

CrossRectify: Leveraging disagreement for semi-supervised object detection

Chengcheng Ma^{a,b}, Xingjia Pan^c, Qixiang Ye^d, Fan Tang^{e,*}, Weiming Dong^a, Changsheng Xu^a

^a National Lab of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, 100190, China

^b School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing, 100049, China

^c Youtu Lab, Tencent Inc., Shanghai, 200233, China

^d School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences (UCAS), Beijing, 101408, China

^e Jilin University, Changchun, 130000, China

ARTICLE INFO

Article history:

Received 15 April 2022

Revised 15 September 2022

Accepted 21 December 2022

Available online 24 December 2022

Keywords:

Object detection

Semi-supervised learning

2D Semi-supervised object detection

3D Semi-supervised object detection

Self-labeling

ABSTRACT

Semi-supervised object detection has recently achieved substantial progress. As a mainstream solution, the self-labeling-based methods train the detector on both labeled data and unlabeled data with pseudo labels predicted by the detector itself, but their performances are always limited. Through experimental analysis, we reveal the underlying reason is that the detector is misguided by the incorrect pseudo labels predicted by itself (dubbed self-errors). These self-errors can hurt performance even worse than random-errors, and can be neither discerned nor rectified during the self-labeling process. In this paper, we propose an effective detection framework named CrossRectify, to obtain accurate pseudo labels by simultaneously training two detectors with different initial parameters. Specifically, the proposed approach leverages the disagreements between detectors to discern the self-errors and refines the pseudo label quality by the proposed cross-rectifying mechanism. Extensive experiments show that CrossRectify achieves outperforming performances over various detector structures on 2D and 3D detection benchmarks.

© 2022 Elsevier Ltd. All rights reserved.

1. Introduction

The success of deep learning has greatly prompted the development of object detection approaches, such as SSD [1], FasterRCNN [2], STDnet-ST [3], MSKR [4], SCA-Net [5], CADN [6], etc, and a large amount of labeled data is essential to the training process of object detectors. However, as illustrated in [7], it is always labor-intensive and expensive to acquire a large amount of labeled data with bounding-box-level annotations. In comparison, unlabeled data are much easier and cheaper to collect. Therefore, semi-supervised object detection [8] is recently investigated to reduce the cost of data annotating, which leverages only a few labeled data and a large amount of unlabeled data to train object detectors.

Semi-supervised object detection (SSOD) has achieved significant progress in recent years, and one mainstream of existing SSOD solutions is based on the self-labeling scheme [9]. The core idea of self-labeling scheme is to first utilize the current detec-

tor to predict pseudo bounding boxes for unlabeled data in each training iteration, then conduct detector training with both labeled data and pseudo-labeled data. However, compared with fully-supervised baselines, the performance increments brought by the self-labeling-based methods are always observed limited. For example, the absolute AP₅₀ gain is only 0.40% (0.46%) without (with) the mix-up data augmentation [10] over the SSD300 [1] detector on the Pascal VOC [11] benchmark (as shown in Fig. 1). To reveal the reason behind such phenomenon, we introduce the ground-truth annotations of unlabeled data for in-depth analysis. Existing self-labeling-based SSOD methods generate pseudo labels by selecting bounding boxes with high confidence scores to ensure the quality of pseudo labels. However, we observe that part of high-confidence boxes are still misclassified (dubbed self-errors), and these self-errors can hurt the detection performance even worse. Specifically, when we replace the misclassified category labels with random labels (dubbed random-errors) during the self-labeling training process, the final performance is even improved. Unfortunately, these self-errors can be neither *discerned* nor *rectified* by the detector itself, which we summarize as two inherent limitations of the self-labeling training scheme. These two limita-

* Corresponding author.

E-mail address: tangfan@jlu.edu.cn (F. Tang).

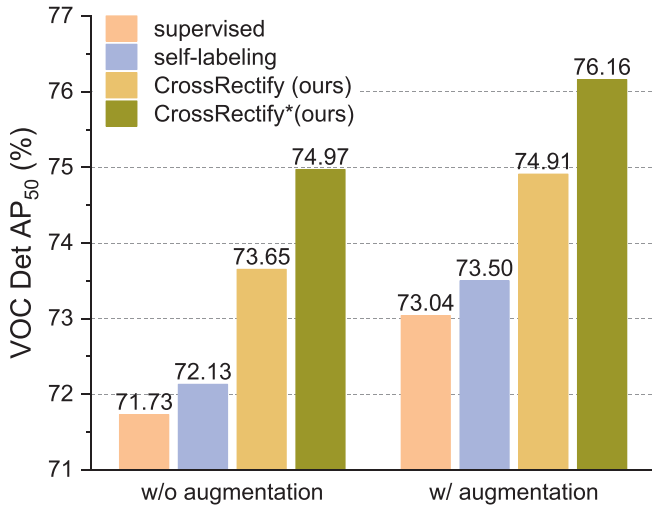


Fig. 1. The proposed CrossRectify and CrossRectify* method can outperform the self-labeling-based semi-supervised object detection method by large margins on the Pascal VOC benchmark dataset without/with the mix-up data augmentation. (For interpretation of the references to color in this figure legend, please refer to the online version of this article.)

tions will lead the detector to be misguided by self-errors, and finally results in insignificant performance increments.

Since one single detector can neither discern nor rectify the misclassified pseudo bounding boxes, it is necessary to introduce the guidance information from other distinct detector to the training process. Recently, a few works, such as Deep Co-Training [12], Dual Student [13], JoCoR [14], Co-teaching+ [15], illustrate the fact that two differently initialized models with an identical structure can yield diverse results on the same training sample during the training process. Inspired by this fact, in this paper, we propose an effective and general training framework named CrossRectify for both 2D and 3D semi-supervised object detection task. In CrossRectify, two detectors with the same structure but different initialization are trained simultaneously, and each detector is supervised by the pseudo labels generated from the proposed cross-rectifying mechanism. Specifically, the proposed cross-rectifying mechanism first leverages the disagreements on the same objects between two detectors to discern the latent self-errors predicted by each single detector. Then, the pseudo labels are generated based on the bounding boxes predicted by two detectors, by adopting a simple yet effective comparison-to-assigning pipeline with confidence scores being considered. To this end, the proposed CrossRectify method can discern and rectify the self-errors and improve the pseudo label quality.

Note that a few recent works utilize another separate detector for pseudo label generation, which can somehow alleviate the limitations of self-labeling training scheme. For instance, recent works including STAC [16], Uncertainty [17], UBTeacher [18], Humble Teacher [19], Soft Teacher [20] adopt the teacher-student mutual learning framework [21] and utilize a specific teacher detector to generate pseudo labels in an offline or online way. However, these approaches still suffer from the following shortcomings: the pseudo labels always remain fixed [16,17], and the teacher detector is converged to the student detector in the late stage of training [18–20], thus the labeling process degenerates into the self-labeling manner and suffers from the same limitations. Similar with our method, [22] proposes the co-rectify method to train two models simultaneously and takes the average on two predictions sets as pseudo labels, which is the only prior work adopting the co-training framework to the best of our knowledge. However, we conduct quantitative comparison on the pseudo label quality,

and the experimental results validate the superiority of our CrossRectify method compared with co-rectify.

