

# Revisiting Class Imbalance for End-to-end Semi-Supervised Object Detection

Purbayan Kar, Vishal Chudasama, Naoyuki Onoe, Pankaj Wasnik\*  
Media Analysis Group, Sony Research India, Bangalore, India

{purbayan.kar, vishal.chudasama1, naoyuki.onoe, pankaj.wasnik}@sony.com

## Abstract

*Semi-supervised object detection (SSOD) has made significant progress with the development of pseudo-label-based end-to-end methods. However, many of these methods face challenges due to class imbalance, which hinders the effectiveness of the pseudo-label generator. Furthermore, in the literature, it has been observed that low-quality pseudo-labels severely limit the performance of SSOD. In this paper, we examine the root causes of low-quality pseudo-labels and present novel learning mechanisms to improve the label generation quality. To cope with high false-negative and low precision rates, we introduce an adaptive thresholding mechanism that helps the proposed network to filter out optimal bounding boxes. We further introduce a Jitter-Bagging module to provide accurate information on localization to help refine the bounding boxes. Additionally, two new losses are introduced using the background and foreground scores predicted by the teacher and student networks to improvise the pseudo-label recall rate. Furthermore, our method applies strict supervision to the teacher network by feeding strong & weak augmented data to generate robust pseudo-labels so that it can detect small and complex objects. Finally, the extensive experiments show that the proposed network outperforms state-of-the-art methods on MS-COCO and Pascal VOC datasets and allows the baseline network to achieve 100% supervised performance with much less (i.e., 20%) labeled data.*

## 1. Introduction

The semi-supervised learning (SSL) theory provides a useful illustration of how the vast amount of unlabeled data can be exploited using a small labeled data set [28]. In this work, we revisit the problem of SSL-based object detection (SSOD), in which an object detector is trained with a large amount of unlabeled data and a small amount of labeled bounding boxes. To achieve this, existing SSOD methods typically use two strategies: consistency-based SSOD [6,24] and pseudo-tagging-based SSOD [1,8,17,29,32,35].

\*Pankaj Wasnik is the corresponding author.

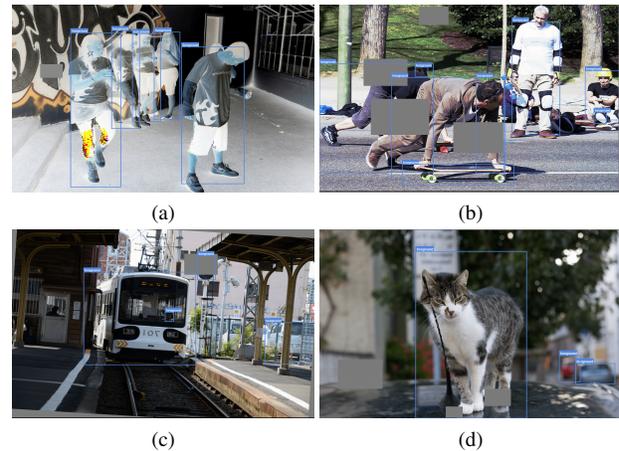


Figure 1. Foreground detection on challenging scenarios such as bad exposure, high perturbations, small, and blur.

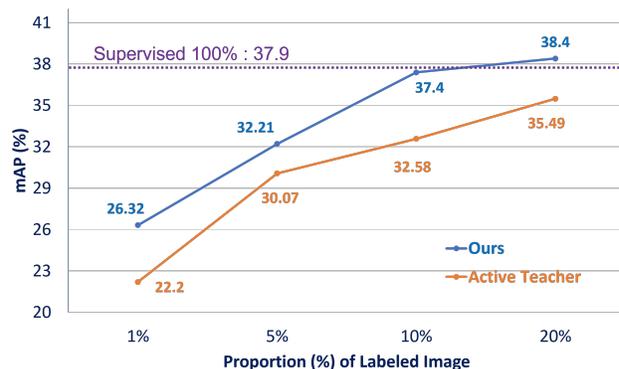


Figure 2. Comparison between proposed network and the existing SOTA method called Active Teacher [17] on MS-COCO dataset.

Consistency-based approaches train their detector by minimizing the inconsistency between the predictions of unlabeled data with different perturbations. Their performance is highly dependent on the design of the perturbations and the consistency measurement. Recently, pseudo-labeling-based frameworks [1,8,17,29,32,35] have become popular. These methods follow teacher-student scheme in which the teacher network generates pseudo-labels using unlabeled

data. Concurrently, the student network is trained at each iteration with the predicted pseudo-labels and few labeled data. The benefit of such learning is that as the network converges during training, the quality of the pseudo-labels increases. However, a high-quality pseudo-label requires both precise classification and localization [29]. In this paper, we examine the root cause of the negative impact of low-quality pseudo labels.

Low-quality pseudo-labeling can aggravate class imbalance issues, resulting in a high false-negative rate (i.e., failing to identify objects from the less prevalent classes) and a low precision rate (i.e., incorrectly identifying objects from the less common classes). To address this, the existing pseudo-label-based SSOD frameworks [25, 32, 38] use a hand-crafted threshold to distinguish pseudo-bounding boxes for student training. However, the hard threshold as a hyperparameter must be carefully tuned and dynamically adjusted according to the model capability in different time steps [31]. To carefully discriminate pseudo-bounding boxes, we introduce an adaptive threshold filter mechanism that adjusts the threshold based on background/foreground bounding boxes at each time step. From experimental analysis, we also proved the effectiveness of adaptive threshold mechanism over static hand-crafted threshold.

Applying pseudo-labels directly to object detection raises the problem of imprecise bounding box localization [29]. Xu *et al.* [32] also analyzed this effect and found that bounding boxes with high foreground scores may not provide accurate localization information and therefore not suitable for box regression tasks. To address this issue, we introduce a Jitter-Bagging module to estimate the reliable bounding boxes by measuring the consistency of its regression prediction. The effectiveness of the proposed Jitter-Bagging is also validated in the experimental section.

Another problem is the low recall rate of pseudo-labels, which impairs model training and causes many candidate boxes to be mistaken for a background category due to poor matching of pseudo-labels [29]. To address this issue, we're introducing two new losses that will help to improve foreground classification accuracy. The first loss is the background similarity loss, which helps the network to match the teacher-generated pseudo-boxes and the predicted boxes. Minimizing this loss ensures that the pseudo-boxes generated by the teacher and predicted by the students become as similar as possible. Another loss function is the foreground-background dissimilarity loss to separate the foreground bounding box from the background boxes. These losses help the proposed network to improve the pseudo-label recall rate, so that it can improve detection performance.

