Low-Confidence Samples Mining for Semi-supervised Object Detection

Guandu Liu\textsuperscript{1,2}, Fangyuan Zhang\textsuperscript{1,2}, Tianxiang Pan\textsuperscript{1,2}, Jun-Hai Yong\textsuperscript{1,2} and Bin Wang\textsuperscript{1,2*}

\textsuperscript{1}School of Software, Tsinghua University, China
\textsuperscript{2}Beijing National Research Center for Information Science and Technology (BNRist), China
{\{liugd21, zhangfy19\}@ mails.tsinghua.edu.cn, ptx9363@gmail.com, \{yongjh, wangbins\}@tsinghua.edu.cn

Abstract

Reliable pseudo-labels from unlabeled data play a key role in semi-supervised object detection (SSOD). However, the state-of-the-art SSOD methods all rely on pseudo-labels with high confidence, which ignore valuable pseudo-labels with lower confidence. Additionally, the insufficient excavation for unlabeled data results in an excessively low recall rate thus hurting the network training. In this paper, we propose a novel Low-confidence Samples Mining (LSM) method to utilize low-confidence pseudo-labels efficiently. Specifically, we develop an additional pseudo information mining (PIM) branch on account of low-resolution feature maps to extract reliable large-area instances, the IoUs of which are higher than small-area ones. Owing to the complementary predictions between PIM and the main branch, we further design self-distillation (SD) to compensate for both in a mutually-learning manner. Meanwhile, the extensibility of the above approaches enables our LSM to apply to Faster-RCNN and Deformable-DETR respectively. On the MS-COCO benchmark, our method achieves 3.54% mAP improvement over state-of-the-art methods under 5% labeling ratios.

1 Introduction

Deep neural networks [Liu et al., 2017; Kim and Lee, 2020] have achieved remarkable progress in the area of object detection. As model complexity increases, a large amount of precisely annotated data is required to train deep networks. To address this need, large-scale object datasets such as MS-COCO [Lin et al., 2014] and Objects365 [Shao et al., 2019] have been proposed in the community. Nevertheless, the process of annotation can be prohibitively expensive for real-world applications.

Recently, semi-supervised object detection (SSOD) has gained attention in the computer vision community [Liu et al., 2021; Zhou et al., 2021; Xu et al., 2021], as it only requires a small amount of annotated data. Most SSOD approaches follow the mean teacher paradigm [Tarvainen and Valpola, 2017], which trains a teacher and student model in a mutually beneficial manner. Pioneering approaches such as UBteacher [Liu et al., 2021] and its variants [Zhang et al., 2022; Chen et al., 2022a] improve detector performance from the perspective of classificatory balance. To further boost performance, SoftTeacher [Xu et al., 2021] has been proposed, but these approaches do not consider the detailed distribution of pseudo-labels, which contain the box and category. Recalling previous works, it is apparent that there exists a significant gap in detection performance for boxes of different scales. This observation motivates the following question: Whether there exist differences in the confidence distribution of pseudo-labels between boxes of different scales?

To address this question, we conducted extensive experiments to explore the relationship between the confidence distribution and the scale of pseudo-labels. Initially, we trained a vanilla Faster-RCNN detector [Ren et al., 2015] using all the labeled COCO training data. Subsequently, we generated pseudo-labels for the COCO validation set images. Following the methodology of prior works [He et al., 2017; Cheng et al., 2020], we computed the IoUs (Intersection-over-Union) between the boxes in pseudo-labels and ground-truths with the same category, and we utilized these values to assess the quality of the pseudo-labels. Figure 1 illustrates the IoU distribution for pseudo-labels of different scales. Figure 1(a) shows that the pseudo-label quality is positively associated with the box’s scale. As the pseudo-
label boxes increase in size, there are more pseudo-labels with high quality (Figure 1(b)), which will provide more precise box and category information for SSOD. Figure 2 visually depicts the pseudo-labels at various scales to qualitatively compare the differences in the confidence distribution of generated pseudo-labels with different scales. We observe that the pseudo-label of bus in Figure 2(b) (i.e., the large pink box with a classification score of 0.72) is accurate, while the pseudo-label of toilet with a higher score but smaller area (the blue box surrounded by red dotted lines in Figure 2(a)) is incorrect. Intuitively, under the same confidence score, the detector tends to make more accurate predictions for larger-area samples than small-area ones. Hence, leveraging the scale information to exploit low-confidence pseudo-boxes adequately is a valuable technique.

