Interactive Self-Training with Mean Teachers for Semi-supervised Object Detection

Qize Yang[1], Xihan Wei[1], Biao Wang[1], Xian-Sheng Hua[1], Lei Zhang[1,2]
[1]Alibaba Group,[2]Hong Kong Polytechnic University
{qize.yqz,xihan.wxh,wb.wangbiao,xiansheng.hxs}@alibaba-inc.com, cslzhang@comp.polyu.edu.hk

Abstract

The goal of semi-supervised object detection is to learn a detection model using only a few labeled data and large amounts of unlabeled data, thereby reducing the cost of data labeling. Although a few studies have proposed various self-training-based methods or consistency regularization-based methods, they ignore the discrepancies among the detection results in the same image that occur during different training iterations. Additionally, the predicted detection results vary among different detection models. In this paper, we propose an interactive form of self-training using mean teachers for semi-supervised object detection. Specifically, to alleviate the instability among the detection results in different iterations, we propose using nonmaximum suppression to fuse the detection results from different iterations. Simultaneously, we use multiple detection heads that predict pseudo labels for each other to provide complementary information. Furthermore, to avoid different detection heads collapsing to each other, we use a mean teacher model instead of the original detection model to predict the pseudo labels. Thus, the object detection model can be trained on both labeled and unlabeled data. Extensive experimental results verify the effectiveness of our proposed method.

1. Introduction

Object detection has undergone substantial progress in recent years since the successful application of deep convolutional neural network (CNN) models. These methods generally fall into two categories: single-stage [25, 26, 20, 36] and two-stage methods [27, 9, 8, 10, 18]. However, these methods require large amounts of training samples annotated with instance-level labels, which limits their scalability. To reduce the cost of labeling, weakly supervised learning and semi-supervised learning methods have gradually attracted attention recently. Weakly supervised object detection methods [47, 31, 14, 37] require image-level annotations, while semi-supervised methods [38, 22, 29] require large quantities of unlabeled data but only a few instance-level labeled data. The weakly semi-supervised object detection methods [34, 42] use both fully labeled data and weakly labeled data. In this paper, we aim to leverage unlabeled data to further improve the object detection performance using semi-supervised learning.

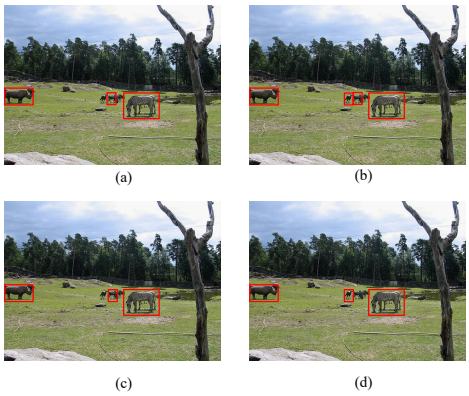


Figure 1. Illustration of our motivation: (a) and (b) show detection results from different iterations. These changes in the detection results hinder the convergence of self-training-based semi-supervised learning models; (c) and (d) show the detection results from different ROI heads.

Although semi-supervised learning has been widely explored in tasks such as image classification [1, 15, 30, 40, 44, 2, 35], only a few works [13, 32] have focused on how to apply semi-supervised learning to object detection. The challenges in applying semi-supervised learning to object detection include the fact that each image may have multiple object instances and object detection methods must regress the location for each object, as shown in Figure 1. Currently, the semi-supervised object detection methods for solving this problem can be divided into two categories: self-training-based methods and consistency regularization-based methods. The self-training-based methods [32] estimates the pseudo labels for unlabeled images using a pre-trained model and then jointly trains the model with both labeled and unlabeled data. However, the pseudo labels are generated only once, and they remain fixed during the semi-

supervised training, (i.e., incorrect pseudo labels are not corrected during the semi-supervised learning process; thus, the improvements offered by such models is limited). In contrast, the consistency-regularization-based methods [13] regularize the consistency of the outputs for the same unlabeled image under different forms of data augmentation. However, this method ignores the discrepancies among the outputs from different iterations.

To overcome the challenges mentioned above, we treat the detection results from different iterations and models as an ensemble rather than using fixed pseudo labels. This approach supports estimating up-to-date detection results for the unlabeled images in the current batch to improve the pseudo label quality during semi-supervised training. However, as illustrated in Figure 1 (a) and (b), the detection results from different iterations can be different. If we use the results directly as pseudo labels for unlabeled data, the training process would be difficult to converge. Additionally, as shown in Figure 1 (c) and (d), the detection results from different detection models (or different region-of-interest (ROI) heads) are also different, which means they may contain complementary information. Thus, the detection results of one model have the potential to improve another model.

In this paper, we propose **i**nteractive **s**elf-training with **m**ean **t**eachers for semi-supervised object detection (ISMT). Specifically, to improve the quality of pseudo labels while ensuring the model convergence during semi-supervised training, we store historical pseudo labels in memory and use nonmaximum suppression (NMS) to fuse the up-to-date detection result with the historical pseudo labels. Then, we update the pseudo label memory bank; this stored version serves as the final pseudo label for the unlabeled data. Second, we use two ROI heads with different structures to mine complementary information from the unlabeled data. Furthermore, to avoid overfitting and prevent the two ROI heads from reaching the same value, we use the mean teacher approach for each student ROI head to estimate the detection results and provide pseudo labels for the other student ROI head. Compared to the existing self-training-based object detection methods [32], our method performs interactive self-training using the mean teachers as an ensemble to combine the knowledge from different iterations and different ROI heads.