In addition, we analyzed impact of the Exponential Moving Average (EMA) update mechanism and found that it faces a lag issue that limits its performance in sudden weight fluctuations. To address this, we employed Double

EMA (DEMA) [19], which gives more weight to the latest observations and removes the lag compared to EMA. As per our knowledge, the DEMA is being used in the SSOD weight update mechanism for the first time, and its effectiveness in detail is discussed in the Section 4.5. Our scheme also applies strict supervision to the teacher network and feeds strong and weak augmented data into the teacher network to generate accurate pseudo-labels. Interestingly, the proposed network can accurately detect the foreground bounding boxes even for small and highly complex objects. This can be verified using Figure 1, where some results generated by the proposed network on challenging scenarios are visualized.

We perform extensive experiments with benchmark datasets, namely MS-COCO [13] and Pascal VOC [4] to validate the proposed method. The experimental analysis not only confirms the significant performance gain over SOTA methods, but also shows that the proposed method allows the baseline network to achieve 100% supervised performance with much less (i.e., 20%) annotated images in the MS-COCO dataset as shown in Figure 2.

Finally, the key contributions of the paper can be summarized as follows:

- The existing class imbalance can significantly hinder the efficacy of pseudo-label generators. This issue can be alleviated by incorporating suitable learning mechanisms in end-to-end manner.
- The high false negative and low precision rates can be improvised by carefully distinguishing the pseudo-bounding boxes. To handle this, we propose an adaptive thresholding mechanism that adjusts the threshold based on background/foreground bounding boxes and helps to filter out optimal bounding boxes.
- To provide accurate localization information, we introduce a Jitter-Bagging module for the regression task that helps the proposed network to refine optimal bounding boxes.
- To improve pseudo-label recall rate and detect blurry and distorted small objects as foreground objects, we introduce two new losses: background similarity loss and foreground-background dissimilarity loss.

## 2. Related works

Existing SSOD frameworks can be categorized as the pseudo-label-based methods [1, 8–12, 14, 17, 20, 23, 29, 30, 32, 34–36, 39] and consistency-based methods [6, 24]. The pseudo-label-based works proved better performance against consistency-based SSOD approaches.

In [23], Shon *et al.* introduce a pseudo-labeling-based method but it lacks consideration of serious data imbalance issues. Zhang *et al.* [34] proposed an adaptive self-training

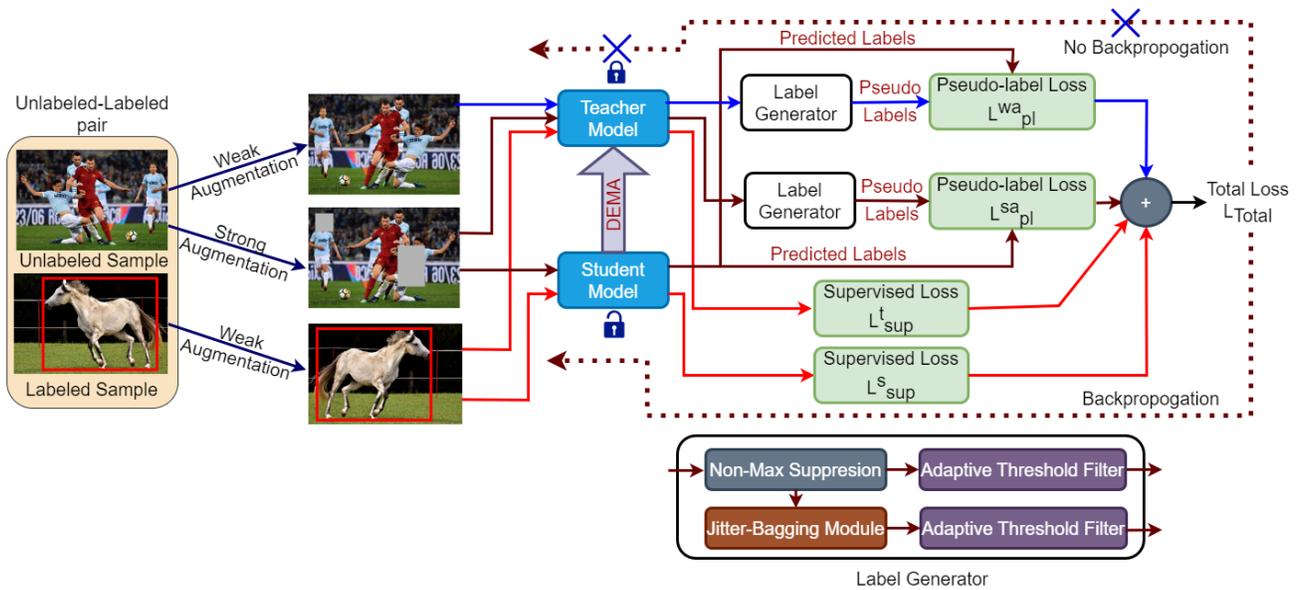


Figure 3. Network architecture of the proposed teacher-student-based network.

model for class-rebalancing but it requires additional memory module. Yang *et al.* [33] proposed interactive form of self-training to tackle the discrepancies in results. However, their model requires two ROI heads to mine complementary information. In contrast, Xu *et al.* [32] present an end-to-end based soft teacher mechanism to generate better pseudo-labels. Similarly, Tang *et al.* [25] follow the teacher-student dual model framework to generate more consistent pseudo-labels. These methods [25, 32] perform better than multi-stage based approaches with less complexity, but they still suffers from class imbalance problem.

Zheng *et al.* [36] observed the effects of single threshold and introduced a two-stage threshold mechanism based dual decoupling framework. Recently, Liu *et al.* [14] proposed a cycle self-training network to overcome the coupling effect of teacher-student learning. In [9], Li *et al.* proposed a method that uses the dense guidance teacher directly to monitor student training. Recently, Mi *et al.* [17] examined teacher-student learning from the perspective of data initialization where the label set is partially initialized and gradually augmented by evaluating key factors of unlabeled examples. In [29], Wang *et al.* identified the inconsistency in object proposals and proposed a framework to overcome the harm caused by insufficient quality of pseudo-labels.