We carry out extensive experiments on both 2D and 3D semi-supervised object detection tasks to verify the effectiveness and versatility of the proposed CrossRectify method. As illustrated in Section 5.3, our method obtains consistent and substantial improvements compared with the state-of-the-art SSOD methods on the Pascal VOC, MS-COCO, and SUN-RGBD benchmark datasets, improving by >1% absolute AP margins.

Our main contributions are summarized as follows:

- 1) We point out that the performances of self-labeling-based SSOD approaches are always limited, and the reason behind such phenomenon lies in that the detector can neither discern nor rectify the misclassified pseudo bounding boxes predicted by itself.
- 2) We propose an effective approach named CrossRectify to discern and rectify the misclassified pseudo bounding boxes using the disagreements between two detectors, which can address the inherent limitations of self-labeling and improve the detection performance.
- 3) We conduct extensive experiments on both 2D and 3D object detection benchmark datasets, and the results verify the superiority of the proposed CrossRectify approach, compared with state-of-the-art approaches.

In the remainder of this paper, we briefly review several related works in Section 2. Then we describe the two limitations of self-labeling-based SSOD approaches in Section 3, and provide the technical details of the proposed CrossRectify method in Section 4. Finally, we display the detection performances of 2D and 3D SSOD tasks in Section 5 and conclude the whole paper in Section 6.

2. Related work

2.1. 2D and 3D object detection

Object detection is one of the most significant tasks in computer vision, including 2D and 3D scenes. In the field of 2D object detection, the structures of detectors can be categorized into single-stage (SSD [1], etc) and two-stage (Faster-RCNN [2], cascade RCNN [23], etc), depending on whether a region proposal network is utilized. Although these detectors have reached outstanding performances, their training processes heavily rely on a large amount of labeled data with bounding box annotations, which are always laborious and expensive to acquire [7]. In this paper, we focus on how to leverage a large amount of unlabeled data for performance increment. For fair comparisons with existing works, we conduct experiments over the SSD300 and Faster-RCNN-FPN detector structures on 2D object detection, and over the VoteNet [24] structure on 3D object detection.

2.2. Semi-supervised object detection (SSOD)

The existing SSOD approaches can be categorized as follows. **Consistency regularization** Many of the existing SSOD methods utilize the consistency regularization proposed in semi-supervised learning (SSL), such as CSD [25], ISD [26], PL [27], etc. The key idea of consistency regularization is to require the detector to predict consistently on both weak- and strong- augmented versions of the same input. We point out that the consistency-regularization-based methods can be regarded as the special case of the self-labeling training scheme, because the detector is supervised by the pseudo labels on weak-augmented image predicted by itself. Note that these studies report large performance improvements over the fully-supervised baselines, but we point out that the experimental

settings are somehow unfair. In fact, under the same data augmentation, the performance increments brought by consistency regularization are always observed limited. We analyze the reason behind such phenomenon in this paper, and point out the inherent limitations of the self-labeling training scheme, as well as the consistency regularization.

Teacher-student mutual learning Beyond the consistency regularization, there are a few SSOD works based on the teacher-student mutual learning framework [21]. In an offline or online manner, the pseudo labels are generated by the teacher model, instead of by the student model itself. As for the former, STAC [16] and Uncertainty [17] first pre-train the teacher model with available labeled data, then utilize the teacher to annotate the entire unlabeled data. However, the pseudo labels are generated only once and remain fixed during semi-supervised training, thus the final performance of student model is limited by that of the teacher model. As for the latter, UBTeacher [18], Humble Teacher [19], and Soft Teacher [20] compute the exponential moving average of student as teacher and the teacher is utilized to generate pseudo labels during the semi-supervised training process. However, we observe that the teacher is converged to the student in the late stage of training, which indicates that the labeling process degenerates into the self-labeling manner and suffers from the same limitations.

Co-rectify and CPS As another research line of semi-supervised learning, co-training methods are proposed to train two models in a collaborative manner, such as Deep Co-Training [12], Dual Student [13], JoCoR [14]. Each model can learn from the pseudo labels predicted by its counterpart, which seems a promising way to mitigate the limitations of self-labeling methods. The only prior work taking the idea of co-training for the SSOD task is co-rectify [22]. In [22], the pseudo boxes for unlabeled data are first predicted by one detector, then refined by the corresponding predictions from another model, with probability scores and coordinates being averaged. Besides, a recent study [28] also adopts co-training and proposes the cross pseudo supervision (CPS) for the semi-supervised semantic segmentation task, where each model is supervised by the pseudo maps predicted by the other model. We note that CPS can be adapted for the SSOD task. However, we conduct quantitative comparisons to show that both co-rectify and CPS cannot fully exploit the advantages of multiple models and improve the quality of pseudo labels.

3. Problem analysis

In this section, we first introduce the preliminaries in semi-supervised object detection (SSOD), then analyze the inherent limitations of self-labeling-based SSOD methods.

3.1. Preliminaries

Under the semi-supervised setting, an object detector f is trained on a labeled dataset $D_l = \{\mathbf{x}_i^l, \mathbf{y}_i^l\}_{i=1}^{N_l}$ with N_l samples and an unlabeled dataset $D_u = \{\mathbf{x}_j^u\}_{j=1}^{N_u}$ with N_u samples. For a labeled image $\mathbf{x}^l$, its annotation $\mathbf{y}^l = (\mathbf{c}, \mathbf{t})$ contains the category labels $\mathbf{c}$ and coordinates $\mathbf{t}$ of all foreground objects.

Overall, the detector model f is optimized by minimizing the supervised loss L_S on labeled data and unsupervised loss L_U on unlabeled data, formulated as:

$$L = L_S + \lambda_U \cdot L_U, \quad (1)$$

where λ_U denotes the weight factor. Generally, the supervised loss L_S consists of the classification loss l_{cls} and coordinate regression loss l_{reg} :

$$L_S = l_{cls}(f_{cls}(\mathbf{x}^l), \mathbf{c}) + l_{reg}(f_{loc}(\mathbf{x}^l), \mathbf{t}), \quad (2)$$

Table 1

Results of self-labeling-based semi-supervised object detection methods under various data augmentations. The benchmark dataset is Pascal VOC and the detector structure is SSD300. "SeLa" stands for self-labeling, "identical", "HF" and "MU" stand for no data augmentation, horizontal flip augmentation (CSD) [25] and mix-up augmentation (ISD) [26], respectively. The figures in brackets are the performance increments over the fully-supervised baselines, which always seem trivial for the self-labeling-based methods.

Method	$A(\cdot)$	Labeled	Unlabeled	AP ₅₀ (%)
Supervised	identical	VOC07	-	71.73
SeLa	identical	VOC07	VOC12	72.13 (+0.40)
Supervised	identical	VOC0712	-	77.37 (+5.64)
Supervised	HF	VOC07	-	71.89
SeLa	HF	VOC07	VOC12	72.35 (+0.46)
Supervised	HF	VOC0712	-	77.26 (+5.37)
Supervised	MU	VOC07	-	73.04
SeLa	MU	VOC07	VOC12	73.50 (+0.46)
Supervised	MU	VOC0712	-	78.83 (+5.79)

where $f_{cls}(\mathbf{x}^l)$ and $f_{loc}(\mathbf{x}^l)$ stand for probabilities and coordinates predicted by the classification and localization branch of detector f , respectively.