Based on these observations, we have designed a novel training procedure for semi-supervised object detection (SSOD) called Low-confidence Sample Mining (LSM). The direct approach of adding large-area pseudo-labels with low confidence has shown limited improvement (as discussed in Section 5.1). Therefore, we propose leveraging low-resolution feature maps to learn reliable large-area candidate boxes, which is more suitable for large-area object training [Singh et al., 2018; Li et al., 2019]. Specifically, LSM introduces an additional branch called pseudo information mining (PIM) for self-learning low-confidence pseudo-labels. PIM downsamples the original image through a feature pyramid network (FPN) to obtain lower resolution feature maps. A lower threshold is set as DDT [Zheng et al., 2022] to allow more pseudo-labels to participate in PIM training and help dig hidden credible low-confidence samples. Since scale information PIM uses can be produced in both Faster-RCNN and Deformable-DETR (DDETR) [Zhu et al., 2021], it is natural to introduce DDETR into SSOD. During the joint training process of the main and PIM branches, we have observed that the candidate boxes learned by both branches have certain complementarity (see Section 5.3). To achieve mutual learning between these two branches, we introduce a self-distillation (SD) module. SD uses the prediction of PIM for low-confidence candidate boxes to supervise the main branch training, and calculates $KL$ divergence loss between classification predictions from the main and PIM branches.

Under the same setting as mean teacher framework [Liu et al., 2021; Xu et al., 2021], our method surpasses the previous state-of-the-arts by significant margins. Especially in the only 5% labeled MS-COCO [Lin et al., 2014], LSM achieves 3.54% mAP improvement over state-of-the-arts. Furthermore, we find that mean teacher paradigm performances are only 54% mAP improvement over state-of-the-arts. Further, we observe the positive correlation between pseudo-labels and IoUs in SSOD and inspire the use of clean low-confidence boxes from a scale perspective. These observations provide a new direction to improve SSOD.

- Based on the above observations, we propose LSM, which uses PIM and SD to exploit clean low-confidence pseudo-labels from low-resolution feature maps efficiently. Extensive experiments are also performed on both Faster-RCNN and DDETR, which demonstrates that LSM does not rely on specific model components.

- We introduce DDETR into SSOD and use ImageNet [Deng et al., 2009] as unlabeled data to conduct the cross-domain task, which indicates the excellent denoise capability of LSM.

Figure 2: Illustration of pseudo-labels with boundary box, category and classification score. The incorrect pseudo-label (the toilet box in bottom) in (a) has a higher classification score than the correct pseudo-label (bus box) in (b).

2 Related Work

2.1 Semi-Supervised Learning

Semi-supervised learning constitutes a fundamental research area within the domain of deep learning. The most prevalent semi-supervised learning approaches are realized through consistency regularization [Izmailov et al., 2018; Sajjadi et al., 2016; Kim et al., 2022; Xie et al., 2020a] and pseudo-labeling. Pseudo-labeling [Sohn et al., 2020a; Berthelot et al., 2019; Xie et al., 2020b] involves appending predictions to the unlabeled data during the training process, utilizing a teacher model to assist in this task. Consequently, high-confidence predictions are selected as supervisory signals, which serve to enhance model training effectively.

2.2 Semi-Supervised Object Detection

The methodologies of semi-supervised object detection (SSOD) primarily stem from semi-supervised learning approaches. STAC [Sohn et al., 2020b] is the first to apply pseudo-labeling and consistency learning to SSOD. It generates pseudo-labels for unlabeled data utilizing a pretrained model, and subsequently trains a student model with strongly augmented unlabeled images. The mean teacher paradigm [Liu et al., 2021; Zhou et al., 2021; Xu et al., 2021] maintains a teacher model for online pseudo-labeling, acquiring reliable pseudo-labels for student model training through a high threshold. However, due to the empirical nature of
this threshold, numerous dependable supervisory signals are discarded. Dynamic threshold strategies [Li et al., 2022b] seek to obtain a higher quantity of high-confidence supervisory samples by employing a variable threshold.