Recent, authors in [16] showed the generalization of the SSOD method to anchor-free detectors and also introduced Listen2Student mechanism to prevent misleading pseudo-labels. Chen *et al.* [2] also studied anchor-free detectors and proposed a dense learning-based framework to generate stable and precise pseudo-labels. In [37], Zhou *et al.* proposed replacing sparse pseudo-boxes with dense predic-

tions to obtain rich pseudo-label information. Li *et al.* [10] introduced noisy pseudo box learning and multi-view scale-invariant learning to provide better pseudo-labels. To take full advantage of labeled data, Li *et al.* [8] proposed a multi-instance alignment model that improves prediction consistency based on global class prototypes. Che *et al.* [1] proposed a framework from two perspectives, i.e., distribution-level and instance-level to handle the class imbalance issue. Recently, Zhang *et al.* [35] introduced a dual pseudo-label polishing framework to reduce the deviation of pseudo-labels from ground truth through dual-polishing learning.

To address the crucial class imbalance issue and produce better pseudo-labels, we also employ pseudo-label-based teacher-student scheme and introduce two crucial modules, two new classification losses and a new learning mechanism.

### 3. Methodology

#### 3.1. Problem Statement

This paper aims to perform robust pseudo-label-based end-to-end SSOD where a set of labeled images  $D_l = \{x_{l,i}; y_{l,i}\}_{i=1}^{N_l}$  and a set of unlabeled images  $D_u = \{x_{u,j}\}_{j=1}^{N_u}$  are used for training. Here,  $N_l$  and  $N_u$  are the number of labeled and unlabeled images. Further,  $x_l$  and  $y_l$  denote the image and its ground-truth annotations i.e., class labels and bounding box coordinates, respectively.

#### 3.2. Overview of the proposed network

The architectural pipeline of proposed end-to-end network is illustrated in Figure 3. In each training iteration,

labeled and unlabeled images are arbitrarily sampled to form an input batch. The teacher network produces the pseudo labels based on the weak and strong augmented unlabeled images. While the student network is trained using weakly augmented labeled images with ground-truth and strongly augmented unlabeled images with pseudo-labels as ground-truth. Here, the student network is trained using the weighted combination of the supervised and pseudo-label loss. This can be expressed mathematically as

$$L_{Total} = L_{sup} + \lambda \cdot (L_{pl}^{wa} + L_{pl}^{sa}). \quad (1)$$

Here,  $\lambda$  controls the contribution of pseudo label loss.  $L_{sup}$  is the supervised loss consisting  $L_{sup}^t$  and  $L_{sup}^s$  loss associated with the teacher and student network, respectively. While  $L_{pl}^{wa}$  and  $L_{pl}^{sa}$  are the pseudo-label losses based on the weak and strong augmented samples, respectively. These losses are mathematically described as

$$L_{sup} = \frac{1}{N_l} \sum_{i=1}^{N_l} \left( L_{sup}^{cls}(I_{l,i}) + L_{sup}^{reg}(I_{l,i}) \right), \quad (2)$$

$$L_{pl}^{wa} = \frac{1}{N_u} \sum_{i=1}^{N_u} \left( L_{pl}^{cls}(I_{u,i}^{wa}) + L_{pl}^{reg}(I_{u,i}^{wa}) \right), \quad (3)$$

$$L_{pl}^{sa} = \frac{1}{N_u} \sum_{i=1}^{N_u} \left( L_{pl}^{cls}(I_{u,i}^{sa}) + L_{pl}^{reg}(I_{u,i}^{sa}) \right). \quad (4)$$

Here,  $I_{l,i}$  denotes the  $i^{th}$  labeled image.  $I_{u,i}^{sa}$  and  $I_{u,i}^{wa}$  indicates the  $i^{th}$  strong augmented and weak augmented unlabeled image, respectively.  $L_{sup}^{cls}$  and  $L_{sup}^{reg}$  are the supervised classification and regression loss. Similarly,  $L_{pl}^{cls}$  and  $L_{pl}^{reg}$  are pseudo-label based classification and regression losses. Here, the number of labeled images and unlabeled images are noted as  $N_l$  and  $N_u$ , respectively.

During the training process, the student network is trained using the weighted loss function given in Eq.1, and the teacher network is updated via the Double Exponential Moving Average (DEMA) update mechanism [19]. The teacher network predicts many bounding boxes for an unlabeled image. Hence, we employ the Non-Max Suppression (NMS) to eliminate redundancy. Although most of the redundant boxes are removed, some non-foreground candidates may remain. Therefore, only candidates with a foreground score<sup>1</sup> greater than an adaptive threshold are retained as pseudo boxes. These pseudo boxes are then utilized in the classification loss. To learn box regression, the bounding boxes are passed through our Jitter-Bagging module to select reliable pseudo boxes, which are subsequently refined by the adaptive threshold filter.

In the following subsections, we discuss the adaptive threshold filter, efficient classification loss, Jitter-Bagging module and update mechanism in detail.

<sup>1</sup>The foreground score is defined as the maximum probability of all non-background categories.

### 3.3. Adaptive threshold filter

The performance of the detector network depends on quality of the pseudo-label. However, quality degrades due to the class imbalance, especially when there are few annotations. For underrepresented classes, the teacher network produces relatively lower confidence score [3], which barely survives the large threshold  $\tau$ . On the other hand, simply lowering  $\tau$  leads to noisier pseudo-labels in common classes. Therefore, we propose adaptive threshold filter that adjusts the threshold value based on the confidence scores of the background and foreground bounding boxes for each category. The adaptive threshold (i.e.,  $\tau_a$ ) is mathematically defined for  $N_b^{fg}$  number of foreground and  $N_b^{bg}$  number of background bounding boxes as

$$\tau_a = \left\lceil \left( \frac{\frac{1}{N_b^{fg}} \sum_{i=1}^{N_b^{fg}} \mathcal{S}_i^{fg}}{\frac{1}{N_b^{bg}} \sum_{j=1}^{N_b^{bg}} \mathcal{S}_j^{bg}} \right)^\gamma \right\rceil, \quad (5)$$

where  $\gamma$  controls the degree of the underrepresented classes and it is set to 0.05,  $\lceil \cdot \rceil$  indicates the closest decimal floor function for single precision (e.g. 0.94 will be set to 0.9). Here,  $\mathcal{S}_i^{fg}$  and  $\mathcal{S}_j^{bg}$  denote the scores obtained from the  $i^{th}$  and  $j^{th}$  foreground and background bounding boxes.