The main contributions of this work are summarized below. (1) We propose using NMS to fuse the up-to-date detection results with the history pseudo labels to improve the quality of pseudo labels and stabilize the semi-supervised training process. (2) We first propose interactive self-training for semi-supervised object detection augmented by the mean teacher approach, in which the ROI heads estimate pseudo labels for each other and learn from unlabeled data. Our proposed method achieves the state-of-the-art semi-supervised object detection performance across the MS-COCO [19] and PASCAL-VOC [6] datasets. We also provide an ablation study and a further analysis to verify the effectiveness of each proposed component.

2. Related Works

Object detection. Object detection is a fundamental computer vision task that has been extensively studied in the literature. Object detection can be divided into two categories depending on whether the method uses a region proposal network (RPN): two-stage methods [27, 9, 8, 10, 18] and one-stage methods[25, 26, 20, 16, 36]. Many of the two-stage object detection methods are based on determining a region of interest (ROI). Ren *et al*. [27] proposed Faster-RCNN, which achieved substantial improvements and provided a foundation for many subsequent research studies. However, these types of methods require large quantities of samples annotated with instance-level labels to train the detection network, which limits their scalability.

Semi-supervised learning. Semi-supervised learning (SSL) approaches have recently achieved progress in image classification with the development of deep learning. The SSL methods leverage large amounts of unlabeled data to obtain decision boundaries that better fit the underlying data structure. Consistency-regularization-based methods [15, 35, 21] apply perturbations to an input image and then minimize the differences between the output predictions. This approach does not require labeled samples because the loss is determined by the differences between the outputs. Consistency-regularization-based methods are known to help smooth the manifold. Self-training-based methods [17, 41] first train a model using supervised learning with some labeled data and then predict pseudo labels on larger amounts of unlabeled data. However, the complexities in the architectural design and multitask learning process of object detectors hinders simply transferring the existing semi-supervised techniques from the image classification task to the object detection task.

Semi-supervised object detection. Self-training [29, 32] improves model performance by utilizing high-confidence samples with pseudo labels during the training process. However, performing the data augmentations required by these methods is time consuming, and the resulting performances depend largely on the quality of the pseudo labels. In particular, Sohn *et al*. [32] proposed a self-training-based method that applied strong data augmentation to unlabeled images to avoid overfitting during training. However, we argue that such augmentations are designed on a case-by-case basis and one type may not be optimal for different scenarios and would significantly increase the difficulty of training an object detection model. On the other hand, Jeong *et al*. [13] proposed a consistency regularization-based method that regularized the consistency of the outputs from horizontally symmetric views of the unlabeled data. Existing omni-supervised learning methods [24, 28] are lower-bounded by performance on existing labeled datasets and require nu-

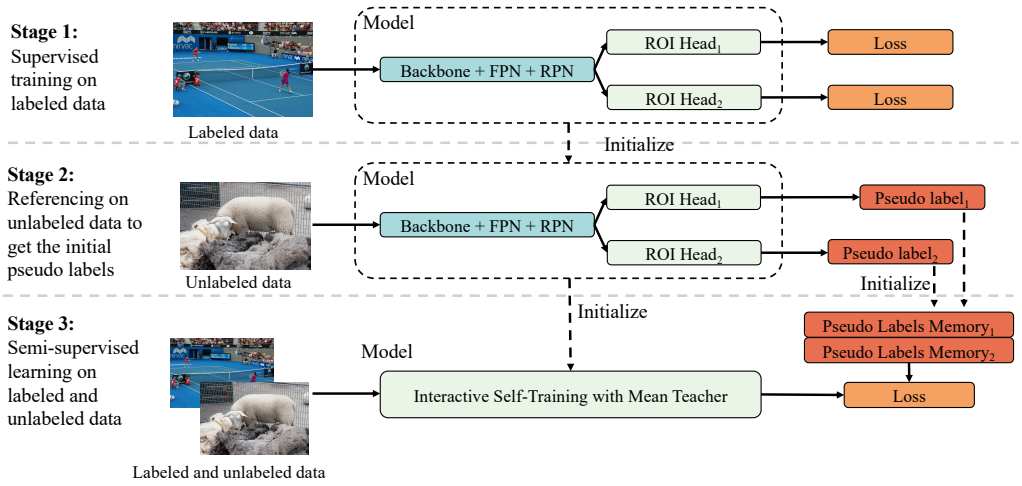


Figure 2. Illustration of our proposed method. First, we train a detection model using only labeled data; then, we use the supervised pretrained model to estimate the detection result after NMS to obtain the initial pseudo labels and form the pseudo label memory. Finally, during semi-supervised training, we use the pretrained parameters to initialize the model and then perform interactive self-training with the mean teacher method. After semi-supervised training (during inference), the stored historical pseudo labels are no longer necessary and we can empirically use only one of the ROI heads to predict the detection result.

merous labeled data and unlabeled internet-scale data.