However, none of the above methods consider the hidden available low-confidence samples. Based on this, recent methods have made efforts in low-confidence samples learning. [Zheng et al., 2022] equips vanilla detector framework with the bypass head to learn pseudo-labels with a lower threshold. [Wang et al., 2022] takes the sum of Top-K probability predictions as the selection basis to expand learning samples. Nevertheless, they all lack further mining that refer to credible information in low-confidence samples and similarly take the incorrect pseudo-labels into training, e.g., the mistaken toilet box in Figure 2(a) will be retained in the above-both methods. Furthermore, the object detector based on transformer has shown powerful performance in recent years. [Carion et al., 2020; Zhu et al., 2021] utilize the attention mechanism to get a larger receptive field on the feature map and apply bipartite graph matching to implement end-to-end training. Limited by mean teacher framework, many SSOD methods [Chen et al., 2022a; Chen et al., 2022b] cannot be directly applied to transformer structure. This hinders the application of Deformable-DETR (DDETR) in SSOD. Owing to the multi-scale feature maps LSM used can be produced by both DDETR and Faster-RCNN. Our work can be applied in DDETR effortlessly.

2.3 Multi-Scale Invariant Learning

Multi-scale invariant learning plays a vital role in object detection (OD) by facilitating the learning of objects across different scales. [Singh et al., 2018] accelerates multi-scale training by sampling low-resolution chips from a multi-scale image pyramid. [Li et al., 2019] employs convolutions with three distinct dilation rates to extract features from objects of varying sizes. Both methods demonstrate remarkable performance in multi-scale learning. Inspired by multi-scale training, our proposed PIM utilizes downsampling and a feature pyramid network (FPN) to generate lower-resolution feature maps for learning large-area objects.

In fact, [Li et al., 2022a] and [Guo et al., 2022] incorporate multi-scale label consistency into the mean teacher framework, striving to learn consistent representations across diverse scales. Although these approaches feature a branch for aligning dense features at different scales, which assists in mining scale-equivariant background features, they still rely on high-confidence pseudo-labels as the training target between the two branches. While our LSM method will utilize supplementary foreground proposals from low-confidence pseudo-labels.

3 Methodology

3.1 Problem Definition

Semi-supervised object detection aims to use a large amount of unlabeled data to improve model performance, where a small labeled dataset \( D^l = \{ I^l_i, Y^l_i \}_{i=1}^{N^l} \) and a large unlabeled dataset \( D^u = \{ I^u_i \}_{i=1}^{N^u} \) are available. \( N^l, N^u \) presents the number of labeled, unlabeled data. \( Y_i \) contains object information of image \( I_i \), including bounding box \( \{ x_j, y_j, w_j, h_j \}_{j=1}^{N_i} \) and category \( \{ c_j \}_{j=1}^{N_i} \). \( N_i \) presents the number of objects in the \( i \)th picture.

3.2 Preliminary: Mean Teacher Framework

In the regime of SSOD, this study takes two-stage methods based on the mean teacher paradigm [Tarininen and Valpola, 2017] as the baseline. Following previous works, we first train the student model on labeled data and then copy the parameters of the student model to the teacher model. The student model accepts both labeled data and unlabeled data, of which supervision signals come from the teacher model's predictions. The loss function of SSOD can be summarized as supervised loss \( L_s \) and unsupervised loss \( L_u \),

\[
L = L_s + \lambda_u L_u, \\
L_s = L_{cls}(I^s, Y^s) + L_{reg}(I^s, Y^s), \\
L_u = L_{cls}(I^u, \hat{Y}^u) + L_{reg}(I^u, \hat{Y}^u).
\]

Among them, \( L_{cls} \) represents classification loss, and \( L_{reg} \) represents regression loss. \( \hat{Y}^u \) represents the pseudo-labels generated by the teacher model, and \( \lambda_u \) is the weight of unsupervised loss. During per iteration, the student model will update the teacher model with its own parameters in the way of exponential moving average (EMA) and generate cleaner pseudo-labels.

\[
\theta_s \leftarrow \theta_s + \frac{\partial L}{\partial \theta_s}, \\
\theta_t \leftarrow \lambda_u \theta_t + (1 - \lambda_u) \theta_s,
\]

where \( \theta_s, \theta_t \) represents the parameters of the student, teacher model, and \( \lambda_u \) represents the ratio of parameter updates.