At the beginning of the training, when parameters of the networks are not fully learned, the majority is mispredictions, i.e., predominantly background predictions than foreground predictions. So adaptive threshold outputs a relatively smaller value and as the training progresses towards convergence, foreground predictions increases and thus, the adaptive threshold provides a larger value and more constraints for selecting appropriate bounding boxes.

### 3.4. Efficient classification loss

For classification task, the overall loss is obtained by combining four different losses: foreground classification loss (i.e.,  $L_{fg}^{cls}$ ), background classification loss (i.e.,  $L_{bg}^{cls}$ ), background similarity loss (i.e.,  $L_{bg}^{sim}$ ) and foreground-background dissimilarity loss (i.e.,  $L_{fg-bg}^{dissim}$ ). The overall classification loss function can be defined as

$$L_{pl}^{cls} = L_{fg}^{cls} + L_{bg}^{cls} + L_{bg}^{sim} + L_{fg-bg}^{dissim}. \quad (6)$$

**1) Foreground classification loss:** Given student-generated foreground bounding boxes (i.e.,  $b^{fg}$ ), the foreground classification loss is defined as

$$L_{fg}^{cls} = \frac{1}{N_b^{fg}} \sum_{i=1}^{N_b^{fg}} l_{cls}(b_i^{fg}(s), \mathcal{B}_{cls}), \quad (7)$$

where  $\mathcal{B}_{cls}$  denotes the set of teacher-generated pseudo boxes used for classification,  $l_{cls}$  is the box classification

loss<sup>2</sup>,  $N_b^{fg}$  is the number of box candidates of box set  $b^{fg}$ .

**2) Background classification loss:** We employ the same loss function proposed by Xu *et al.* [32] for the background classification loss. Given background bounding boxes (i.e.,  $b^{bg}$ ), the classification loss is calculated as

$$L_{bg}^{cls} = \sum_{j=1}^{N_b^{bg}} \delta_j l_{cls}(b_j^{bg}(s), \mathcal{B}_{cls}) \quad (8)$$

where,  $\delta_j$  denoted as reliability weighting factor associated with  $j^{th}$  sample and the same can be expressed as

$$\delta_j = \frac{r_j}{\sum_{k=1}^{N_b^{bg}} r_k} \quad (9)$$

Here  $r_j$  is the reliability score for  $j^{th}$  background box candidate,  $N_b^{bg}$  is the number of box candidates of box set  $b^{bg}$ .

**3) Background similarity loss:** Inspired from [18], we introduce a novel loss to match the scores obtained from the background bounding boxes generated through teacher and student networks. Minimizing this loss will ensure that the teacher-generated pseudo boxes and student-predicted boxes become as similar as possible. The background similarity loss can be expressed as follows:

$$L_{bg}^{sim} = \frac{1}{N_b^{bg}} \sum_{i=1}^{N_b^{bg}} \beta \cdot \log(|e^{|\mathcal{S}_i^{bg}(s)}| - e^{|\mathcal{S}_i^{bg}(t)}| + 1), \quad (10)$$

where,  $\beta$  denotes the controlling parameter,  $\mathcal{S}_i^{bg}(t)$  and  $\mathcal{S}_i^{bg}(s)$  indicates  $i^{th}$  scores obtained from the background bounding boxes generated using teacher and student networks, respectively.

**4) Foreground-Background dissimilarity loss:** Inspired from relativistic average discriminator [7], a novel loss is introduced to separate out the foreground and background bounding boxes. The introduced loss considers the dissimilarity between scores obtained from background and foreground bounding boxes. Mathematically, this can be addressed as follows:

$$L_{fg-bg}^{dissim} = \frac{1}{N_b^{fg}} \sum_{i=1}^{N_b^{fg}} \left(1 - \left| \mathcal{S}_i^{fg}(s) - \frac{1}{N_b^{bg}} \sum_{j=1}^{N_b^{bg}} \mathcal{S}_j^{bg}(s) \right| \right). \quad (11)$$

Here,  $\mathcal{S}_i^{fg}(s)$  and  $\mathcal{S}_i^{bg}(s)$  indicate  $i^{th}$  and  $j^{th}$  scores obtained from the student-generated foreground and background bounding boxes.

### 3.5. Jitter-Bagging module

Xu *et al.* [32] found that the selection of the teacher-generated pseudo boxes according to the foreground score

<sup>2</sup>We use standard cross-entropy loss function as classification loss.

is not suitable for box regression. To tackle this issue, we introduce a Jitter-Bagging module where we sample a jittered-box around the teacher-generated pseudo box candidate  $b_i$  and then feed to the teacher network to obtain refined box  $b'_i$ . This can be formulated as

$$b'_i = f_{Jitter}(b_i). \quad (12)$$

Here,  $f_{Jitter}$  denotes the function of the Jitter operation. This procedure is repeated several times to collect the set of  $N_{jitter}$  refined jittered boxes (i.e.,  $\{b'_i\}$ ). These refined jittered boxes are then fed to the traditional bagging algorithm which helps to obtain optimum refined boxes. Mathematically, this can be stated as:

$$\hat{b}_i = f_{Bagging}(\{b'_i\}), \quad (13)$$

where,  $f_{Bagging} = \max(\cdot)$  is the bagging operation which selects the bounding box candidate with maximum area. The obtained bounding boxes (i.e.,  $\hat{b}_i$ ) are further passed through an adaptive threshold filter to generate foreground bounding boxes (i.e.,  $\hat{b}_i^{fg}$ ). Finally, given the pseudo boxes  $\mathcal{B}_{reg}$  for training the box regression on unlabeled data, the regression loss is formulated as

$$L_{pl}^{reg} = \frac{1}{N_b^{fg}} \sum_{i=1}^{N_b^{fg}} l_{reg}(\hat{b}_i^{fg}, \mathcal{B}_{reg}), \quad (14)$$

where,  $N_b^{fg}$  is the total number of foreground box,  $l_{reg}$  is the box regression loss<sup>3</sup>.