3. Method

3.1. Preliminary

We first define some notations for the semi-supervised object detection problem. Assume we are given sets of labeled $\mathcal{X} = \{\mathbf{x}_i|_{i=1}^{N_l}\}$ and unlabeled data $\mathcal{U} = \{\mathbf{u}_i|_{i=1}^{N_u}\}$, where each labeled image has multiple objects $\{\mathbf{p}^*, \mathbf{t}^*\}$, and $\mathbf{p}^*$ and $\mathbf{t}^*$ represent the ground-truth box class labels and coordinate labels, respectively. For simplicity, we build our model based on the Faster-RCNN model [27] with feature pyramid networks (FPN) [18], which is composed of a CNN backbone, an FPN, an RPN, and ROI heads. For supervised object detection, the detection loss function can be formulated as follows:

$$\mathcal{L}_l(\mathbf{x}, \mathbf{p}^*, \mathbf{t}^*) = \sum_b \left[\frac{1}{N_c} \sum_j \mathcal{L}_{cls}(\mathbf{p}_j, \mathbf{p}^*_{j,b}) + \frac{\lambda}{N_c} \sum_j \mathcal{L}_{reg}(\mathbf{t}_j, \mathbf{t}^*_b) \right], \tag{1}$$

where b is the index of the ground-truth bounding box and j is the index of the anchor, λ denotes the weight of $\mathcal{L}_{reg}$. $\mathbf{p}_i$ is the predictive probability of an anchor being positive, and $\mathbf{t}_i$ denotes the 4-dimensional coordinates of an anchor. $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ are the classification loss and regression loss, respectively.

In this paper, we propose an interactive self-training method based on a mean teacher approach to improve the quality of pseudo labels and avoid overfitting unlabeled data. Specifically, as shown in Figure 2, we build our model based on a Faster-RCNN with two ROI heads that have different structures. First, we use labeled data to pretrain the model; then, we can estimate the detection result for unlabeled data. We set a threshold η to filter out low-quality detection results, and the remaining results form the initial pseudo labels for the unlabeled data. Finally, during semi-supervised training, two ROI heads provide pseudo labels for each other. We employ DropBlock[7] to increase the differences between the input feature maps of the different ROI heads by forcing them to focus on different parts of the feature map. To avoid the predictions of the two ROI heads from converging to each other and to leverage the ensemble of historical knowledge, we use the mean teachers of the ROI heads to estimate the pseudo labels. In the following, we first introduce how to the predicted detection results from different iterations; then, we introduce the interactive self-training process without mean teachers and subsequently describe how to apply the mean teachers to interactive self-training.

3.2. Pseudo Labels Fusion

As shown in Figure 1, the predicted detection results from different iterations are different; thus, if we were to use such unstable results directly as the pseudo labels for unlabeled data, the training process might be unstable and have difficulty converging. However, the outputs from different iterations contain various knowledge; thus, an ensemble constructed from these outputs would improve pseudo label quality.

To smooth the detection results and leverage the discrepancies among the outputs from different iterations during semi-supervised training, we propose using NMS to fuse these outputs. Specifically, we use the pretrained model to

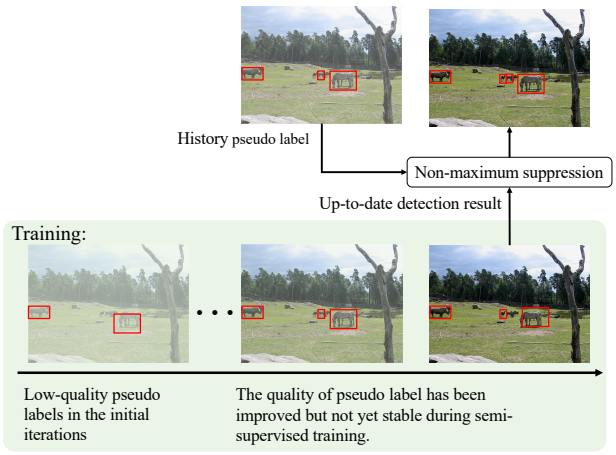


Figure 3. Illustration of pseudo label fusion. The pseudo label for semi-supervised learning is the fusion of the up-to-date detection result and the historical pseudo label. Such fusion can improve the quality and stability of pseudo labels, which is beneficial to convergence during training.

estimate the detection results for each unlabeled case; then, those results are stored in memory. Networks with memory were recently introduced that enable more powerful learning and reasoning ability for deep learning because the memory allows past knowledge to be remembered and can model the data distribution of the entire dataset [5, 39, 43]. Previous methods [5, 39] stored a feature embedding for each image and updated the feature embedding using an exponential moving average; in contrast, our method stores the detection result and updates it by nonmaximum suppression (NMS). Specifically, let $\{\bar{\mathbf{p}}, \bar{\mathbf{t}}\}$ represent the stored predicted detection result of an image in pseudo label memory, and let $\{\mathbf{p}, \mathbf{t}\}$ be the up-to-date prediction result from the network during semi-supervised training. Then, the updating process can be formulated as follows:

$$\{\hat{\mathbf{p}}, \hat{\mathbf{t}}\} = \mathrm{NMS}(\mathrm{CAT}(\{\bar{\mathbf{p}}, \bar{\mathbf{t}}\}, \{\mathbf{p}, \mathbf{t}\})), \tag{2}$$

where NMS represents the nonmaximum suppression operation and CAT represents the concatenation operation between the up-to-date detection results and the historical pseudo label. After updating, $\{\hat{\mathbf{p}}, \hat{\mathbf{t}}\}$ will be stored in memory and later used as pseudo labels for unlabeled data.

