arXiv:2307.12101v1 [cs.CV] 22 Jul 2023

Spatial Self-Distillation for Object Detection with Inaccurate Bounding Boxes

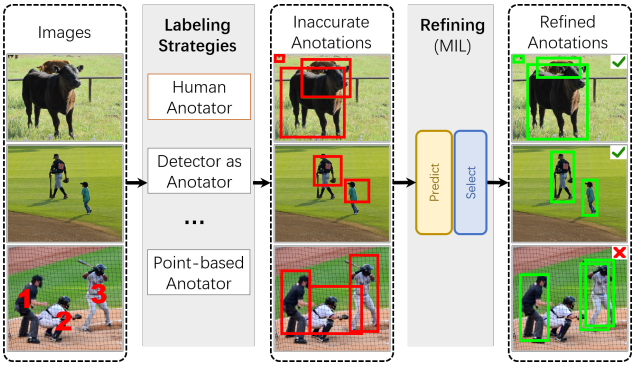
Di Wu*, Pengfei Chen*, Xuehui Yu*, Guorong Li, Zhenjun Han†, Jianbin Jiao
University of Chinese Academy of Sciences

Abstract

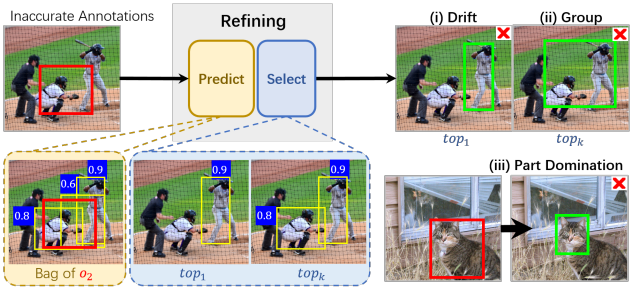
*Object detection via inaccurate bounding boxes supervision has boosted a broad interest due to the expensive high-quality annotation data or the occasional inevitability of low annotation quality (*e.g. *tiny objects). The previous works usually utilize multiple instance learning (MIL), which highly depends on category information, to select and refine a low-quality box. Those methods suffer from object drift, group prediction and part domination problems without exploring spatial information. In this paper, we heuristically propose a* ***Spatial Self-Distillation based Object Detector (SSD-Det)*** *to mine spatial information to refine the inaccurate box in a self-distillation fashion. SSD-Det utilizes a Spatial Position Self-Distillation* ***(SPSD)*** *module to exploit spatial information and an interactive structure to combine spatial information and category information, thus constructing a high-quality proposal bag. To further improve the selection procedure, a Spatial Identity Self-Distillation* ***(SISD)*** *module is introduced in SSD-Det to obtain spatial confidence to help select the best proposals. Experiments on MS-COCO and VOC datasets with noisy box annotation verify our method's effectiveness and achieve state-of-the-art performance. The code is available at* https://github.com/ucas-vg/PointTinyBenchmark/tree/SSD-Det.

1. Introduction

Object detection [18, 35, 50, 28, 49] relying on large-scale datasets like MS-COCO[29] has significantly progressed and achieved good performance. However, accurate bounding box annotations are expensive and challenging in natural contexts [12]. Especially in many professional scenarios, it is difficult to label accurate annotations without domain knowledge (*e.g.*, agricultural crop observation and medical image processing) [12, 1]. As shown in Fig. 1a in some complex datasets, the human annotators may also annotate inaccurate bounding boxes due to the inherent ambiguities[20] of objects. In addition, labelling with a detector or weak signal[11] (*e.g.*, point) is much cheaper but brings more inaccuracy. Therefore learning robust detectors with inaccurate bounding boxes[12, 4, 33, 45, 24] is a practical and meaningful task and has boosted a broad interest.



(a) Labelling strategies lead to inaccurate annotations (red box).



(b) Three problems during previous MIL refining. Because their selections depend solely on classification, (i,ii): the refined box (green box) **drifts** to another object or makes a **group prediction** (merging across multiple objects) due to neighbor disturbance. Yellow boxes are proposals in the middle person's MIL bag. (iii): Local **part** may be more discriminative than the entire object and will be predicted.

Figure 1: The sources of inaccurate box annotations and three problems caused by previous refinement methods.

To use the inaccurate annotations, most related methods [12, 11] refine the inaccurate annotations as Fig. 1a shows, and then train a detector head or re-train a detector with the refined box as the new supervision. There are two main steps during refining: 1) **Bag Construction**: For each object, obtaining some proposals around the inaccurate annotated bounding box to form the object-level proposal bag; 2) **Proposal Selection**: Selecting the top-k proposals with the highest classification confidence from each bag and then

* Equal contribution.
† Corresponding authors. (hanzhj@ucas.ac.cn)

weighting average them to obtain the refined box.