In each training iteration, the self-labeling-based SSOD method utilizes the current detector to predict bounding boxes on the unlabeled inputs $\mathbf{x}^u$, and then selects the pseudo labels $\hat{\mathbf{y}}$ with confidence larger than the threshold τ , and finally computes the unsupervised loss:

$$L_U = l_{cls}(f_{cls}(\mathbf{x}^u), \hat{\mathbf{c}}) + l_{reg}(f_{loc}(\mathbf{x}^u), \hat{\mathbf{t}}), \quad (3)$$

where $\hat{\mathbf{y}} = (\hat{\mathbf{c}}, \hat{\mathbf{t}}) = (\arg\max f_{cls}(\mathbf{x}^u), f_{loc}(\mathbf{x}^u))$ and $\max f_{cls}(\mathbf{x}^u) > \tau$ are correspondingly satisfied.

As the special case of self-labeling training, the consistency-regularization-based methods introduce weak data augmentation $\alpha(\cdot)$ and strong data augmentations $A(\cdot)$, and the Eq. (3) is updated as:

$$L_U = l_{cls}(f_{cls}(A(\mathbf{x}^u)), \arg\max f_{cls}(\alpha(\mathbf{x}^u))) + l_{reg}(f_{loc}(A(\mathbf{x}^u)), f_{loc}(\alpha(\mathbf{x}^u))). \quad (4)$$

Note that as for the consistency regularization, the loss term Eq. (4) can be computed on both labeled and unlabeled data, and the strong augmentations can also boost the performances of fully-supervised training [26]. Accordingly, the total loss in Eq. (1) is augmented as $L = L_S + \lambda_U \cdot (L_U(\mathbf{x}^l) + L_U(\mathbf{x}^u))$, and the fully-supervised baseline is trained by optimizing $L = L_S + \lambda_U \cdot L_U(\mathbf{x}^l)$.

3.2. Limitations of self-labeling

To verify the performances of existing self-labeling-based SSOD methods, we conduct experiments on the Pascal VOC benchmark dataset [11] based on the SSD300 structure [1]. We use the trainval set of VOC07 as labeled data and trainval set of VOC12 as unlabeled data, and finally report the AP₅₀ performance on the test set of VOC07. The confidence threshold τ is fixed as 0.5, similar with [25] and [26]. For comparison, we conduct fully-supervised training with same hyper-parameters (batch size, iteration number, etc) as baseline. As shown in Table 1, the AP₅₀ improvement achieved by self-labeling methods is only 0.40%. Besides, we test with two representative consistency-regularization-based methods, namely CSD [25] and ISD [26]. Similarly, the detection results achieve only 0.46% absolute AP₅₀ gains over the baseline, which indicates the inefficiency of self-labeling training scheme.

To reveal the reason behind such phenomenon and find out possible solutions, we introduce the ground-truth category labels of all pseudo bounding boxes for in-depth analysis. Although all existing self-labeling-based SSOD methods generate pseudo labels

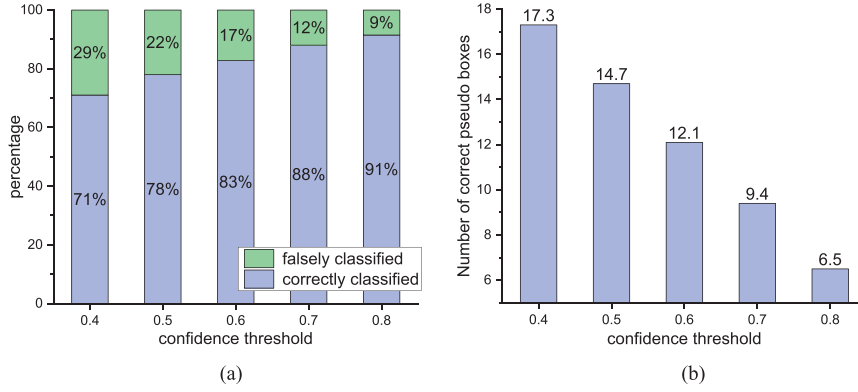


Fig. 2. Pseudo label quality under different confidence thresholds in self-labeling training. (a) Precision of pseudo labels. (b) Average number of correctly classified pseudo boxes in each iteration. (For interpretation of the references to color in this figure legend, please refer to the online version of this article.)

Table 2

Analysis on the limitations of self-labeling methods. The benchmark dataset is Pascal VOC and the detector structure is SSD300. TP (FP) stands for the correctly (falsely) classified pseudo boxes. “SeLa” stands for self-labeling, which uses both TP and FP for training. The figures in brackets are the AP_{50} increments over the fully-supervised baseline.

Method	Labeled	Unlabeled	τ	AP_{50}
Supervised	VOC07	-	-	71.73
SeLa	VOC07	VOC12	0.5	72.13 (+0.40)
SeLa	VOC07	VOC12	0.5→0.8	72.09 (+0.36)
SeLa	VOC07	VOC12	0.8	72.12 (+0.39)
SeLa (use TP and discard FP)	VOC07	VOC12	0.5	74.03 (+2.30)
SeLa (use TP and random labeled FP)	VOC07	VOC12	0.5	73.87 (+2.14)
SeLa (use GT labels for TP and FP)	VOC07	VOC12	0.5	74.86 (+3.13)
Supervised	VOC0712	-	-	77.37 (+5.64)

by selecting bounding boxes with high confidence scores to ensure the quality of pseudo labels, Fig. 2(a) illustrates that part of high-confidence pseudo bounding boxes can also be misclassified. We name these incorrect boxes “self-errors” for clarity. Since all pseudo boxes are predicted by the detector, it is impossible for the detector itself to discern the self-errors, which we summarize as the inherent limitation of the self-labeling process. We further conduct an experiment to illustrate how much can such limitation affect the detection performance: in each training iteration, when we use the correct pseudo bounding boxes and discard the incorrect ones for training, the AP_{50} result increases from 72.13% to 74.03% (see the 2nd and 5th rows in Table 2). Besides, we note that the detection performance cannot be improved by naively increasing the confidence threshold τ . As shown in Fig. 2(a) and (b), the precision of pseudo bounding boxes increases from 71% to 91% as the threshold increases from 0.4 to 0.8, while a large threshold can also overkill the correct pseudo boxes and waste the unlabeled training data. Correspondingly, we conduct self-labeling training under three settings of the threshold τ : (i) fixed as 0.5, (ii) fixed as 0.8, (iii) rising from 0.5 to 0.8 gradually during the whole training process. As displayed in Table 2 (from 2nd row to 4th row), the trade-off between precision and recall leads to similar final performances (about 72.1% AP_{50}).

Since the self-errors cannot be discerned during the self-labeling process, it is also impossible for the detector to rectify them, which we summarize as the second limitation of the self-labeling process. In each training iteration, when we utilize the ground-truth (GT) category labels for all pseudo bounding boxes, the AP_{50} result increases to 74.86%, obtaining a 3.13% absolute gain rather than 0.40% (see the penultimate row in Table 2). Such phenomenon shows the effectiveness of pseudo label rectification. Besides, we find another interesting phenomenon: when we replace the misclassified category labels with random labels (dubbed random-errors) during the self-labeling training process, the final

performance increases to 73.87% (see the third row from the bottom in Table 2). We also conduct experiments on the Faster-RCNN-FPN structure [2] and observe the similar trend: the AP_{50} obtains with a 0.2% gain by replacing the misclassified labels with random labels during training. Such phenomena imply that the detector model can be misguided more severely by self-errors than random-errors.