Features *vs*. labels. As mentioned above, some previous methods [5, 39, 43] use memory to track the feature embedding for each image. However, each image has multiple proposals or objects in object detection which makes the feature embeddings quite large. Moreover, the indexes of the proposals might change during the semi-supervised training process. Therefore, tracking the feature embedding of each object in images for object detection purposes is difficult. Instead, using NMS to fuse the detection results from different iterations saves memory and is more efficient.

3.3. Interactive Self-Training

Self-training has been widely used in semi-supervised learning [32, 17, 41]. However, using the pseudo labels estimated by a single ROI head to train itself tends to result in overfitting. The noisy student approach [32, 41] is a self-training extension that randomly augments the unlabeled samples and transforms its pseudo labels correspondingly; then, it uses the noisy unlabeled data and noisy pseudo labels to train a student network. This noisy student approach avoids overfitting to some extent. In this paper, we propose using two ROI heads with different structures to estimate pseudo labels for each other to further avoid overfitting during interactive self-training. Specifically, we first train a detection model with the two ROI heads of different structures using labeled data; then, we use trained model to estimate pseudo labels for the unlabeled data. Let $\{\hat{\mathbf{p}}, \hat{\mathbf{t}}\}$ represent a pseudo label from one of the ROI heads. For the other head, the loss function of the unlabeled data can be formulated as follows:

$$\mathcal{L}_u(\mathbf{u}, \hat{\mathbf{p}}, \hat{\mathbf{t}}) = \sum_b \left[\frac{1}{N_c} \sum_j \mathcal{L}_{cls}(\mathbf{p}_j, \hat{\mathbf{p}}_{j,b}) + \frac{\lambda}{N_c} \sum_j \mathcal{L}_{reg}(\mathbf{t}_j, \hat{\mathbf{t}}_b) \right], \tag{3}$$

where $\{\mathbf{p}, \mathbf{t}\}$ are the outputs of the other ROI head. To increase the discrepancies among the detection results from two ROI heads, we introduce the DropBlock [7] module to each ROI head, which randomly drops a contiguous region of a feature map. In this way, the different ROI heads observe the feature map from different views, allowing these heads to capture different key information to improve the detection results.

Our proposed method is related to deep cotraining [46] or deep mutual learning [45]. Cotraining and mutual learning refer to the output of a network as the goal by which a network wins a competition , thereby avoiding problems encountered in self-training. Deep mutual learning [45] starts with a pool of untrained students who simultaneously learn to solve the task together by training with a conventional supervised learning loss and a mimicry loss. Our method is more similar to cotraining [46, 3, 23]. Deep cotraining [23] trains multiple deep neural networks to learn different views and exploits adversarial examples to encourage these view differences to prevent the networks from collapsing into each other. In our proposed method, the two ROI heads have different structures and use DropBlock modules, enabling them to take different view feature maps as inputs. Furthermore, our model could flexibly use more ROI heads to enable interactive self-training.

3.4. Mean Teacher

In interactive self-training, each ROI head provides pseudo labels for the other ROI head. However, such a setting will eventually result in the heads collapsing into to each other because they attempt to mimic each other. Second, to ensure the stability of the estimated pseudo labels

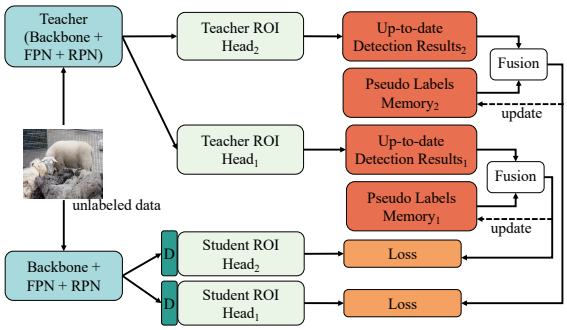


Figure 4. Illustration of interactive self-training with mean teachers. In figure, "D" means DropBlock [7]. The parameters of the teacher modules are the exponential moving averages of the parameters of their corresponding student modules.