3.3 Low-confidence Samples Mining (LSM)

In this section, we introduce the LSM method. The reliable latent low-confidence pseudo-labels are mined adequately through pseudo information mining (PIM) and self-distillation (SD). The confidence in this section later refers to the classification score.

Overview

We introduce the whole pipeline of our LSM method in Figure 3 and LSM is applied in the student model after burn-in under the mean teacher framework. First of all, the teacher model trained on a small amount of labeled data \( D^l \) is utilized to produce pseudo-labels for \( D^u \). Regarding the hybrid training of ground-truths and pseudo-labels, LSM consists of the main branch and pseudo information mining (PIM) for low-confidence pseudo-labels learning. As shown in the yellow line in Figure 3(b), the main branch receives the original feature maps group and leverages pseudo-labels with a high threshold \( t \). While the PIM receives downsampling feature maps to learn low-confidence pseudo-labels. The total loss of LSM can be formulated as follow,

\[
L_{total} = L_m + \lambda_u (L_m^u + L_p^u + L_{distill}),
\]

where \( L_m^u \) and \( L_p^u \) refer to supervised loss and unsupervised loss from the main branch, which are the same loss from the
mean teacher paradigm. While $\mathcal{L}_p^s$ represents unsupervised loss from PIM. $\mathcal{L}_{distill}$ means distillation loss under the self-distillation (SD) strategy.

**Pseudo Information Mining (PIM)**

Motivated by the positive correlation between the area and IoUs of pseudo labels observed in Figure 1, we aim to mine reliable pseudo-labels from a scale perspective. As shown in Section 5.1, directly incorporating large-area pseudo-labels (area > 96 x 96) with a lower-confidence threshold of 0.5 into training yields some improvement. However, utilizing an area threshold of (96 x 96) as an empirical parameter is inappropriate for defining large-area boxes. Since detectors tend to learn large objects from low-resolution feature maps, we are inspired to extract valuable large-area pseudo-labels from small-scale images. Consequently, we design PIM from a multi-scale standpoint to learn reliable large-area pseudo-labels with low confidence. The green line in Figure 3(b) depicts the forward procedure of PIM.

Initially, PIM downsamples the original image by a ratio of 0.5 (both width and height are reduced to half of their original sizes), and then inputs the downsampled image into the backbone. A Feature Pyramid Network (FPN) generates a series of small-scale feature maps from the downsampled image, referred to as downsampling feature maps. The $i$th downsampling feature map shares the same size as the $(i+1)$th original feature map. Subsequently, PIM establishes a lower threshold $\alpha$ to include more pseudo-labels in PIM training, fostering the extraction of diverse information by the detector. As downsampling feature maps possess lower resolution, the box features extracted from the detector in PIM are inclined to learn credible large-area candidate boxes. These box features from PIM are then fed into an auxiliary classifier and regression head to compute $\mathcal{L}_p^{cls}$ and $\mathcal{L}_p^{reg}$. PIM ensures that the detector learns valuable information from low-confidence pseudo-labels, while the detector’s bias towards large-area pseudo-labels on low-resolution feature maps mitigates the adverse impact of noisy small-area pseudo-labels. Moreover, we decrease the loss weight of PIM to further minimize the influence of noisy pseudo-labels.