Based on the above experimental analysis, we draw that two inherent limitations in self-labeling-based SSOD methods will lead the detector to be misguided by self-errors, and self-errors will hurt the detector performance even worse than random-errors.

4. Methodology

4.1. CrossRectify

Since one single detector can neither discern nor rectify its self-errors, an intuitive idea is to utilize another model to deal with them. Inspiring by the fact that two models with same structure but different initialization can yield different predictions on the same input, we present the **CrossRectify** method to address the inherent limitations of self-labeling.

In CrossRectify, two detectors with the same structure but distinct initialization¹, f_A and f_B , are trained simultaneously. Both detectors are trained by jointly optimizing the supervised and unsupervised loss in Eq. (1). For simplicity, we only introduce how to generate pseudo bounding boxes $\hat{y}_A$ for training detector f_A , since the pseudo boxes $\hat{y}_B$ for training detector f_B are generated in the same way. There are three steps in generating $\hat{y}_A$, including: 1) conducting detector feed-forward; 2) matching predicted bounding

¹ Take 2D object detection task for example. The backbone parameters are both initialized by the ImageNet-pretrained model, while the parameters of detection heads are randomly initialized.

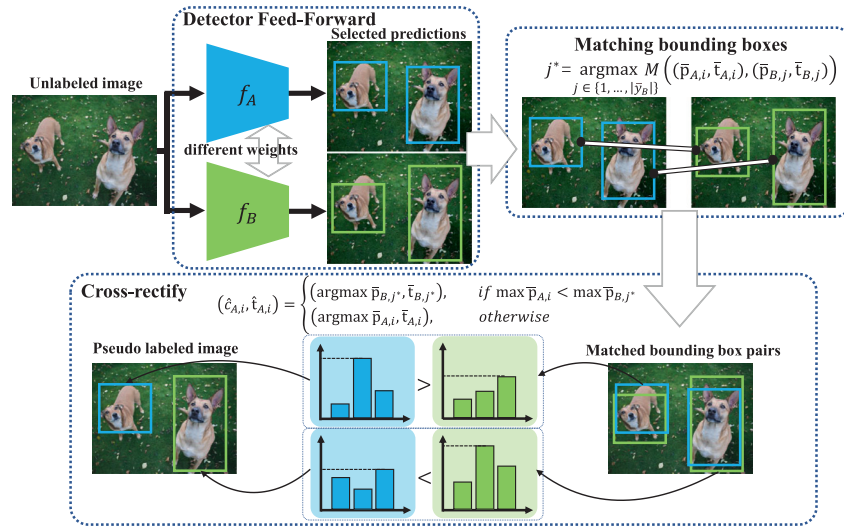


Fig. 3. Overview of the pseudo label generation process in the proposed algorithm. Refer to Section 4 for more details. (For color discrimination in this figure, please refer to the online version of this article.)

boxes; **3)** cross-rectifying the matched boxes to generate pseudo labels. They will be explained in details sequentially. The label generation process of $\hat{\mathbf{y}}_A$ is illustrated in Fig. 3 and briefly summarized in Algorithm 1.

Algorithm 1 Generating pseudo bounding boxes $\hat{\mathbf{y}}_A$ via CrossRectify for training detector f_A .

Input: Object detectors f_A and f_B , and unlabeled input.

Output: The pseudo bounding boxes for training f_A .

- 1: Utilize f_A and f_B to predict on unlabeled input
- 2: Select the predicted boxes with their max probability scores larger than threshold τ , denoted as $\bar{\mathbf{y}}_A$ and $\bar{\mathbf{y}}_B$.
- 3: Compute the matching metrics $M(\cdot, \cdot)$ for each bounding boxes in $\bar{\mathbf{y}}_A$ with all bounding boxes in $\bar{\mathbf{y}}_B$.
- 4: Find the best matching box for each bounding boxes in $\bar{\mathbf{y}}_A$ (using Eq.(5)).
- 5: Compare max probability scores between each matched pair and obtain the pseudo boxes (using Eq.(8))

Detector feed-forward In each training iteration, we utilize two detector models f_A and f_B to predict on the unlabeled inputs, then select the bounding boxes with max probability scores higher than threshold τ , denoted as $\bar{\mathbf{y}}_A = (\bar{\mathbf{p}}_A, \bar{\mathbf{t}}_A)$ and $\bar{\mathbf{y}}_B = (\bar{\mathbf{p}}_B, \bar{\mathbf{t}}_B)$. $\bar{\mathbf{p}}$ and $\bar{\mathbf{t}}$ denote the probability scores and coordinates of all predicted bounding boxes.

Matching bounding boxes For each box in $\bar{\mathbf{y}}_A$, we search its best match box among all boxes in $\bar{\mathbf{y}}_B$. For example, for i th box $(\bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i})$ in $\bar{\mathbf{y}}_A$, the matching process is formulated as:

$$j^* = \operatorname{argmax}_{j \in \{1, \dots, |\bar{\mathbf{y}}_B|\}} M((\bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i}), (\bar{\mathbf{p}}_{B,j}, \bar{\mathbf{t}}_{B,j})), \quad (5)$$

where $M(\cdot, \cdot)$ stands for the matching metric and is slightly different for kinds of detectors. For the Faster-RCNN [2], $M((\bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i}), (\bar{\mathbf{p}}_{B,j}, \bar{\mathbf{t}}_{B,j}))$ is the area of intersection over union (IoU) between two boxes as:

$$M((\bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i}), (\bar{\mathbf{p}}_{B,j}, \bar{\mathbf{t}}_{B,j})) = \frac{\bar{\mathbf{t}}_{A,i} \cap \bar{\mathbf{t}}_{B,j}}{\bar{\mathbf{t}}_{A,i} \cup \bar{\mathbf{t}}_{B,j}}. \quad (6)$$

Specifically, if the IoU areas between $(\bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i})$ and all boxes in $\bar{\mathbf{y}}_B$ are all below a certain threshold δ , we create a virtual bounding box $\bar{\mathbf{t}}_{B,j^*}$ to match it, as $(\bar{\mathbf{p}}_{B,j^*}, \bar{\mathbf{t}}_{B,j^*}) = (\bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i})$ and the matching metric $M(\cdot, \cdot)$ equals 1. For the SSD structure [1],

$M((\bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i}), (\bar{\mathbf{p}}_{B,j}, \bar{\mathbf{t}}_{B,j}))$ equals 1 if two boxes are based on the same anchor, otherwise 0. Note that $M(\cdot, \cdot)$ for SSD can also be specified as the IoU areas as like that for Faster-RCNN, but we find it more effective to adopt the anchor correspondence in our experiments. For the VoteNet [24], the matching metric $M(\cdot, \cdot)$ is specified as the negative Euclidean distance between the centers of two bounding boxes, computed as:

$$M((\bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i}), (\bar{\mathbf{p}}_{B,j}, \bar{\mathbf{t}}_{B,j})) = -\|C(\bar{\mathbf{t}}_{A,i}) - C(\bar{\mathbf{t}}_{B,j})\|_2, \quad (7)$$

where $C(\cdot)$ denotes the center of a certain box.