during the training process and facilitate the optimization of the network, we introduce the concept of the mean teacher [35]. The teacher parameter is the moving average of its corresponding student parameter. That is, even for each unlabeled image, the estimated pseudo labels should have a certain consistency throughout the different epochs:

$$\boldsymbol{\theta}'^{t} = \alpha\boldsymbol{\theta}'^{t-1} + (1-\alpha)\boldsymbol{\theta}^{t}, \tag{4}$$

where α is a smoothing coefficient hyperparameter and t is the iteration, and $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are the parameters of the student model and teacher model, respectively. SWA [12] and MEA both ensemble different stages of the model to improve the performance. However, SWA is the average of the selected models, while EMA is the exponential moving average. As shown in Figure 4, the "Teacher ROI Head 1" is the exponential moving average of "Student ROI Head 1" and it estimates the up-to-date detection results for "Student ROI Head 2". Then, the up-to-date detection results are fused with the history pseudo labels, as introduced in Section 3.2. The slowly progressing teacher model can be regarded as an ensemble of student models across different training iterations. The mean teacher aggregates the information after every step instead of only after every epoch. In addition, because the weight averages improve all the layer outputs, not just the output of the top layer, the target model achieves better intermediate representations. Thus, the quality of the pseudo labels produced by the teacher models is much higher than those produced by the student models.

3.5. Overview of Our Model

As shown in Figure 2, we first train a supervised detection model using only labeled data, and then we use this initial trained model to estimate the pseudo labels for all the unlabeled data to initialize the pseudo label memory. In the semi-supervised learning stage, the model takes both labeled and unlabeled data as input. For the labeled data, the loss function is the same as during the supervised learning. For the unlabeled data, the teacher ROI head is the exponential moving average of the student ROI head, and it estimates the up-to-date detection result for the unlabeled data of the current batch; then, the up-to-date detection result is fused by NMS with the corresponding history pseudo label for the same image in the pseudo label memory to obtain the final pseudo labels. For each student ROI head, the pseudo labels are fused from those of the other teacher head to leverage the complementary information and to avoid biasing each other. The final loss function of our proposed method is as follows:

$$\mathcal{L} = \mathcal{L}_l + \gamma\mathcal{L}_u, \tag{5}$$

where γ is the weight of the loss of unlabeled data.

During semi-supervised learning, the quality of the pseudo labels significantly improves the model. In this paper, we use pseudo label fusion and mean teachers to improve the pseudo label quality. Pseudo label fusion is similar to that in the temporal ensemble [15], which maintains an exponential moving average prediction for each training example. However, in object detection, we cannot accumulate the network output directly because each image has multiple objects. Thus, we use NMS to construct a temporal ensemble of network output. However, because each target is updated only once per epoch, the learned information is incorporated into the training process at a relatively slow pace. In contrast, the mean teacher aggregates information after every step instead of after every epoch. In addition, since the weight averages improve all the layer outputs, not just the output of the top layer, the target model obtains better intermediate representations.

4. Experiments

4.1. Datasets and Evaluation

We validated the effectiveness of our proposed method on two popular public benchmarks for object detection: MS-COCO and PASCAL-VOC(07, 12) [6]. MS-COCO contains more than 118K labeled images that include approximately 850k labeled instances from 80 classes. Additionally, there are 123K unlabeled images that can be used for semi-supervised learning. PASCAL-VOC07 contains 5,011 images from 20 categories for training, while PASCAL-VOC12 contains 11,540 images . For the experiments on MS-COCO, following the example of STAC [32], we randomly sampled 1%, 2%, 5% and 10% of the labeled training data as labeled data and treated the remainder as unlabeled data. For these experiments, we created 3 data folds. Following [33], we also used all the labeled training data as labeled data and used additional unlabeled data as unlabeled data. We adopted mean average precision $AP_{50:95}$ (denoted by mAP) as the evaluation metric. For the PASCAL-VOC dataset, we used the PASCAL-VOC07 dataset as the labeled data and the PASCAL-VOC12 dataset or 20 classes of MS-COCO as the unlabeled data. We evaluated the detection performances on the VOC07 test set.

Methods	1% COCO	2% COCO	5% COCO	10% COCO	100% COCO
Supervised [32]	9.05 ± 0.16	12.70 ± 0.15	18.47 ± 0.22	23.86 ± 0.81	37.63
STAC [32]	13.97 ± 0.35	18.25 ± 0.25	24.38 ± 0.12	28.64 ± 0.21	39.21
Our baseline	9.49 ±0.32	12.86 ± 0.26	16.66 ± 0.31	24.44 ± 0.72	37.81
ISMT (Ours)	18.88 ± 0.74	22.43 ± 0.56	26.37 ± 0.24	30.53 ± 0.52	39.64

Table 1. The performance (%) of our proposed method compared with others on the COCO dataset. Here, "1% COCO" means we used 1% data of the COCO dataset as labeled data and the remaining data as unlabeled data; the notation is similar for the other datasets. "100%" means we used the entire COCO dataset as labeled data and all additional data was unlabeled data.

4.2. Implementation Details

Following STAC [32] and CSD [32], we built our model based on Faster-RCNN [27] using an FPN [18] as our object detector and a ResNet-50 [11] model as the backbone CNN. The backbone is initialized by a model pretrained on ImageNet. During the semi-supervised learning process, the confidence threshold for pseudo labels is set to 0.9 to filter out noisy detection results, and then NMS is applied to the history pseudo label, for which the IOU threshold is 0.5. The smoothing coefficient hyperparameter α and the weight of the loss for unlabeled images γ were empirically set to 0.99 and 2, respectively. In our experiments, the number of ROI heads is 2. The dropout rate and the kernel size of DropBlock are 0.3 and 3, respectively. Compared to STAC, we only use color jittering on the unlabeled data to train the student network, while the detection results are estimated by the teacher model without any augmentation. During testing, we empirically used one of the teacher ROI heads to estimate the detection results. Our implementation is based on MMDetection [4].