To reduce the computational load during forward propagation, PIM reuses the original proposals from the main branch (purple line in Figure 3; more details about original information are provided in the supplementary material). Notably, PIM employs $P_2, P_3, P_4$ feature maps to align the $P_5, P_4, P_3$ feature maps from the main branch in a shift alignment manner. As a result, the main branch can share proposals generated by the Region Proposal Network (RPN) with PIM. The loss of PIM can be formulated as follows:

$$
\begin{align*}
\mathcal{L}_m^s &= \mathcal{L}_{m}^{cls}(F_{main}(E(I_m^s)), Y^s) + \mathcal{L}_{m}^{reg}(R(E(I_m^s)), Y^s) \\
\mathcal{L}_m^u &= \mathbb{I}_{[Y_{score} > t]}\mathcal{L}_{m}^{cls}(F_{main}(E(I_m^u)), \hat{Y}^u) + \mathcal{L}_{m}^{reg}(R(E(I_m^u)), \hat{Y}^u), \\
\mathcal{L}_p^s &= \mathbb{I}_{[Y_{score} > \alpha]}\mathcal{L}_{p}^{cls}(F_{aux}(E(I_p^s)), \hat{Y}^u) + \mathcal{L}_{p}^{reg}(R(E(I_p^s)), \hat{Y}^u). 
\end{align*}
$$

Among them, $t$ is the filtering threshold of the main branch. And $\alpha$ is the filtering threshold of PIM, which is lower than $t$. $E$ is the feature extractor, and $R$ is the regression head shared with the main branch and PIM. $F_{main}$ is the main classifier, while $F_{aux}$ is the auxiliary classifier. At the same time, $\mathcal{L}_p^s$ and $\mathcal{L}_m^u$ also indicate that the model is forced to learn consistent representations under different-scale features in score interval $[t, \infty]$ to enhance the robustness of the detector model. Eq. 6 indicates that PIM can acquire more valuable pseudo boxes from the low-confidence samples due to a lower threshold filtering strategy, especially in Section 4.3 we show that this method can improve recall rate well compared to previous state-of-the-art methods.

PIM has certain similarities with multi-scale label consistency (MLC) [Li et al., 2022a]. However, MLC aims to improve the robustness of the model via learning the same pseudo-labels from different-scale feature maps, which also
ignores low-confidence pseudo boxes. Thereby it interferes the further learning on pseudo-labels. The convincing experimental results are presented in Table 4.

Self Distillation (SD)

LSM processes the box predictions from two branches by feeding them into the main classifier and auxiliary classifier, respectively. Due to the complementary predictions observed from the auxiliary classifier, the detector employs self-distillation to incorporate the knowledge of low-confidence bounding boxes learned by the auxiliary classifier into the main classifier. Specifically, we generate categorical predictions using an auxiliary classifier for bounding boxes with confidences in the $[\alpha, 1]$ range. Then, we employ the categorical predictions generated by the main classifier, with confidences in the same interval, to fit the corresponding predictions produced by the auxiliary classifier. The distillation loss is expressed as:

$$L_{\text{distill}} = \mathbb{I}_{[\alpha < \hat{y}^\text{aux} < t]} \text{KL}(F_{\text{main}} E(I_m^u), F_{\text{aux}} E(I_p^u)).$$

(7)

$\text{KL}$ calculates the KL divergence between the output of the main and auxiliary classifiers. Threshold $t$ and $\alpha$ are the same as those set in Eq. 5 and Eq. 6. Regarding the categorical distribution of low-confidence candidate boxes, it is unsuitable to directly choose the category with maximum probability as the hard label due to noise interference. Moreover, some categories with high probability may also be potential labels for the candidate box. Therefore, SD aids the main branch in learning soft predictions from PIM. Additionally, although the auxiliary classifier learns external pseudo-labels, it is still affected by noisy labels. Leveraging the complementarity of dual classifiers, we do not detach the gradient for PIM in SD, allowing the main branch to supervise PIM in a mutually-learning manner. This approach not only ensures that the main classifier learns more pseudo-labels but also mitigates the impact of noisy labels on the auxiliary classifier.