Cross-rectifying Based on the matched bounding box pair, each pseudo bounding box $(\hat{\mathbf{e}}_{A,i}, \hat{\mathbf{t}}_{A,i})$ within $\hat{\mathbf{y}}_A$ can be generated as:

$$(\hat{\mathbf{e}}_{A,i}, \hat{\mathbf{t}}_{A,i}) = \begin{cases} (\operatorname{argmax}_{j} \bar{\mathbf{p}}_{B,j}, \bar{\mathbf{t}}_{B,j}), & \text{if } \max \bar{\mathbf{p}}_{A,i} < \max \bar{\mathbf{p}}_{B,j} \\ (\operatorname{argmax}_{i} \bar{\mathbf{p}}_{A,i}, \bar{\mathbf{t}}_{A,i}), & \text{otherwise.} \end{cases} \quad (8)$$

Note that the Eq. (8) covers two situations. (a) When both detectors predict the same class on a certain object, we adopt it as the pseudo label, since two decisions are more reliable than one decision. (b) When two detectors have disagreements on a certain object, such bounding box tends to be unreliable. To this end, the bounding box with higher confidence is regarded as the pseudo label. The rationality behind such cross-rectifying mechanism lies in that the bounding boxes with higher confidence scores are more likely to be correctly classified (see Fig. 2(a)). Thus, the wrong elements in $\bar{\mathbf{y}}_A$ can be both discerned and rectified in such manner.

When the training process ends, we evaluate the performance of one single detector. Moreover, to exploit the different detection abilities, we propose to adopt the weighted boxes fusion (WBF) [29] strategy to ensemble two predictions sets. The corresponding performance is denoted as **CrossRectify***.

4.2. Comparisons with other works

Now we conduct quantitative analysis to show the superiority of CrossRectify on improving pseudo label quality, comparing with other recent works.

Teacher-student mutual learning As discussed in Section 2.2, some recent SSOD works such as STAC [16], Uncertainty [17], UBTeacher [18], Humble Teacher [19], Soft Teacher [20], are established on the offline/online teacher-student mutual learning [21]. These works can alleviate the self-errors in self-labeling process by introducing another separate object detector for generation of pseudo labels. However, as for offline methods [16,17], the pseudo

Table 3
2D Semi-supervised Object Detection performances (AP_{50}) on Pascal VOC benchmark dataset.

Model	Backbone	Method	Labeled	Unlabeled	Threshold	AP_{50}
SSD300	VGG-16	Supervised	VOC07	-	-	71.73
		Self-Labeling	VOC07	VOC12	0.5	72.13 (+0.40)
		Online Teacher-Student Mutual Teaching	VOC07	VOC12	0.5	72.56 (+0.83)
		Offline Teacher-Student Mutual Teaching [17]	VOC07	VOC12	-	72.52 (+0.79)
		Cross Pseudo Supervision [28]	VOC07	VOC12	-	72.56 (+0.83)
		Co-rectify [22]	VOC07	VOC12	0.5	72.48 (+0.75)
		CrossRectify (ours)	VOC07	VOC12	0.5	73.56 (+1.83)
		CrossRectify* (ours)	VOC07	VOC12	0.5	74.97 (+3.24)
		Supervised + MixUp	VOC07	VOC12	0.5	73.04
		Self-Labeling + MixUp (ISD [26])	VOC07	VOC12	0.5	73.50 (+0.46)
		CrossRectify + MixUp (ours)	VOC07	VOC12	0.5	74.91 (+1.87)
		CrossRectify* + MixUp (ours)	VOC07	VOC12	0.5	76.16 (+3.12)
Faster-RCNN-FPN	ResNet-50	Supervised	VOC07	-	-	76.90
		CSD [25]	VOC07	VOC12	-	77.50 (+0.60)
		STAC [16]	VOC07	VOC12	-	77.50 (+0.60)
		Co-rectify [22]	VOC07	VOC12	0.5	79.20 (+2.30)
		Combating Noise [35]	VOC07	VOC12	-	80.60 (+3.70)
		Humble Teacher [19]	VOC07	VOC12	0.7	80.94 (+3.94)
		Unbiased Teacher [18]	VOC07	VOC12	0.7	80.51 (+3.61)
		CrossRectify (ours)	VOC07	VOC12	0.7	81.56 (+4.66)
		CrossRectify* (ours)	VOC07	VOC12	0.7	82.34 (+5.44)

Table 4
2D Semi-supervised object detection performances ($AP_{50:95}$) on MS-COCO benchmark dataset.

Model	Backbone	Method	Proportion of labeled data			
			1%	2%	5%	10%
Faster-RCNN-FPN	ResNet-50	Supervised	9.05 $\pm$ 0.16	12.70 $\pm$ 0.15	18.47 $\pm$ 0.22	23.86 $\pm$ 0.81
		CSD [25]	10.51 $\pm$ 0.06	13.93 $\pm$ 0.12	18.63 $\pm$ 0.07	22.46 $\pm$ 0.08
		STAC [16]	13.97 $\pm$ 0.35	18.25 $\pm$ 0.25	24.38 $\pm$ 0.12	28.64 $\pm$ 0.21
		Unbiased Teacher [18]	20.75 $\pm$ 0.12	24.30 $\pm$ 0.07	28.27 $\pm$ 0.11	31.50 $\pm$ 0.10
		Humble Teacher [19]	16.96 $\pm$ 0.38	21.72 $\pm$ 0.24	27.70 $\pm$ 0.15	31.61 $\pm$ 0.28
		Co-rectify [22]	18.05 $\pm$ 0.15	22.45 $\pm$ 0.15	26.75 $\pm$ 0.05	30.40 $\pm$ 0.05
		Combating Noise [35]	18.41 $\pm$ 0.10	24.00 $\pm$ 0.15	28.96 $\pm$ 0.29	32.43 $\pm$ 0.20
		Soft Teacher [20]	20.46 $\pm$ 0.39	26.20 $\pm$ 0.10	30.74 $\pm$ 0.08	34.04 $\pm$ 0.14
		CrossRectify (ours)	21.90 $\pm$ 0.11	26.70 $\pm$ 0.07	31.70 $\pm$ 0.04	34.89 $\pm$ 0.07
		CrossRectify* (ours)	22.50 $\pm$ 0.12	27.60 $\pm$ 0.07	32.80 $\pm$ 0.05	36.30 $\pm$ 0.07

labels are generated only once and remain fixed when training student detector, so the student performance is upper-bounded by that of teacher. For instance, we conduct experiments with Faster-RCNN-FPN detector on the MS-COCO benchmark dataset under 10% degree of supervision. The AP_{50} performance of teacher detector after fully-supervised pre-training is 23.86%, while that of student detector supervised by teacher detector only increases with a 3.30% absolute gain, far away from the results in Table 4 (34.89% AP_{50}). Similar phenomenon can be observed with respect to the SSD300 structure on the Pascal VOC benchmark dataset in Table 3 (obtaining only a 0.79% AP_{50} gain).

As for online methods [18–20], the teacher detector is converged to the student detector and yields similar predictions in the late stage of training, thus the pseudo label generation process degenerates to the self-labeling process and suffers from the same limitations. For instance, we conduct online teacher-student mutual learning based on SSD300 and Pascal VOC. As shown in Fig. 4(a), the average KL-divergence between probability scores predicted by teacher and student detector reaches zero in the last 40k iterations. Correspondingly, the detection performance shown in Table 3 also indicate the ineffectiveness of online teacher-student mutual learning.