4.3. Comparison to the state-of-the-art

We compared our model with the state-of-the-art semi-supervised object detection method. Because only a few semi-supervised object detection methods have been proposed in recent years, we compared our method with STAC [32] on the MS-COCO dataset and PASCAL VOC dataset, and for CSD [13], we made comparisons on the PASCAL VOC dataset. The results are shown in Table 1 and Table 2. As shown in Table 1, our method significantly outperforms the baseline model that only uses the labeled data, and it achieves better scores than STAC by a large margin. Specifically, our proposed method achieves 18.95% when using only 1% of labeled data, while the supervised learning method requires 5 times that volume of labeled data (i.e., 5%) to achieve a comparable performance. This result indicates that our proposed method can effectively reduce the cost of data labeling. Even when the entire MS-COCO dataset is treated as labeled data and the additional MS-COCO unlabeled images form the unlabeled data, our method still outperforms the compared methods. Compared with existing omni-supervised learning methods [24, 28], our proposed method improves more when additional unlabeled data are used (i.e., improve by 1.83 (ours), 0.9 [24], and 1.2 [28], respectively). From Table 2, we can see that the performance of our proposed method is higher than the compared methods on $AP_{50:95}$ but slightly lower on AP_{50}. Our proposed method uses NMS to update pseudo labels during training. Thus, some low-quality pseudo labels with low IOU scores are suppressed. Additionally, the performance of the baseline model of STAC is higher than ours (i.e. 76.30 and 72.75, respectively). These experimental results verify the effectiveness of our proposed method. Our method not only leverages the complementary information from the different ROI heads but also ensembles the pseudo labels through multiple iterations, ensuring that our method improves the quality of pseudo labels and mines more information from the images. In contrast, STAC [32] fixed the pseudo labels, which means that it cannot improve the pseudo label quality during training. Compared to CSD [13], our method performs consistency regularization by regularizing the consistency between the output from the student ROI heads and the pseudo labels from the other teacher ROI head.

Labeled	Unlabeled	Methods	AP_{50}	$AP_{50:95}$
VOC07	None	Supervised	72.75	42.04
VOC07	VOC12	CSD [13]	74.70	-
		STAC [32]	77.45	44.64
		Ours	77.23	46.23
VOC07	VOC12 + COCO (20 classes)	CSD [13]	75.10	-
		STAC [32]	79.08	46.01
		Ours	77.75	49.59

Table 2. Experimental results on the PASCAL-VOC dataset

4.4. Ablation Study

We conducted an ablation study to demonstrate the effectiveness of each component (interactive self-training, mean teachers, and pseudo label fusion) of our method on the MS-COCO dataset. The experimental results are shown in Table 3.

Effectiveness of interactive self-training. Interactive self-training aims to avoid overfitting and provide complementary information during semi-supervised learning. To verify the effectiveness of interactive self-training, the ablated model has only one ROI head and it fuses the detection results from its teacher ROI head with the historical pseudo

Methods	1% COCO	10% COCO
baseline	9.49	24.44
without IST	15.94	26.75
without MT	15.23	28.33
without PLF	14.74	26.65
ISMT (Ours)	18.88	30.53

Table 3. Ablation study of out proposed method on MS-COCO dataset. Here, "IST", "MT", and "PLF" are the abbreviations for the interactive self-training, mean teacher, and pseudo label fusion, respectively.

ROI head structures	1% COCO	10% COCO
without DropBlock	17.32	29.48
One ROI head	15.64	26.47
Two ROI heads (ours)	18.88	30.53
Three ROI heads	18.53	30.74

Table 4. Analysis of the ROI head structures on the MS-COCO dataset

labels. We can see that the performance drops significantly (from 30.42 to 26.43) when training with 10% labeled data and 90% unlabeled data; nevertheless, this model is still better than the baseline model, verifying the indispensability of interactive self-training. Without interactive self-training, the model is more likely to overfit.

Effectiveness of the mean teachers T0o validate the effectiveness of the mean teachers, we removed the teacher modules from our model and directly fused the detection results from the student ROI heads with the historical pseudo labels. The results of this experiment are shown in Table 3. The performance drops by 2.20 when using 10% of the MS-COCO dataset as labeled data and the remainder as unlabeled data. Thus, the mean teacher not only effectively avoids biases from the different ROI heads but also treats the weights of different model iterations as an ensemble. Thus, the pseudo labels from the teacher models are better than those from the student models.

Effectiveness of pseudo label fusion. The detection results for the same image vary among different epochs, especially when the size of the dataset is large. Thus, if we were to use the detection result of the current epoch directly as the pseudo labels, the target would change significantly and increase the difficulty of training. As shown in Table 3, without the pseudo label fusion operation, the performance drops from 30.42% to 25.32%. In addition, the pseudo label fusion process helps filter out low-quality pseudo labels and gradually increases the number of label instances in the unlabeled images during training.