LSM for Deformable-DETR (DDETR)

PIM, combined with SD, constitutes the LSM method. Since LSM does not rely on specific network components, it can be effectively transferred to DDETR. The only difference between Faster-RCNN and DDETR, both implemented with LSM, is the original proposals. The results of bipartite graph matching between box predictions from the main branch and pseudo-labels are reused in PIM, as depicted by the purple line in Figure 3(b). Due to memory limitations, we employ the STAC [Sohn et al., 2020b] and pretrain-finetune training strategies for DDETR in the SSOD setting. Specifically, in the first stage, both strategies generate pseudo-labels for unlabeled data using a pre-trained model. In the second stage, STAC trains DDETR with a combination of labeled data and high-confidence unlabeled data, while pretrain-finetune first trains DDETR with unlabeled data and then finetunes it on labeled data. Notably, LSM can be applied in the second stage of both strategies.

In detail, we feed two stacks of original feature maps and low-resolution feature maps into the encoder to obtain two sets of reference points. Then, the object queries conduct cross attention with the reference points from the two sets, respectively, generating two groups of predictions from the two branches. The predictions from the two branches and the pseudo-labels in the two threshold intervals compute $L_{\alpha}^m, L_{\alpha}^p$, respectively. Finally, the dual classifiers generate classification predictions to compute $L_{\text{distill}}$. During the inference phase, only the main branch is used for forward computation, and the PIM branch is discarded.

4 Experiment

4.1 Datasets

In this section, we carry out extensive experiments to validate the effectiveness of LSM on the MS-COCO [Lin et al., 2014], PASCAL VOC [Everingham et al., 2010], and ImageNet [Deng et al., 2009] benchmarks.

MS-COCO contains two training sets, the train2017 dataset with 118K labeled images and the unlabeled2017 dataset with 123K unlabeled images. Following previous methods, we conduct experiments under three settings: (1) COCO-standard: we sample 1%, 2%, 5%, and 10% of the images from train2017 as labeled data, while the rest are treated as unlabeled data. (2) COCO-additional: We use the full train2017 dataset as labeled data and the unlabeled2017 dataset as unlabeled data. (3) VOC: We use the VOC07-trainval as the labeled dataset and the VOC12-trainval as the unlabeled dataset. We evaluate the model on COCO-val2017 for (1)/(2) and VOC07-test for (3).

In addition to these three traditional settings, we find that previous SSOD methods have not conducted cross-domain experiments on a more noisy unlabeled dataset. To demonstrate the denoising capacity of LSM-equipped Deformable-DETR on cross-domain tasks under SSOD settings, we introduce a fourth experimental setting: (4) COCO-ImageNet: We use the full train2017 dataset as labeled data and randomly choose 20% of ImageNet as noisy unlabeled data. The pseudo-labels for unlabeled data are predicted by the Faster-RCNN trained on the train2017 dataset.

4.2 Implementation Details

To be fair, we use Faster-RCNN as our base object detector as same as previous studies [Liu et al., 2021; Xu et al., 2021]. The weights of the backbone are initialized with ImageNet pre-trained model. For the main branch, we set pseudo boxes filtering threshold $t$ to 0.7. While for LSM, which can have a higher tolerance for pseudo boxes, we set the threshold $\alpha$ to 0.5. In all training settings, each of our training batches follows previous correspond works. For COCO-standard, the entire training steps are 180,000, of which the first 20,000 steps are used to pre-train the student model with labeled images. For COCO-additional, pre-training steps are 90,000, and the whole training steps are 360,000. For COCO-ImageNet, it takes the same training steps as COCO-additional due to the data size is close. In our experiments, strong data augmentation involves random jittering, gaussian noise, crop, and weak data augmentation involves random resize and flip. Moreover, we follow the existing work [Liu et al., 2021; Xu et al., 2021] to set the above hyperparameters.
Table 1: Comparison with the state-of-the-arts from different percentages of labeled MS-COCO. The margins of error are reported under 5 different random seeds. Where “*” means the corresponding result is not available.

Table 2: Comparison with the state-of-the-arts on VOC.

Table 3: Comparison with Multi-scale Label Consistency (MLC). “∆” indicates that the model is trained using MLC.

Figure 4: Comparison of the average recall and average precision curves for UBTeacher and LSM(ours) on 5% COCO-standard.