Co-rectify and CPS Recently, a co-training based SSOD method named co-rectify has been proposed in [22], which is the only prior work taking the idea of co-training in the SSOD task to the best of our knowledge. In co-rectify, the pseudo bounding boxes are first predicted by one detector, then refined by corresponding

predictions from another model, with probability scores and coordinates being averaged. Besides, we notice that a recent work proposes cross pseudo supervision (CPS) [28] for the semi-supervised semantic segmentation task and achieves the state-of-the-art performances, where each model directly takes the predictions from the other model as pseudo labels. CPS can be adapted to the SSOD task, as each detector is supervised by the other detector. However, as shown in Fig. 4(b), the precision values of pseudo label of these methods are inferior to that of CrossRectify (conducting semi-supervised training with SSD300 on Pascal VOC). We consider the reason behind such phenomenon lies in that simply averaging multiple predictions (co-rectify) or directly taking predictions from other models as supervision (CPS) cannot fully exploit the advantages of multiple models, in comparison to our cross-rectifying mechanism. Their inferior performances shown in Tables 3 and 6 also validate our consideration. Besides, we also investigate more alternative strategies on pseudo label rectification and observe that cross-rectifying turns out to be most effective strategy (as detailed in Section 5.4).

5. Experiments

5.1. Datasets and evaluation metrics

2D semi-supervised object detection We evaluate the proposed CrossRectify on two widely-used benchmark datasets, i.e., Pascal VOC [11] and MS-COCO [30]. Pascal VOC has 20 object categories.

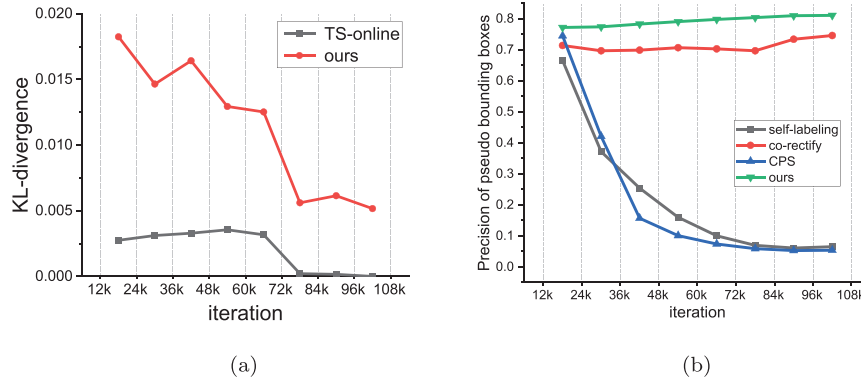


Fig. 4. Left: (a) comparison on the average KL-divergence between probability scores predicted by two detectors over each 12k iterations. “TS-online” stands for the teacher-student mutual learning in the online manner. Right: (b) comparison on the average precision of pseudo bounding boxes among different methods over each 12k iterations. (For interpretation of the references to color in this figure legend, please refer to the online version of this article.)

We take the VOC07 trainval set (5,011 images) as labeled and the VOC12 trainval set (11,540 images) as unlabeled. The detection performance is evaluated on the VOC07 test set (4,952 images) using the VOC style AP_{50} metric. MS-COCO has 80 object categories. We follow the same settings as that in [16] to randomly sample 1/2/5/10% of the COCO2017 train set (118,287 images) as labeled and take the remaining part as unlabeled. Also, we create five data folds under each degree of supervision, and finally report the mean and standard deviation from five results. The detection performance is evaluated on the COCO2017 val set (5,000 images) using the COCO style $AP_{50:95}$ metric.

3D semi-supervised object detection We follow [31] to conduct experiments on the SUN-RGBD benchmark dataset [32]. We randomly sample 5% of 5285 training samples as labeled and take the remaining part as unlabeled. The detection performances is evaluated on 5050 validation samples, using both AP_{25} and AP_{50} metrics.

5.2. Implementation details

Detector structures We carry out experiments on the Pascal VOC dataset with two detector structures, that are SSD300 [1] with VGG-16 backbone and Faster-RCNN-FPN [2,33] with ResNet-50 backbone. The latter structure is also utilized in experiments on the MS-COCO dataset. As for 3D detection, we utilize VoteNet [24] with PointNet++ backbone [34].

Training settings We utilize the Pytorch implementation² to train SSD300 on Pascal VOC. Within a total of 120k iterations, we conduct fully-supervised training in the first 12k iterations as warm-up. We ramp-up/down the unsupervised loss weight λ_U , and set threshold τ and batch size as 0.5 and 32 according to Jeong et al. [26]. We utilize the Detectron2 platform³ to train Faster-RCNN-FPN on Pascal VOC. We train a total of 36k iterations with first 6k being fully-supervised warm-up, and adopt the same data augmentation strategy as that in [18]. We set λ_U as 2.0 and threshold τ as 0.7 following [18]. The batch sizes for labeled data and unlabeled data are both 16. The threshold δ on matching metric is 0.5. To show the generality across different platforms of our CrossRectify method, we adopt the MMDetection⁴ to train Faster-RCNN-FPN on MS-COCO. We train a total of 180k iterations and adopt the data augmentation strategies in [20]. Under 1% degree of supervision, we conduct fully-supervised warm-up in the first 80k iterations to ensure the stability of training. We set λ_U

as 4.0 and threshold τ as 0.9 according to [20]. The batch sizes for labeled data and unlabeled data are respectively 8 and 32. The threshold δ on matching metric is 0.5. To train VoteNet on SUN-RGBD, we first conduct fully-supervised pre-training by 900 iterations, then conduct semi-supervised training by 1k iterations, following [31].

Note that the exponential moving average (EMA) strategy is commonly used for the pseudo label-based methods, since a detector model aggregated by EMA can yield more conservative and stable predictions than the detector itself [18]. For fair comparisons, We follow such common practice in our experiments, as we utilize the EMAs of two detectors to conduct the detector feed-forward process.

5.3. Results

Pascal VOC Table 3 shows the results of our CrossRectify method compared with other training frameworks on Pascal VOC. As for the SSD300 detector, we take the 71.73% AP_{50} performance of fully-supervised training as the baseline. As can be seen, our proposed method can obtain a 73.65% AP_{50} result, while the results of all compared approaches are only about 72.50%. Such comparison validates the effectiveness of our CrossRectify on improving pseudo label quality. Besides, the WBF-merged [29] results from both detectors can further boost the final performances, denoted as CrossRectify*. Under the mix-up data augmentation [10], our CrossRectify method can still show better performance than the self-labeling-based method, ISD [26] (by a 1.41% margin).

As for the Faster-RCNN-FPN detector, we compare CrossRectify with previous methods, and our method can improve the AP_{50} result with a 4.66% margin over fully-supervised baseline, achieving the state-of-the-art performance. Note that Unbiased Teacher [18] reports the performance based on the COCO style AP_{50} metric in their paper. For a fair comparison, we instead adopt the VOC style AP_{50} metric for Unbiased Teacher, and the AP_{50} raises from 77.37% to 80.51%, still surpassed by that of CrossRectify with a 0.59% margin.