4.5. Further Analysis

Architecture with multiple ROI heads. We analyze the architecture with multiple ROI heads on the MS-COCO dataset to further mine the potential of our model. To this end, we either removed the DropBlock [7] from each ROI head or increased the number of ROI heads. As shown in Table 4, the performance improvement becomes considerably smaller as the number of ROI heads increases. Specifically, the performance drops by 1.26 when we remove the DropBlocks. The DropBlock further increases the discrepancies between the different ROI heads, causing them to mine different information from the feature map. Thus, two ROI heads are complementary to each other. Although increasing the number of ROI heads can further improve the performance, doing so would also increase the training time.

γ	0.5	1	2	4	8
10% COCO	24.82	29.42	30.53	28.94	18.10

Table 5. Analysis of the loss weight for unlabeled data on the MS-COCO dataset

α	0.5	0.9	0.99	0.999	0.9999
10% COCO	27.82	28.42	30.53	29.94	26.10

Table 6. Analysis of the EMA rate α on the MS-COCO dataset.

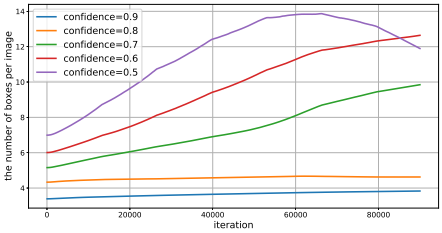
The loss weight for unlabeled samples. We varied the unsupervised loss weight γ to analyze the effect of the loss for the unlabeled data. The results are reported in Table 6. The model reaches its best performance (i.e., 30.53) when the weight of $\mathcal{L}_u$ is 2. When γ is too large, the model pays more attention to unlabeled data with noisy pseudo labels; thus, the performance falls as γ increases. On the other hand, when γ is too small, the model gains less from the unlabeled data, but still performs better than the baseline. Nevertheless, we can see that our proposed method is universally effective when $\gamma \in [0.5, 4]$.

The effect of the smoothing coefficient. We varied the smoothing coefficient α to evaluate the effect of the mean teacher. We introduced mean teachers to smooth the pseudo labels and avoid two ROI heads collapsing to each other. As shown in Table 6, we can see that the performance reaches its best value of 30.53 when α is 0.99. When α is too small, the parameters of the teacher model change too rapidly. Thus, the estimated detection results depend more on the student models of the several most recent iterations, and the performance is similar to that "without MT", as shown in Table 3. When the teacher model changes rapidly, the teacher model becomes similar to the student model. Thus, the two ROI heads might collapse to each other. However, when the smoothing coefficient is too large (e.g., $\alpha = 0.9999$), the weight of the teacher model depends largely on the previous teacher model. Thus, the pseudo labels change in an overly smooth fashion, and the student also learns slowly .

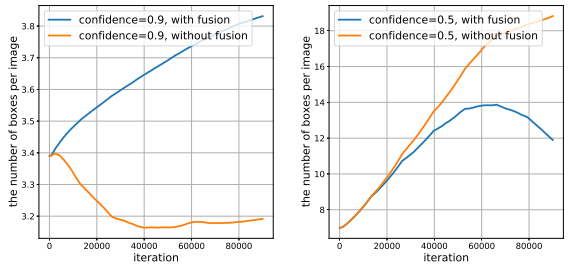
The confidence threshold of pseudo labels. The confi-

η	0.5	0.6	0.7	0.8	0.9
10% COCO	23.42	25.32	31.23	30.76	30.53

Table 7. Analysis of the smoothing coefficient α on the MS-COCO dataset



(a) with pseudo label fusion



(b) confidence threshold $\eta = 0.9$ (c) confidence threshold $\eta = 0.5$

Figure 5. The number of boxes per image of unlabeled data under different confidence thresholds η

dence threshold η determines the number of boxes for unlabeled data. A lower confidence threshold means that more boxes are considered, which allows the model to mine more information and detect more objects. However, a lower threshold also introduces additional noisy pseudo labels, which are detrimental to the detection performance. Therefore, we varied the confidence threshold η from 0.5 to 0.9 to evaluate its effect. As shown in Figure 5 (a), the number of boxes increases as the threshold η decreases. The corresponding performances are reported in Table 7. When $\eta = 0.5$, the number of boxes per image is highest and increases rapidly, but the model does not perform significantly better than does the baseline model. The pseudo labels are substantially noisier when the threshold is very low. On the other hand, although the pseudo label quality is better when $\eta = 0.9$, the number of pseudo labels also becomes smaller, meaning that more the model ignores more objects during the semi-supervised learning process. The model achieves its best performances when η is 0.7. As shown in Figure 5, the number of boxes per image is unstable when we train the model without pseudo label fusion. When $\eta = 0.9$, the number of boxes decreases gradually, which means that the model is inclined to miss more targets during training. However, when η is 0.5 and pseudo label