4.3 Results

COCO-standard

We first evaluate our method under the COCO-standard setting. As shown in Table 1, UBTeacher [Liu et al., 2021] equipped with LSM can perform better than previous work. When trained on 5% COCO-standard, LSM outperforms the UBTeacher by 3.54% mAP. Even if LSM is applied to SoftTeacher (PseCo), it can also improve by 2.71% (1.47%) mAP on average in 1%, 5% and 10% labeled data. We attribute the success of model performance to the stronger ability to capture object boxes in LSM. As shown in Figure 4(a), under the setting of 5% COCO-standard, the recall rate of LSM is higher than that of the UBTeacher in the whole training stage, which means that after adding LSM, the model can better detect the previous missing boxes. Figure 4(b) indicates that the mAP metric of LSM is better than that of UBTeacher. Theoretically, LSM adds an extra branch to learn low-confidence pseudo-labels, and the downsampling operation biases the model to learn more clean large-area pseudo boxes thus extracting additional information.

COCO-additional

In this section, we verify LSM can be further improved when trained on large-scale labeled data with additional unlabeled data. As shown in Table 1, when LSM is applied to the
UBTeacher, it can improve 1.93% mAP compared to the UBTeacher baseline. While LSM also achieves 1.2% mAP improvement with applying in SoftTeacher baseline. These results indicate that our method achieves satisfactory improvement on large-scale unlabeled datasets.

**VOC**

We evaluate models on a balanced dataset VOC to demonstrate the generalization of LSM. Table 2 provides the mAP results of CSD, STAC, UBTeacher, Humble Teacher and our LSM-equipped UBTeacher. Our method achieves 4.21% mAP improvement compared with UBTeacher baseline and 1.86% mAP improvement compared with Humble Teacher. Our method surpasses the other state-of-the-art results with a large margin. These results demonstrate that LSM can improve the existing SSOD consistently in various datasets.

**COCO-ImageNet**

To verify the effectiveness of LSM-equipped DDETR, we propose a new cross-domain setting: COCO-ImageNet. Considering that DDETR converges slowly, we use epochs as the unit in the training process.

As shown in Table 3 row 2, the detector reduces 1.73% mAP compared to the fully supervised mode in UBTeacher paradigm. While LSM demonstrates excellent denoise capability, with an improvement of 1.4% mAP compared to the UBTeacher baseline. Furthermore, in the training mode of pretrain-finetune, we find that DDETR performs better (0.7% mAP) than the supervised baseline, which indicates that the pretrain-finetune mode can better utilize more noisy pseudo-labels. Moreover, after applying LSM to the pre-training stage of DDETR, we observe that the model can achieve a 1.3% mAP improvement. This shows that LSM not only can be applied to Faster-RCNN and DDETR as a decoupling method but also has excellent learning ability in noisy labels.

**Compared with Multi-scale Label Consistency (MLC)**

The downsampling method used by our PIM follows the multi-scale label consistency method. MLC is widely used in object detection as an incremental method. However, existing methods force the model to learn a consistent representation of high-confidence pseudo-labels between the two branches. PIM, on the other hand, equips the downsampling branch with a lower filtering threshold, to capture more information from pseudo-labels. To verify that our PIM outperforms the MLC method, we apply these two methods under two settings of 5% COCO-standard and COCO-ImageNet, respectively.

As shown in Table 4, under the setting of 5% COCO, applying the MLC on UBTeacher can improve 1.79% mAP, while applying PIM can improve 3.54% mAP. In the COCO-ImageNet setting, we find that applying MLC on STAC brings a limited improvement (0.3% mAP) while applying the PIM can bring 1.4% mAP improvement.

### 5 Ablation Study

#### 5.1 Effects of Pseudo Information Mining Branch

PIM uses downsampling method to obtain three different-resolution feature maps of $P^d_2$, $P^d_3$, and $P^d_4$ generated by feature pyramid network (FPN). As shown in Table 5, we select multiple combinations from three feature maps to learn low-confidence samples. From row 2, the baseline has a certain improvement (0.76% mAP) through directly adding large-area pseudo-labels exceeding a lower threshold to the training. Whereas we find that using $P^d_2$, $P^d_3$, and $P^d_4$ simultaneously in the PIM, the model performs the best, 3.2% mAP higher than the UBTeacher baseline in row 6. As shown in row 3, if we only use the $P^d_2$, $P^d_3$, we find that the extra object information learned by the PIM is very limited, which is only 0.4% mAP higher than the baseline. When we add the lower resolution feature map $P^d_4$ (as shown in row 4), we find that the performance will be significantly improved, which is 2.41% mAP higher than the baseline. Through the comparison of the row 3 and the row 5 of Table 5, we can find that the combined detection of $P^d_2$ and $P^d_3$ on large objects is 3.54% mAP higher than that of $P^d_2$ and $P^d_3$. This shows that using lower resolution feature maps for PIM can indeed better mine large objects with lower confidence.