### 3.6. Update mechanism

At each iteration, the teacher weights get marginal updates using the student's weights via the update mechanism. The gradually updated teacher is prone to weight fluctuations of the student network when a teacher network mispredicts a label. In [25, 32], authors used the Exponential Moving Average (EMA) mechanism to mitigate the negative effect of incorrect pseudo-labels [26]. However, the EMA update mechanism faces an issue of lagging, which limits its performance during sudden weight fluctuations. In this work, we employ EMA's extension, i.e., Double Exponential Moving Average (DEMA) [19, 27]. The DEMA provides higher weight to the most recent observations and removes the inherent lag as compared to the EMA update mechanism. Further, the DEMA update mechanism can be mathematically defined as:

$$w(t)_{ts} = 2 \cdot EMA(t)_{ts} - EMA(EMA(t))_{ts}, \quad (15)$$

where, the  $EMA$  can be expressed as

$$EMA(t)_{ts} = \alpha^2 w(t)_{ts-1} + (1 - \alpha^2) w(s)_{ts}. \quad (16)$$

Here,  $w(t)_{ts}$  and  $w(s)_{ts}$  are the weights of teacher and student network at current timestamp  $ts$  and  $\alpha$  is set to 0.999.

<sup>3</sup>We use standard mean absolute error for regression task.

## 4. Experimental Setup and Discussion

### 4.1. Details of Dataset and Evaluation

We present our results on benchmark MS-COCO [13] and Pascal VOC [4] datasets.

**MS-COCO dataset:** It comprises of more than 118k labeled images (train2017 set), 123k unlabeled images (unlabeled2017 set) and 5k labeled validation dataset (val2017 set). For validation purpose, we follow the principles suggested in [6, 23, 24] and the same is discussed below:

- **Partially Labeled Data:** Here, 1%, 5%, and 10% of the train2017 set are sampled as labeled data, while remaining unsampled images are operated as unlabeled data. The network is performed on five different folds and evaluated by taking an average of all five folds.
- **Fully Labeled Data:** This setting is more challenging. It aims to enhance a trained detector on large-scale labeled data by using the extra unlabeled data. The training process uses the entire train2017 set as the labeled data and unlabeled2017 set as unlabeled dataset.

**Pascal VOC dataset:** It takes VOC07 *trainval* set as labeled data having 5,011 images, and 11,540 images from the *trainval* set of VOC12 as an unlabeled data. The performance is evaluated on the test set of VOC07 in three experimental setups: (a) fully supervised on VOC07 labeled set; (b) VOC07 labeled set and VOC12 additional unlabeled set; and (c) VOC07 labeled set and VOC12 & COCO20cls<sup>4</sup> as additional unlabeled sets.

Furthermore, we follow the data augmentation guidelines given in STAC [23] and FixMatch [22] for training and pseudo-label generation. For evaluation, we use the mean average precision (mAP)<sup>5</sup> metric with its different variants i.e., mAP at IoU=0.5 (mAP@50), IoU=0.75 (mAP@75) and IoU=0.5:0.95 (mAP).

### 4.2. Training Setups and Hyper-parameter Tuning

All experiments are performed with NVIDIA A100 dual GPUs. To allow a fair comparison with previous methods, we use Faster R-CNN [21] as our default detection network with pre-trained ResNet-50 [5] as a backbone. For training and inference, 2k and 1k region proposals have been generated using a non-maximum suppression threshold of 0.7. We sample 512 out of 2k proposals as the box candidates in each training step.

The proposed network is trained for 180k with 0.2 data sampling ratio and 720k iterations with 0.5 data sampling ratio for partially labeled data setting and fully labeled data setting, respectively. We adopt SGD as optimizer with learning rate of 0.001 divided by 10 at 120k and 160k iterations for partially labeled data setting and at 480k and 680k

<sup>4</sup>COCO20cls is generated by leaving only COCO images whose object categories overlap with the object categories used in PASCAL VOC07.

<sup>5</sup>Average Precision (AP) is finding the area under precision-recall curve. The mAP is defined as the average of AP.

Table 1. Comparison with supervised baseline for various %s of labeled data.

	1%		5%		10%		100%	
	Supervised	Ours	Supervised	Ours	Supervised	Ours	Supervised	Ours
mAP@50	21.3	44.6	41.0	52.1	45.1	57.6	57.6	65.2
mAP@75	11.0	28.0	24.4	34.6	28.4	41.0	40.4	48.1
mAP	11.4	26.3	23.4	32.2	27.1	37.4	37.9	44.0

Table 2. Comparison between different semi-supervised methods on val2017.

Methods	Remarks	1%	5%	10%
Soft-teacher [32]	ICCV 2021	20.46 ± 0.39	30.74 ± 0.08	34.04 ± 0.14
ACRST [34]	Arxiv 2021	26.07 ± 0.46	31.35 ± 0.13	34.92 ± 0.22
DDT [36]	AAAI 2022	19.44 ± 0.32	29.92 ± 0.12	33.46 ± 0.18
CST [14]	ACM MM 2022	22.73 ± 0.14	30.83 ± 0.08	33.90 ± 0.17
Active Teacher [17]	CVPR 2022	22.20	30.07	32.58
MA-GCP [8]	CVPR 2022	21.31 ± 0.28	31.67 ± 0.16	35.02 ± 0.26
DCST [29]	IJCAI 2022	23.02 ± 0.23	32.10 ± 0.15	35.20 ± 0.20
PseCo [10]	ECCV 2022	22.43 ± 0.36	32.50 ± 0.08	36.06 ± 0.24
LabelMatch [1]	CVPR 2022	25.81 ± 0.28	<b>32.70 ± 0.18</b>	35.49 ± 0.17
Polishing Teacher [35]	AAAI 2023	23.55 ± 0.25	32.10 ± 0.15	35.30 ± 0.15
Ours	—	<b>26.32 ± 0.35</b>	32.21 ± 0.08	<b>37.40 ± 0.15</b>

iterations for fully labeled data setting. Initially, we set foreground threshold to 0.5 and then it adjusts itself adaptively between 0.5 to 0.9.

We set  $N_{jitter}$  to 10 in the proposed Jitter-Bagging module for regression task. The jittered boxes are randomly sampled by adding the offsets on four coordinates, and the offsets are uniformly sampled from [-6%, +6%] of the height or width of the pseudo box candidates.

### 4.3. Result analysis of proposed network

This section provides the result comparison between the proposed network along with its supervised network on MS-COCO dataset. The validation performance is tabulated in Table 1. Here, one can see that our proposed network shows a significant performance improvement than the supervised baseline network in all protocols. In addition, we present the visual comparison for different proportion of labeled data in Figure 4. Here, it is clearly observed that the proposed network is able to detect tiny and occluded objects with better confidence score than that of the supervised framework.