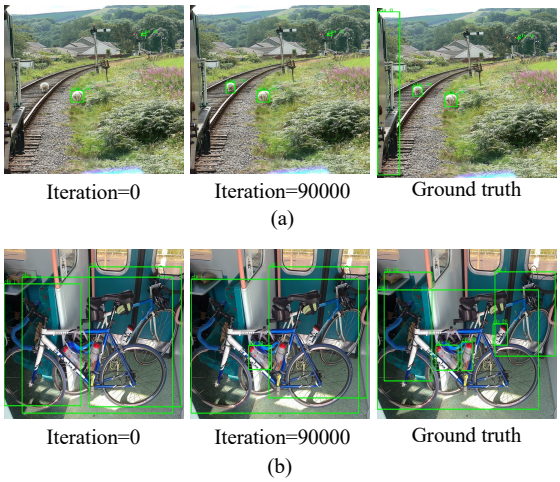


Figure 6. Visualization of the change in pseudo labels for the same image during training. The confidence threshold is 0.9.

fusion is removed, the number of boxes increases rapidly. When we train the model with pseudo label fusion, the number of boxes will increase smoothly, and their confidence scores will be higher, signifying an improved pseudo label quality.

Pseudo label visualizations. We visualized some of the detection results of the unlabeled data during the semi-supervised learning process to observe the pseudo label improvements. Figure 6 shows that the unlabeled images initially contain fewer boxes. As training progresses; however, the number of boxes in the unlabeled image increases. Furthermore, low-quality boxes are suppressed, and new targets are discovered.

5. Conclusion

In this paper, we proposed an interactive self-training framework with mean teachers. The proposed framework avoids overfitting and improves the quality of pseudo labels for semi-supervised object detection. To improve pseudo label quality, we proposed a pseudo label fusion method based on nonmaximum suppression that fuses the up-to-date detection result with historical pseudo label results. Thus, the high-quality pseudo labels are preserved, while the low-quality pseudo labels are filtered out. To overcome overfitting and leverage the discrepancies among pseudo label predictions, we introduced the mean teacher concept to estimate the detection result instead of the student ROI head. Then, we used the pseudo labels from one of the mean teacher heads as the target for unlabeled data to compute the loss of the other student head. The experimental results on MS-COCO and PASCAL-VOC validate the effectiveness of our method.

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in neural information processing systems*, pages 3365–3373, 2014. 1

[2] 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. 1

[3] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. 4

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[5] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Semi-supervised deep learning with memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–283, 2018. 4

[6] 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. 2, 5

[7] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pages 10727–10737, 2018. 3, 4, 5, 7

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 2

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1, 2

[10] 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. 1, 2

[11] 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. 6

[12] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 5

[13] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Advances in neural information processing systems*, pages 10759–10768, 2019. 1, 2, 6

[14] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1377–1385, 2017. 1

[15] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 1, 2, 5

[16] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 2

[17] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, pages 10276–10286, 2019. 2, 4

[18] 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. 1, 2, 3, 6

[19] 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

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 2

[21] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 2

[22] Nhu-Van Nguyen, Christophe Rigaud, and Jean-Christophe Burie. Semi-supervised object detection with unlabeled data. In *VISIGRAPP (5: VISAPP)*, pages 289–296, 2019. 1

[23] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, pages 135–152, 2018. 4

[24] Ilija Radosavovic, Piotr Dollar, 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. 2, 6

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 2

[26] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1, 2

[27] 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, 2, 3, 6

[28] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Alexander G Schwing, and Jan Kautz. Ufo2: A unified framework towards omni-supervised object detection. In *European Conference on Computer Vision*, pages 288–313. Springer, 2020. 2, 6

[29] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. 2005. 1, 2

[30] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, pages 1163–1171, 2016. 1

[31] Miaojing Shi, Holger Caesar, and Vittorio Ferrari. Weakly supervised object localization using things and stuff transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3381–3390, 2017. 1

[32] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 1, 2, 4, 5, 6

[33] Peng Tang, Chetan Ramaiah, Ran Xu, and Caiming Xiong. Proposal learning for semi-supervised object detection. *arXiv preprint arXiv:2001.05086*, 2020. 5

[34] Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandréa, Robert Gaizauskas, and Liming Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2119–2128, 2016. 1

[35] 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. 1, 2, 5

[36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 1, 2

[37] Jiajie Wang, Jiangchao Yao, Ya Zhang, and Rui Zhang. Collaborative learning for weakly supervised object detection. *arXiv preprint arXiv:1802.03531*, 2018. 1

[38] Keze Wang, Xiaopeng Yan, Dongyu Zhang, Lei Zhang, and Liang Lin. Towards human-machine cooperation: Self-supervised sample mining for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1605–1613, 2018. 1

[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

[40] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 1

[41] 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. 2, 4

[42] Ziang Yan, Jian Liang, Weishen Pan, Jin Li, and Changshui Zhang. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv preprint arXiv:1702.08740*, 2017. 1

[43] Qize Yang, Hong-Xing Yu, Ancong Wu, and Wei-Shi Zheng. Patch-based discriminative feature learning for unsupervised person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3633–3642, 2019. 4

[44] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1476–1485, 2019. 1

[45] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 4

[46] Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *IJCAI*, volume 5, pages 908–913, 2005. 4

[47] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1850, 2017. 1