MS-COCO Table 4 shows the performances of our CrossRectify method compared with previous state-of-the-arts on the MS-COCO dataset. Under different degrees of supervision 1%, 2%, 5% and 10%, the proposed CrossRectify can obtain consistent and substantial improvements, surpassing those of Soft Teacher [20] by 1.46%, 0.50%, 0.96%, and 0.85% $AP_{50:95}$ margins. These comparative results further verify the effectiveness of the proposed method. Moreover, we visualize the pseudo bounding boxes for some unlabeled images in Fig. 5. Compared with self-labeling training scheme, our method can yield more accurate pseudo boxes.

² <https://github.com/amdegroot/ssd.pytorch>.

³ <https://github.com/facebookresearch/detectron2>.

⁴ <https://github.com/open-mmlab/mmdetection>.



Fig. 5. The visual comparisons between Self-Labeling (the first row) and the proposed CrossRectify method (the second row) on MS-COCO under 1% degree of supervision. (For color discrimination in this figure, please refer to the online version of this article.)

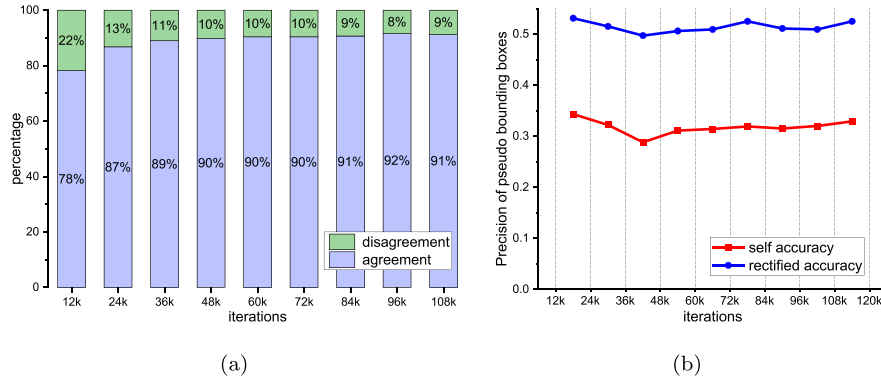


Fig. 6. (a) Ratio of disagreement between two models. (b) Precision of pseudo bounding boxes in disagreement between two models.

Table 5

3D Semi-supervised object detection performances (AP_{25} and AP_{50}) on SUN-RGBD benchmark dataset.

Model	Backbone	Method	AP_{25}	AP_{50}
VoteNet	PointNet+	Supervised	29.9 ± 1.5	10.5 ± 0.5
		SESS [36]	34.2 ± 2.0	13.1 ± 1.0
		3DloUMatch [31]	39.0 ± 1.9	21.1 ± 1.7
		CrossRectify (ours)	42.1 ± 1.7	23.0 ± 1.2

SUN-RGBD Table 5 shows the comparison with all previous works (i.e., SESS [36] and 3DloUMatch [31]) on the SUN-RGBD benchmark dataset. Under 5% degree of supervision, the performance of our CrossRectify can outperform that of the state-of-the-art 3DloUMatch method by 3.1 AP_{25} and 1.9 AP_{50} margins. The results validate the efficiency of CrossRectify on 3D semi-supervised object detection task. We omit the WBF-merging performance CrossRectify*, because WBF does not support on 3D bounding boxes with different rotation angles.

5.4. Empirical study

How many labels are found to be in disagreement between two models? We monitor the percentage of pseudo labels in disagreement between models A and B during the whole training process. The detector structure is SSD300 and the dataset is VOC07 (labeled) plus VOC12 (unlabeled). As shown in Fig. 6(a), there are 22% pseudo labels in disagreement in the early stage of training, and around 10% in the tail stage of training, which indicates that the pseudo label quality keeps improving during the whole process. Also, we measure the precision of pseudo labels in disagreement during the whole process. As can be seen in Fig. 6(b), the precision of pseudo labels in disagreement is only $\sim 30\%$, while our

CrossRectify method can always bring about a $\sim 20\%$ improvement. As we illustrate in Table 6, such $\sim 20\%$ margin can contribute to a 1.92 AP_{50} gain with respect to the final detection performance.

Pseudo label rectification strategy As we mentioned in the last paragraph, our CrossRectify can improve the precision of pseudo labels in disagreement with a $\sim 20\%$ margin during the whole training process. As we illustrate in Table 6, such $\sim 20\%$ margin can contribute to a 1.92 AP_{50} gain with respect to the final detection performance. Moreover, it can be seen that other strategies such as intersection (only use pseudo labels in agreement), difference set (only use pseudo labels in disagreement predicted by the model itself), cross pseudo supervision (directly use pseudo labels predicted by the other model), co-rectify (use averaged pseudo labels predicted by two models), cannot significantly improve the detection performance in comparison with CrossRectify.

Ablation study on threshold δ when matching pseudo bounding boxes We have examined different threshold δ (ranging from 0.4 to 0.9) with Faster-RCNN-FPN on MS-COCO, under the degree of supervision 10%. As can be found in Table 7, the detection performance is robust with threshold δ ranging from 0.4 to 0.7. A high value such as 0.9 may lead too many pseudo bounding boxes unmatched and the CrossRectify is just skipped, so the training process will degrade to vanilla self-labeling-based training.

Extension to more detectors Our proposed CrossRectify method can be easily extended to train more than two detectors simultaneously. Specifically, during the pseudo label rectification process, each pseudo bounding box is re-labeled by the majority of all predicted classes, and re-located by the average of all predicted co-ordinates. As displayed in Table 8, CrossRectify over four SSD300 detectors can bring a 0.06% AP_{50} improvement for each detector on average. We believe that two-detector scenario is already able to cross-rectify the misclassified pseudo labels adequately.

Table 6

Performance comparison among different pseudo label generation methods. Figures in parentheses are the improvement over the fully-supervised baseline. Best results are in **bold**.

Model	Labeled	Unlabeled	Strategy	AP ₅₀
SSD300	VOC07	-	-	71.73
	VOC07	VOC12	self-labeling	72.13 (+0.40)
	VOC07	VOC12	intersection	72.59 (+0.86)
	VOC07	VOC12	difference set	65.52 (-6.21)
	VOC07	VOC12	cross pseudo supervision	72.56 (+0.83)
	VOC07	VOC12	co-rectify	72.48 (+0.75)
	VOC07	VOC12	CrossRectify	73.65 (+1.92)

Table 7

Ablation study on the pre-defined threshold δ when matching bounding boxes.

δ	0.4	0.5	0.6	0.7	0.8	0.9
AP _{50:95}	34.87	34.89	34.84	34.72	34.31	34.13

Table 8

Extension to four detector models on Pascal VOC dataset.