<table>
<thead>
<tr>
<th>$P^d_2$</th>
<th>$P^d_3$</th>
<th>$P^d_4$</th>
<th>$AP_{50-95}$</th>
<th>$AP_5$</th>
<th>$AP_{35}$</th>
<th>$AP_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>20.75</td>
<td>9.21</td>
<td>21.73</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>21.41</td>
<td>9.23</td>
<td>22.45</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>21.15 (+0.4)</td>
<td>9.10</td>
<td>22.63</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>23.16 (+2.41)</td>
<td>11.36</td>
<td>25.73</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>23.24 (+2.49)</td>
<td>11.48</td>
<td>25.88</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>23.95 (+3.2)</td>
<td>11.90</td>
<td>26.83</td>
</tr>
</tbody>
</table>

Table 5: Ablation study on PIM under 1% COCO-standard. The row 1 represents the UBTeacher baseline without using $P^d_2$, $P^d_3$, $P^d_4$. The row 2 indicates that large objects exceeding a lower threshold ($t > 0.5$, area > 96 × 96) are added to the training of UBTeacher baseline.

#### 5.2 Effects of Filter Threshold

Threshold plays a key role in screening high-quality pseudo-labels. Figure 5 shows the performance of the model under 1% COCO-standard at different thresholds. The red line represents the corresponding performance of the UBTeacher after adjusting the threshold $t$. The blue line shows the corresponding performance of the LSM-equipped UBTeacher after adjusting the threshold $\alpha$. For UBTeacher, it is difficult for the model to utilize the useful low-confidence pseudo-labels, and the performance of the model becomes worse as the threshold decreases. Moreover, it is difficult for the model to improve further after the threshold exceeds 0.6. For LSM, the performance of the model reaches the highest 23.95% mAP for $\alpha = 0.5$. Therefore, our method can make sufficient use of pseudo-labels with low confidence (i.e., $Y_u^{score} \in [0.5, 0.7]$), which is not achieved by previous methods.

#### 5.3 Effects of Self-distillation

In this experiment, the original PIM and the PIM equipped with SD are compared. Table 6 shows the performance gain on the UBTeacher (1% COCO-standard setting) and DDETR models respectively. As can be seen, PIM+SD can improve 0.2% (resp. 0.4%) mAP on UBTeacher (resp. DDETR) than PIM, which demonstrates the effectiveness of SD. As
the PIM branch has learned external low-confidence pseudo-labels, the PIM branch would be complementary to the main branch. This complementarity is illustrated with more visual results in the supplementary material.

<table>
<thead>
<tr>
<th>Method</th>
<th>UB</th>
<th>DDETR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP_{50/95}</td>
<td>34.55</td>
<td>34.75</td>
</tr>
</tbody>
</table>

Table 6: Ablation study on self-distillation.

6 Conclusion

In this study, we dive into the problem of discarding numerous low-confidence samples. Motivated by the positive correlation between area and IoUs of pseudo boxes, we propose the LSM method consists of PIM and SD. As high-level feature maps is conducive to learn large candidate boxes, PIM utilizes downsampling method and a lower threshold to extract diverse information from low-confidence pseudo-labels. Moreover, LSM takes advantage of SD to make PIM and main branch in mutually-learning manner. Sufficient experiments on benchmark demonstrate the superiority of our method. At the same time, our method can be freely applied to DETR framework, and shows excellent denoise ability on the cross-domain task.

Acknowledgments

This work was supported by the NSFC under Grant 62072271. Jun-Hai Yong was supported by the NSFC under Grant 62021002.

References


