5. Comparisons on the MPII [1] test set (PCKh@0.5). Our method uses HRNetW32 [38] as backbone and size is 256 × 256. The MPII and AIC (w/o labels) [43] dataset are used for training.

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	Total
Newell et al. [31]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Xiao et al. [44]	98.5	96.6	91.9	87.6	91.1	88.1	84.1	91.5
Ke et al. [18]	98.5	96.8	92.7	88.4	90.6	89.4	86.3	92.1
Sun et al. [38]	98.6	96.9	92.8	89.0	91.5	89.0	85.7	92.3
Zhang et al. [54]	98.6	97.0	92.8	88.8	91.7	89.8	86.6	92.5
Xie et al. [46]	98.7	97.3	93.7	90.2	92.0	90.3	86.5	93.0
Ours	**98.7**	**97.5**	**94.0**	**90.6**	**92.5**	**91.1**	**87.1**	**93.3**

Data augmentation. Our data augmentation settings are consistent with previous work [46]. Easy data augmentation: random rotation ($-30°\sim30°$), random scale ($0.75\sim1.25$). Hard data augmentation: random rotation ($-60°\sim60°$), random scale ($0.75\sim1.25$). The random rotation range used by the fisheye dataset is ($-180°\sim180°$).

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 ResNet18 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-

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

Method	Backbone	Train	Test	AP
Baseline [44]	ResNet18	COCO 10K	COCO	51.1
+DUAL [46]	ResNet18	COCO 10K	COCO	58.7
+DUAL [46] +JC [46]	ResNet18	COCO 10K	COCO	59.6
+DUAL [46] +SSCO(Ours)	ResNet18	COCO 10K	COCO	60.1
+PCM(Ours)	ResNet18	COCO 10K	COCO	59.6
+PCM(Ours) +Cutout [9]	ResNet18	COCO 10K	COCO	60.0
+PCM(Ours) +Mixup [53]	ResNet18	COCO 10K	COCO	58.2
+PCM(Ours) +CutMix [50]	ResNet18	COCO 10K	COCO	60.5
+PCM(Ours) +Rand Augment [8]	ResNet18	COCO 10K	COCO	59.9
+PCM(Ours) +JC [46]	ResNet18	COCO 10K	COCO	60.0
+PCM(Ours) +SSCO(Ours)	**ResNet18**	**COCO 10K**	**COCO**	**60.7**
+DUAL [46] +JC [46]	ResNet18	COCO 5K	COCO	55.6
+PCM(Ours) +SSCO(Ours)	**ResNet18**	**COCO 5K**	**COCO**	**57.5**
+DUAL [46] +JC [46]	ResNet18	COCO 1K	COCO	44.6
+PCM(Ours) +SSCO(Ours)	**ResNet18**	**COCO 1K**	**COCO**	**46.9**

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

Method	N	Backbone	Train	Test	AP
+PCM +SSCO	0	ResNet18	COCO 10K	COCO	59.6
+PCM +SSCO	1	ResNet18	COCO 10K	COCO	60.2
+PCM +SSCO	**2**	**ResNet18**	**COCO 10K**	**COCO**	**60.7**
+PCM +SSCO	3	ResNet18	COCO 10K	COCO	60.4
+PCM +SSCO	4	ResNet18	COCO 10K	COCO	60.5

els. Compared with the supervised training only using the labeled part, the improvement is obvious. We also reported our results on the COCO [28] $TEST-DEV$ dataset, as shown in Table. 4. It can be seen that although the previous methods have achieved high performance, our methods can still improve performance on this basis and outperform the previous methods [15, 26, 38, 44, 46, 51]. In addition, we also experimented with the MPII [1] training set as a labeled dataset and the AIC [43] dataset as an unlabeled dataset, as shown in Table. 5. The results of our method on the MPII [1] test set exceed those of previous methods [18, 31, 38, 44, 46, 54].

4.4. Ablation Study

To study the effectiveness of our method, we have conducted a large number of ablation studies. We conducted the experiment under the setting of COCO 10K. The model used is ResNet18 [13].

Effect of the PCM. As shown in Table. 6, compared with the baseline, the results obtained by using the PCM module increase 8.4 mAP. Then, we compared the results of using the DUAL [46] module and the PCM module alone. Our method is 0.8 mAP higher than the best method before.

Effect of the SSCO. As shown in Table. 6, by adding the SSCO module to the PCM module, 1.2 mAP can be improved. We also added the SSCO module to the previous best method DUAL [46], which improved 1.4mAP. In addition, compared with Cutout [9], Mixup [53], CutMix [50], Rand Augment [8] and JC [46], our SSCO shows better performance. As shown in Table. 7, we carried out the hyper-parameter analysis of the SSCO module. When we use 2 local limb images, we get the best results.

By observing the table. 6, we can see that the performance is improved by adding **PCM**, **SSCO** modules, which verifies the effectiveness of these modules. The less labeled data we use, the more obvious our method will be.