### 4.4. Comparison with state-of-the-art methods

#### 4.4.1 Comparison on MS-COCO:

**Partially labeled data setting:** This section compares our network with existing state-of-the-art (SOTA) SSOD methods under the partially labeled data setting. The corresponding average mAP measures of 5 folds are noted in Table 2. From the table, we can observe that the proposed network outperforms all other methods and obtain +0.25%, and +1.34% higher mAP values than previous best performing SOTA methods [10, 34] on 1%, and 10% labeled data, respectively. While in case of 5% labeled data setting, it performs slightly inferior than SOTA methods [1, 10].

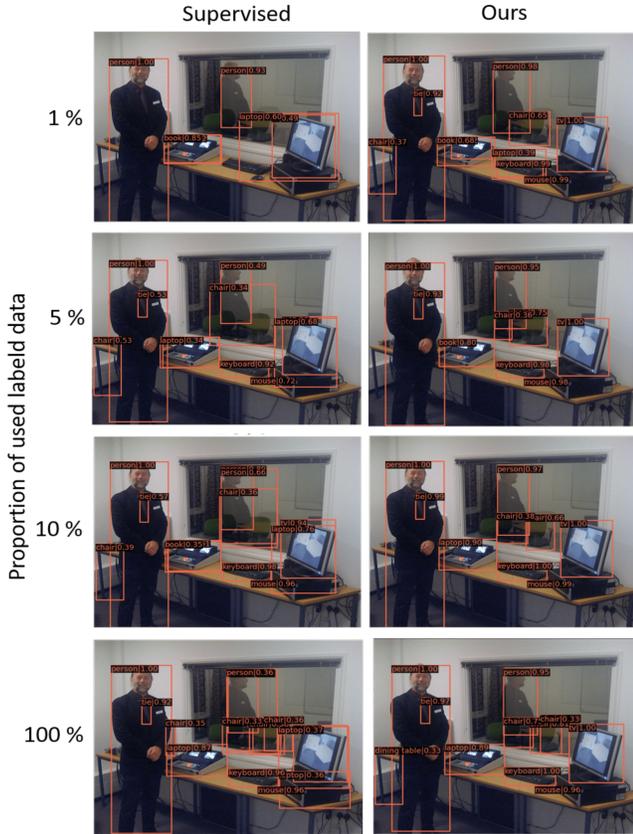


Figure 4. Visual comparison with supervised results.

Table 3. Comparison with other state-of-the-arts under the setting of fully labeled data of train2017 set.

Methods	Remarks	Extra Dataset	mAP
Self-training [39]	NIPS 2020	ImageNet+OpenImages	41.1 $\xrightarrow{+0.8}$ 41.9
Soft-teacher [32]	ICCV 2021	unlabeled2017	40.9 $\xrightarrow{+3.6}$ 44.5
MA-GCP [8]	CVPR 2022	unlabeled2017	40.9 $\xrightarrow{+5.0}$ 45.9
LabelMatch [1]	CVPR 2022	unlabeled2017	40.3 $\xrightarrow{+5.0}$ 45.3
DDT [36]	AAAI 2022	unlabeled2017	37.6 $\xrightarrow{+4.6}$ 42.2
CST [14]	ACM MM 2022	unlabeled2017	37.6 $\xrightarrow{+5.7}$ 43.3
DTG-SSOD [9]	Arxiv 2022	unlabeled2017	40.9 $\xrightarrow{+4.8}$ 45.7
PseCo [10]	ECCV 2022	unlabeled2017	41.0 $\xrightarrow{+5.1}$ 46.1
DCST [29]	IJCAI 2022	unlabeled2017	40.9 $\xrightarrow{+3.7}$ 44.6
Ours	—	unlabeled2017	37.9 $\xrightarrow{+6.1}$ 44.0

**Fully labeled data setting:** Here, we compare our network with other methods in the fully labeled data setting. Since the reported performance of the supervised baseline varies, we report the results of the comparison methods and their baseline simultaneously in Table 3. The additional unlabeled dataset required to improve the baseline performance is also mentioned here and the corresponding improvement is noted in this table. One can see that the proposed network shows a larger performance gain (i.e., +6.1%) than the existing state-of-the-art methods.

Table 4. Comparison with other state-of-the-art methods on the VOC07 test set.

Model	Remarks	mAP	mAP@50	mAP@75
VOC07 labeled data (Supervised)	—	41.91	66.0	45.1
VOC07 labeled set + VOC12 unlabeled set				
RPL [11]	Arxiv 2021	54.60	79.00	59.40
CST [14]	ACM MM 2022	51.50	78.70	—
MA-GCP [8]	CVPR 2022	—	81.72	—
DDT [36]	AAAI 2022	54.70	82.40	59.80
LabelMatch [1]	CVPR 2022	55.11	<b>85.48</b>	—
Polishing Teacher [35]	AAAI 2023	52.40	82.50	—
Ours	—	<b>56.92</b>	82.04	<b>62.84</b>
VOC07 labeled set + VOC12 & COCO20cls unlabeled set				
Unbiased teacher [15]	ICLR 2021	50.34	78.82	—
Instant Teaching [38]	CVPR 2021	50.80	79.90	55.70
RPL [11]	Arxiv 2021	56.10	79.60	61.20
CST [14]	ACM MM 2022	53.50	80.50	—
DDT [36]	AAAI 2022	55.90	<b>82.50</b>	61.10
Ours	—	<b>57.10</b>	82.21	<b>63.47</b>

#### 4.4.2 Comparison on Pascal VOC:

We also evaluate our network on the Pascal VOC benchmark dataset and the comparison is presented in Table 4. When utilizing VOC07 labeled and VOC12 unlabeled data, the proposed network obtains +15.01%, +16.04% and +17.74% higher values compared to the supervised setting on mAP, mAP@50 and mAP@75, respectively. The proposed network also outperforms the previous best performing SOTA methods [1, 36] by +1.81%, and +3.04% on mAP, and mAP@75, respectively. To analyze how increasing unlabeled data can help to improve performance, we have used the COCO20cls dataset as an additional unlabeled set. As a result, the proposed network shows an absolute improvement of +15.19%, +16.21% and +18.37% compared to the fully supervised baseline. We can also see that the proposed network outperforms the SOTA methods [11] by +1.00%, and +2.27% on mAP, and mAP@75, respectively. These results verify that our network can further improve object detection by using more unlabeled data.

#### 4.5. Ablation Analysis

All experiments are carried out on 10% partially labeled setting of MS-COCO. However, the analysis on 1% and 5% data settings are covered in the supplementary material.