Model	Index	Single	Average	WBF-Merged
SSD300	detector #1	73.67	73.65	74.83
	detector #2	73.63		
	detector #1	73.60	73.71	75.84
	detector #2	73.71		
	detector #3	73.80		
	detector #4	73.73		

6. Conclusion

In this paper, we first illustrate the fact that the performances of self-labeling-based semi-supervised object detection (SSOD) approaches are actually always limited. Through experimental analysis, we find the reason behind such phenomenon lies in that the detector can neither discern nor rectify the misclassified pseudo bounding boxes predicted by the detector itself (dubbed self-errors). To remedy such inherent limitations, we propose the CrossRectify method, where two detectors with the same structure but different initial parameters are simultaneously trained. During training, the pseudo bounding boxes predicted by each detector are sent to the other detector for a further judgement, and the self-errors can be realized and rectified in such way. Our CrossRectify method is reasonable and robust, based on our two consistent observations across various datasets and detector structures. First, we can always observe that the pseudo bounding boxes with high scores are more likely to be correctly classified than ones with low scores, so the self-errors can be realized and rectified by the prediction disagreements. Second, we surprisingly observe that even if the rectified pseudo labels are still wrong, the non-self-errors can still mitigate the performance degrade compared with the self-errors, which shows the robustness of CrossRectify. Some may argue the complexity of CrossRectify since we need to train two detectors, however we show by experiments that two final predictions can be directly merged to further obtain a $\sim 1\%$ AP gain and achieve a new state-of-the-art performance. We believe CrossRectify can enlighten the entire semi-supervised learning field, such as image classification and semantic segmentation, which leaves as future works.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data will be made available on request.

Acknowledgment

This work was supported by the National Natural Science Foundation of China 61832016, U20B2070, 6210070958, 62102162.

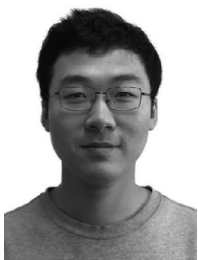
Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2022.109280.

References

- [1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
- [2] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, *Adv. Neural Inf. Process. Syst.* 28 (2015) 91–99.
- [3] B. Bosquet, M. Mucientes, V.M. Brea, STDnet-ST: spatio-temporal convnet for small object detection, *Pattern Recognit.* 116 (2021) 107929.
- [4] H. Wang, Q. Wang, P. Li, W. Zuo, Multi-scale structural kernel representation for object detection, *Pattern Recognit.* 110 (2021) 107593.
- [5] Y. Kong, M. Feng, X. Li, H. Lu, X. Liu, B. Yin, Spatial context-aware network for salient object detection, *Pattern Recognit.* 114 (2021) 107867.
- [6] J. Zhang, H. Su, W. Zou, X. Gong, Z. Zhang, F. Shen, CADN: a weakly supervised learning-based category-aware object detection network for surface defect detection, *Pattern Recognit.* 109 (2021) 107571.
- [7] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, et al., The open images dataset V4, *Int. J. Comput. Vis.* 128 (7) (2020) 1956–1981.
- [8] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05)-Volume 1-Volume 01, 2005, pp. 29–36.
- [9] Y.M. Asano, C. Rupprecht, A. Vedaldi, Self-labelling via simultaneous clustering and representation learning, in: International Conference on Learning Representations (ICLR), 2020.
- [10] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.
- [11] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, *Int. J. Comput. Vis.* 88 (2) (2010) 303–338.
- [12] S. Qiao, W. Shen, Z. Zhang, B. Wang, A. Yuille, Deep co-training for semi-supervised image recognition, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 135–152.
- [13] Z. Ke, D. Wang, Q. Yan, J. Ren, R.W. Lau, Dual student: breaking the limits of the teacher in semi-supervised learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6728–6736.
- [14] H. Wei, L. Feng, X. Chen, B. An, Combating noisy labels by agreement: a joint training method with co-regularization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13726–13735.
- [15] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, M. Sugiyama, How does disagreement help generalization against label corruption? in: International Conference on Machine Learning, PMLR, 2019, pp. 7164–7173.
- [16] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, T. Pfister, A simple semi-supervised learning framework for object detection, *arXiv preprint arXiv:2005.04757* (2020).
- [17] Z. Wang, Y. Li, Y. Guo, L. Fang, S. Wang, Data-uncertainty guided multi-phase learning for semi-supervised object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4568–4577.
- [18] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, P. Vajda, Unbiased teacher for semi-supervised object detection, in: International Conference on Learning Representations, 2020.

- [19] Y. Tang, W. Chen, Y. Luo, Y. Zhang, Humble teachers teach better students for semi-supervised object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3132–3141.
- [20] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, Z. Liu, End-to-end semi-supervised object detection with soft teacher, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3060–3069.
- [21] A. Tarvainen, H. Valpola, Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 1195–1204.
- [22] Q. Zhou, C. Yu, Z. Wang, Q. Qian, H. 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, 2021, pp. 4081–4090.
- [23] Z. Cai, N. Vasconcelos, Cascade R-CNN: delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162.
- [24] C.R. Qi, O. Litany, K. He, L.J. Guibas, Deep hough voting for 3D object detection in point clouds, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9277–9286.
- [25] J. Jeong, S. Lee, J. Kim, N. Kwak, Consistency-based semi-supervised learning for object detection, Adv. Neural Inf. Process. Syst. 32 (2019) 10759–10768.
- [26] J. Jeong, V. Verma, M. Hyun, J. Kannala, N. Kwak, Interpolation-based semi-supervised learning for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11602–11611.
- [27] P. Tang, C. Ramaiah, Y. Wang, R. Xu, C. Xiong, Proposal learning for semi-supervised object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 2291–2301.
- [28] X. Chen, Y. Yuan, G. Zeng, J. Wang, Semi-supervised semantic segmentation with cross pseudo supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2613–2622.
- [29] R. Soloviyev, W. Wang, T. Gabruseva, Weighted boxes fusion: ensembling boxes from different object detection models, Image Vis. Comput. 107 (2021) 104117.
- [30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
- [31] H. Wang, Y. Cong, O. Litany, Y. Gao, L.J. Guibas, 3DloUMatch: leveraging iou prediction for semi-supervised 3D object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14615–14624.
- [32] S. Song, S.P. Lichtenberg, J. Xiao, SUN RGB-D: a RGB-D scene understanding benchmark suite, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 567–576.
- [33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
- [34] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017).
- [35] Z. Wang, Y.-L. Li, Y. Guo, S. Wang, Combating noise: semi-supervised learning by region uncertainty quantification, Adv. Neural Inf. Process. Syst. 34 (2021).
- [36] N. Zhao, T.-S. Chua, G.H. Lee, SESS: self-ensembling semi-supervised 3D object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11079–11087.



Chengcheng Ma received the BSc degrees in automation from Northwestern Polytechnical University, China in 2017. Currently, he is a PhD candidate in the Institute of Automation, Chinese Academy of Sciences, China. His research interests include semi-supervised learning and adversarial learning.



Xingjia Pan received the BSc degree in automation and finance from Nankai University in 2015 and received PhD from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences in 2020. His research interests include computer graphics, computer vision and machine learning.



Qixiang Ye received the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2006. He has been a professor with the University of Chinese Academy of Sciences since 2009, and was a visiting assistant professor with the Institute of Advanced Computer Studies (UMIACS), University of Maryland, College Park until 2013.



Fan Tang is an Assistant Professor in School of Artificial Intelligence, Jilin University. He received the PhD degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2019. His research interests include computer graphics, computer vision, and machine learning.



Weiming Dong is a Professor in the Sino-European Lab in Computer Science, Automation and Applied Mathematics (LIAMA) and National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences. His research interests include image synthesis and image recognition. He is a member of the ACM and IEEE.



Changsheng Xu is a Professor in National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences and Executive Director of China-Singapore Institute of Digital Media. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops. He is IEEE Fellow, IAPR Fellow and ACM Distinguished Scientist.