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

Methods	Labeled Dataset	Unlabeled Dataset	AP	AR
Supervised [44]	WEPDTOF-Pose	—	49.5	53.4
Cons [46]	WEPDTOF-Pose	CEPDOF	54.6	58.1
Dual [46]	WEPDTOF-Pose	CEPDOF	55.1	59.0
Ours	**WEPDTOF-Pose**	**CEPDOF**	**55.6**	**60.0**
Supervised [44]	WEPDTOF-Pose	—	49.5	53.4
Cons [46]	WEPDTOF-Pose	BKFisheye UNLABEL	57.2	61.4
Dual [46]	WEPDTOF-Pose	BKFisheye UNLABEL	57.2	61.5
Ours	**WEPDTOF-Pose**	**BKFisheye UNLABEL**	**59.1**	**63.7**
Supervised [44]	BKFisheye	—	65.2	70.4
Cons [46]	BKFisheye	BKFisheye UNLABEL	68.2	72.7
Dual [46]	BKFisheye	BKFisheye UNLABEL	68.4	73.0
Ours	**BKFisheye**	**BKFisheye UNLABEL**	**68.7**	**73.7**

4.5. Results on WEPDTOF-Pose and BKFisheye Datasets

To further verify the effectiveness of our method, we have conducted lots of experiments on WEPDTOF-Pose and BKFisheye. 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 BKFisheye $UNLABEL$ as the unlabeled dataset for experiments, as shown in Table. 8 (Middle). In addition, we also conducted the same experiment on the BKFisheye dataset (labeled training set is BKFisheye $TRAIN$, unlabeled training set is BKFisheye $UNLABEL$, and the test set is BKFisheye $TEST$), 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.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 3, 6, 7, 8

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 2

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019. 3

[4] Yanrui Bin, Xuan Cao, Xinya Chen, Yanhao Ge, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, Changxin Gao, and Nong Sang. Adversarial semantic data augmentation for human pose estimation. In *European conference on computer vision*, pages 606–622. Springer, 2020. 1, 3

[5] Jinmiao Cai, Nianjuan Jiang, Xiaoguang Han, Kui Jia, and Jiangbo Lu. Jolo-gcn: mining joint-centered light-weight information for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2735–2744, 2021. 1

[6] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018. 1, 3

[7] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5386–5395, 2020. 3

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 8

[9] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 8

[10] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022. 1

[11] Zhihao Duan, Ozan Tezcan, Hayato Nakamura, Prakash Ishwar, and Janusz Konrad. Rapid: Rotation-aware people detection in overhead fisheye images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 636–637, 2020. 3, 6, 8

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 2, 3

[13] 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. 7, 8

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 1

[15] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5700–5709, 2020. 7, 8

[16] Linzhi Huang, Jiahao Liang, and Weihong Deng. Dh-aug: Dh forward kinematics model driven augmentation for 3d human pose estimation. *arXiv preprint arXiv:2207.09303*, 2022. 1

[17] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 784–799, 2018. 2

[18] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In *European Conference on Computer Vision*, pages 713–728, 2018. 7, 8

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learn-ing Representations*, 2015. 7

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[21] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. 3

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

[23] Gang Li, Xiang Li, Yujie Wang, Shanshan Zhang, Yichao Wu, and Ding Liang. Pseco: Pseudo labeling and consistency training for semi-supervised object detection. *arXiv preprint arXiv:2203.16317*, 2022. 2

[24] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11025–11034, 2021. 1, 3

[25] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 13147–13156, 2022. 1

[26] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019. 7, 8

[27] Zheng Li, Jingwen Ye, Mingli Song, Ying Huang, and Zhigeng Pan. Online knowledge distillation for efficient pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11740–11750, 2021. 1, 3

[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. 1, 3, 6, 7, 8

[29] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017. 1

[30] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *Advances in neural information processing systems*, 30, 2017. 1, 3

[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499, 2016. 7, 8

[32] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 1

[33] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4119–4128, 2018. 1, 2, 3, 7

[34] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021. 2

[35] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 1171–1179, Red Hook, NY, USA, 2016. Curran Associates Inc. 3

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[37] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Neural Information Processing Systems*, 2020. 2, 3

[38] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 3, 7, 8

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[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

[41] Ozan Tezcan, Zhihao Duan, Mertcan Cokbas, Prakash Ishwar, and Janusz Konrad. Wepdtof: A dataset and benchmark algorithms for in-the-wild people detection and tracking from overhead fisheye cameras. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 503–512, 2022. 3, 6

[42] Zhenyu Wang, Yali Li, Ye Guo, Lu Fang, and Shengjin Wang. Data-uncertainty guided multi-phase learning for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2021. 2

[43] J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu, Y. Wang, and Y. Wang. Large-scale datasets for going deeper in image understanding. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1480–1485, 2019. 3, 6, 7, 8

[44] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision*, pages 466–481, 2018. 7, 8

[45] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 3

[46] Rongchang Xie, Chunyu Wang, Wenjun Zeng, and Yizhou Wang. An empirical study of the collapsing problem in semi-supervised 2d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11240–11249, 2021. 1, 2, 3, 7, 8

[47] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 1

[48] Lei Yang, Xinyu Zhang, Li Wang, Minghan Zhu, Chuang Zhang, and Jun Li. Mix-teaching: A simple, unified and effective semi-supervised learning framework for monocular 3d object detection. *arXiv preprint arXiv:2207.04448*, 2022. 2

[49] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995. 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

[51] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7093–7102, 2020. 7, 8

[52] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2019. 1, 3

[53] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 8

[54] Hong Zhang, Hao Ouyang, Shu Liu, Xiaojuan Qi, Xiaoyong Shen, Ruigang Yang, and Jiaya Jia. Human pose estimation with spatial contextual information. *arXiv preprint arXiv:1901.01760*, 2019. 7, 8

[55] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11656–11665, 2021. 1