**Effectiveness of adaptive threshold filter:** To prove the effectiveness of proposed adaptive threshold filter, few experiments have been carried out and the corresponding results are presented in Table 5. Here, we can see that the proposed adaptive threshold filter performs better than the static threshold values. To check its effectiveness over the threshold module proposed by Li *et al.* [11], we employed their thresholding module in our framework. The corresponding results are added in Table 5, which is marginally inferior to the proposed thresholding module. In our adaptive mechanism, we have used discrete thresholding to re-

Table 5. Analysis to validate adaptive threshold mechanism.

Proposed Network	mAP	mAP@50	mAP@75
with static 0.7 threshold	33.0	52.7	36.8
with static 0.8 threshold	36.3	55.6	39.1
with static 0.9 threshold	37.0	56.4	40.3
with dynamic thresholding [11]	37.1	56.8	40.6
with continuous form-based threshold	36.1	55.9	39.5
with proposed adaptive threshold	<b>37.4</b>	<b>57.5</b>	<b>41.0</b>

Table 6. Analysis to validate the Jitter-Bagging module.

Proposed Network	mAP	mAP@50	mAP@75
without Jitter-Bagging	36.2	56.9	39.7
with Box Jittering [32]	36.8	57.0	40.1
with Jitter-Bagging	<b>37.4</b>	<b>57.5</b>	<b>41.0</b>

Table 7. Analysis to validate introduced losses for classification.

Network	mAP	mAP@50	mAP@75
Case I: $L_{fg}^{cls} + L_{bg}^{cls}$	36.8	56.9	40.4
Case II: $L_{fg}^{cls} + L_{bg}^{cls} + L_{bg}^{sim}$	37.2	57.3	40.6
Case III: $L_{fg}^{cls} + L_{bg}^{cls} + L_{fg-bg}^{dissim}$	37.1	57.2	40.6
Proposed: $L_{fg}^{cls} + L_{bg}^{cls} + L_{bg}^{sim} + L_{fg-bg}^{dissim}$	<b>37.4</b>	<b>57.5</b>	<b>41.0</b>

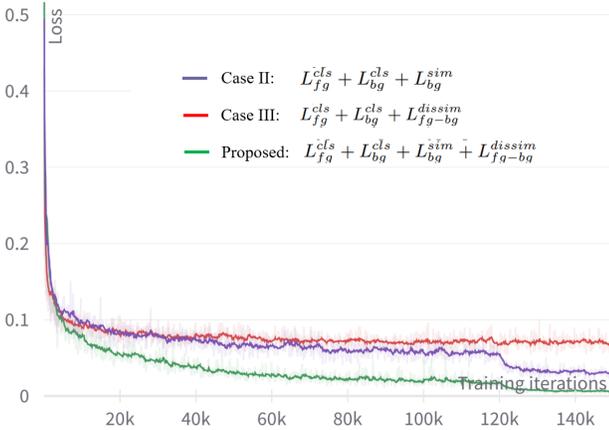


Figure 5. Analysis of the introduced classification loss during training iterations.

duce fluctuations in the threshold value. Further, we trained a variant of our model with a continuous form of threshold and found lower performance than proposed discrete form.

**Importance of Jitter-Bagging module:** The ablation analysis of Jitter-Bagging module is presented in Table 6, where we can see that the proposed Jitter-Bagging module achieves highest performance and shows +1.2% absolute improvement in mAP measure over without Jitter-Bagging module. Interestingly, when we employ the Box Jittering [32] in our network, we observe that the proposed Jitter-Bagging still obtains +0.6% higher mAP than Box Jittering.

**Effect of losses for classification:** For classification task, we introduce two novel losses; background similarity loss i.e.,  $L_{bg}^{sim}$  and foreground-background dissimilarity loss i.e.,  $L_{fg-bg}^{dissim}$ . To check its effectiveness, the proposed network is trained without both introduced losses

Table 8. Analysis to validate the DEMA update mechanism.

Network	mAP	mAP@50	mAP@75
Deep Copy	32.1	51.5	33.8
EMA update	36.2	56.4	39.9
DEMA update	<b>37.4</b>	<b>57.5</b>	<b>41.0</b>

Table 9. Analysis to check importance of label generators.

Network	mAP	mAP@50	mAP@75
Proposed (Both label generators)	<b>37.4</b>	<b>57.5</b>	<b>41.0</b>
- w/o label generator (weak augmented data)	35.2	55.4	38.9
- w/o label generator (strong augmented data)	36.6	56.8	40.4

(i.e., Case I), with background similarity loss (i.e., Case II) and with foreground-background dissimilarity loss (i.e., Case III). The corresponding measures are depicted in Table 7. Here, it can be noticed that both introduced losses help the proposed network to obtain better mAP measures. Additionally, Figure 5 shows the effect of loss values during the training iteration. Here, it can be seen that the proposed network with both losses converges better than others.

**Effect of update mechanism:** To verify the effectiveness of the DEMA update mechanism, we ablate the proposed network trained using EMA mechanism as well as employing deep copy configuration (i.e., weights of teacher network are copied from student network). The corresponding results are noted in Table 8, where it can be seen that DEMA helps to obtain +1.2% higher mAP measure, demonstrating its efficacy over EMA update mechanism.

**Importance of label generator module:** In our proposed network, we use two label generator modules; one associated with weak augmented samples while the other is based on strong augmented sample. To see the importance of this setting, the proposed network with individual label generator is trained and the obtained results are presented in Table 9. Here, it is observed that the proposed network with both label generators outperforms the individual label generator settings.

## 5. Conclusion

In this paper, we present an end-to-end teacher-student network to address the class imbalance issue in semi-supervised object detection. It successfully examines the effect of class imbalance on pseudo-label generation and proposes novel learning mechanisms to improve the pseudo-label quality. Specifically, we tackle the high false-negative and low precision rates using the proposed adaptive threshold mechanism and refine optimal bounding boxes using our Jitter-Bagging module. We further introduce two novel losses based on background and foreground bounding boxes to improve the pseudo-label recall rate so that it can detect small objects as foreground. Finally, our extensive experimentation shows that the proposed network outperforms existing state-of-the-art SSOD methods on MS-COCO and Pascal VOC benchmark datasets.

## References

- [1] Binbin Chen, Weijie Chen, Shicai Yang, Yunyi Xuan, Jie Song, Di Xie, Shiliang Pu, Mingli Song, and Yueting Zhuang. Label matching semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14381–14390, 2022. [1](#), [2](#), [3](#), [6](#), [7](#)
- [2] Binghui Chen, Pengyu Li, Xiang Chen, Biao Wang, Lei Zhang, and Xian-Sheng Hua. Dense learning based semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4815–4824, 2022. [3](#)
- [3] Achal Dave, Piotr Dollár, Deva Ramanan, Alexander Kirillov, and Ross Girshick. Evaluating large-vocabulary object detectors: The devil is in the details. *arXiv preprint arXiv:2102.01066*, 2021. [4](#)
- [4] 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](#), [6](#)
- [5] 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](#)
- [6] Jisoo Jeong, Seungeui Lee, Jeessoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. *Advances in neural information processing systems*, 32, 2019. [1](#), [2](#), [6](#)
- [7] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. [5](#)
- [8] Aoxue Li, Peng Yuan, and Zhenguo Li. Semi-supervised object detection via multi-instance alignment with global class prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9809–9818, 2022. [1](#), [2](#), [3](#), [6](#), [7](#)
- [9] Gang Li, Xiang Li, Yujie Wang, Yichao Wu, Ding Liang, and Shanshan Zhang. Dtg-ssod: Dense teacher guidance for semi-supervised object detection. *arXiv preprint arXiv:2207.05536*, 2022. [2](#), [3](#), [7](#)
- [10] Gang Li, Xiang Li, Yujie Wang, Yichao Wu, Ding Liang, and Shanshan Zhang. Pseco: Pseudo labeling and consistency training for semi-supervised object detection. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 457–472. Springer, 2022. [2](#), [3](#), [6](#), [7](#)
- [11] Hengduo Li, Zuxuan Wu, Abhinav Shrivastava, and Larry S Davis. Rethinking pseudo labels for semi-supervised object detection. *arXiv preprint arXiv:2106.00168*, 2021. [2](#), [7](#), [8](#)
- [12] Yandong Li, Di Huang, Danfeng Qin, Liqiang Wang, and Boqing Gong. Improving object detection with selective self-supervised self-training. In *European Conference on Computer Vision*, pages 589–607. Springer, 2020. [2](#)
- [13] 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](#), [6](#)
- [14] Hao Liu, Bin Chen, Bo Wang, Chunpeng Wu, Feng Dai, and Peng Wu. Cycle self-training for semi-supervised object detection with distribution consistency reweighting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6569–6578, 2022. [2](#), [3](#), [6](#), [7](#)
- [15] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. *arXiv preprint arXiv:2102.09480*, 2021. [7](#)
- [16] Yen-Cheng Liu, Chih-Yao Ma, and Zsolt Kira. Unbiased teacher v2: Semi-supervised object detection for anchor-free and anchor-based detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9819–9828, 2022. [3](#)
- [17] Peng Mi, Jianghang Lin, Yiyi Zhou, Yunhang Shen, Gen Luo, Xiaoshuai Sun, Liujuan Cao, Rongrong Fu, Qiang Xu, and Rongrong Ji. Active teacher for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14482–14491, 2022. [1](#), [2](#), [3](#), [6](#)
- [18] Jing Mu, Xinfeng Zhang, Shuyuan Zhu, and Ruiqin Xiong. Riemannian loss for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. [5](#)
- [19] Patrick Mulloy. Smoothing data with faster moving averages. *Technical Analysis of Stocks and Commodities Magazine*, Volume 12, Issue 1, January, 1994 (11-19). [2](#), [4](#), [5](#)
- [20] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omnibus supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4119–4128, 2018. [2](#)
- [21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. [6](#)
- [22] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020. [6](#)
- [23] 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. [2](#), [6](#)
- [24] Peng Tang, Chetan Ramaiah, Yan Wang, Ran Xu, and Caiming Xiong. Proposal learning for semi-supervised object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2291–2301, 2021. [1](#), [2](#), [6](#)
- [25] Yihe Tang, Weifeng Chen, Yijun Luo, and Yuting Zhang. Humble teachers teach better students for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3132–3141, 2021. [2](#), [3](#), [5](#)

- [26] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, 2017. 5
- [27] CFI Team. Double exponential moving average (dema), <https://corporatefinanceinstitute.com/resources/equities/double-exponential-moving-average-dema/>, last access: 2022-10-22. 5
- [28] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 1
- [29] Kuo Wang, Yuxiang Nie, Chaowei Fang, Chengzhi Han, Xuewen Wu, Xiaohui Wang, Liang Lin, Fan Zhou, and Guanbin Li. Double-check soft teacher for semi-supervised object detection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022. 1, 2, 3, 6, 7
- [30] 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. 2
- [31] Xinjiang Wang, Xingyi Yang, Shilong Zhang, Yijiang Li, Litong Feng, Shijie Fang, Chengqi Lyu, Kai Chen, and Wayne Zhang. Consistent targets provide better supervision in semi-supervised object detection. *arXiv preprint arXiv:2209.01589*, 2022. 2
- [32] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021. 1, 2, 3, 5, 6, 7, 8
- [33] Yang et al. Interactive self-training with mean teachers for semi-supervised object detection. In *CVPR*, 2021. 3
- [34] Fangyuan Zhang, Tianxiang Pan, and Bin Wang. Semi-supervised object detection with adaptive class-rebalancing self-training. *arXiv preprint arXiv:2107.05031*, 2021. 2, 6
- [35] Lei Zhang, Yuxuan Sun, and Wei Wei. Mind the gap: Polishing pseudo labels for accurate semi-supervised object detection. *arXiv preprint arXiv:2207.08185*, 2022. 1, 2, 3, 6, 7
- [36] Shida Zheng, Chenshu Chen, Xiaowei Cai, Tingqun Ye, and Wenming Tan. Dual decoupling training for semi-supervised object detection with noise-bypass head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3526–3534, 2022. 2, 3, 6, 7
- [37] Hongyu Zhou, Zheng Ge, Songtao Liu, Weixin Mao, Zeming Li, Haiyan Yu, and Jian Sun. Dense teacher: Dense pseudo-labels for semi-supervised object detection. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 35–50. Springer, 2022. 3
- [38] Qiang Zhou, Chaohui Yu, Zhibin Wang, Qi Qian, and Hao Li. Instant-teaching: An end-to-end semi-supervised object detection framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4081–4090, 2021. 2, 7
- [39] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020. 2, 7