During the proposal selection, they usually utilize multiple instance learning (MIL) [10] supervised by category information to choose the proposals with high classification confidence from the constructed bags. However, they pay less [12] or no [11] attention to mining spatial information, leading to the following problems as shown in Fig. 1b: (1) **Object Drift**: For each object, some proposals in the constructed bag do not have a high IoU with the original object but with another nearby object. These proposals are not spatially adjacent to the original object but still have high classification confidence, as the rightmost (the yellow box) proposal of O_2 shows in the left-bottom corner of Fig. 1b. Only the category confidence is relied on for selecting proposals for O_2, and the rightmost proposal will be selected as the refined box. It means the refined box drifts to another object (Fig. 1b (i)), reducing the recall; (2) **Group Prediction:** Most works [7, 11, 48] select the top-k proposals by classification confidence and then weight average them as the refined box, causing the group prediction problem, as shown in Fig. 1b (ii); (3) **Part Domination**: The detector often focuses on the object's semantic region, which can statistically represent the category (*e.g.* the face). As shown in Fig. 1b (iii), the high classification confidence of the animal is in the discriminant part (the face) rather than the entire object as mentioned by [39, 36].

To address these problems, we propose a **Spatial Self-Distillation** based detector **(SSD-Det)** to integrate the spatial cues into the bounding box refinement. SSD-Det has two important components: the Spatial Position Self-Distillation **(SPSD)** module for the bag construction step and the Spatial Identity Self-Distillation **(SISD)** module for the proposal selection step. To construct high-quality proposal bags, SPSD utilizes a neighborhood sampler to generate a balanced and flexible initial proposal bag for each object and then trains a regressor with the supervision of the annotated inaccurate bounding boxes. Finally, high-quality proposal bags are constructed with proposals corrected by the regressor. The mechanism behind SPSD is that the network learns the spatial information from the reliable samples, *e.g.* those low-noise annotations, in the dataset and then guides the noisy samples to produce high-quality proposals, as shown in Fig. 2. In addition, to further combine the category information and the spatial information, an interactive structure is implemented by alternately using SPSD to mine spatial cues and MIL to utilize the category information. With SPSD and the interactive structure, a high-quality proposal bag can be constructed. Experiments on MS-COCO show that SPSD can significantly improve the mean/max IoU between objects and proposals (about 18/10 points, Fig. 5) in the constructed bag. Instead of selecting proposals by classification confidence, we have proposed the SISD module in the proposal selection step.

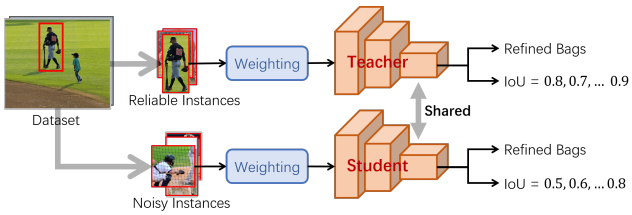


Figure 2: The mechanism of Spatial Self-Distillation. By assigning higher weight, the low-noise annotations can be seen as reliable samples to guide the training of proposals' spatial position and identity learning in SPSD and SISD.

We use it to obtain each proposal's spatial confidence by predicting the IoU with the object and combining the IoU with classification confidence to select the top-k proposals. It is worth mentioning that SISD is an object-related IoU predictor, which means that the predicted IoU may be different for the same proposal that appears in different objects' bags. Accordingly, it guarantees that SISD can better handle object drift and group prediction problems. Experiments on MS-COCO and VOC datasets verify the effectiveness of our method and bring state-of-the-art performance. The contributions are as follows:

1) We further investigate the inaccurate-box supervised object detection tasks and propose an end-to-end training SSD-Det that combines the spatial and the category information in an interactive fashion.

2) We utilize an SPSD module to generate higher-quality proposals sampling through statistic-guide spatial position distillation, raising the upper bound of the refinement.

3) To add spatial cues to classification confidence, we also introduce an SISD module to select a proposal belonging to the object rather than the category.

4) The performance of our proposed SSD-Det improves the mean average precision (AP) of the best previous method (*e.g.* over 10 AP on 40% noisy MS-COCO) and achieves state-of-the-art under various noise rate box supervision on MS-COCO and VOC datasets.

2. Related Work

2.1. Object Detection

Classic object detection [15, 35, 34, 30, 28, 3, 38, 49] is supervised by an accurate bounding-box. One-stage detectors utilize anchors as the sliding-window, such as YOLO [34], SSD [30], and RetinaNet [28]. Two-stage detectors mine spatial information to predict proposals (*e.g.* selective search [41] in Fast R-CNN [15] or RPN in Faster R-CNN [35]) and conduct classification and bounding-box regression with filtered proposals sparsely. Transformer-based (*i.e.* DETR [3], Deformable-DETR [54], and Swin-Transformer [31]) detectors utilize global information for better representation. Sparse R-CNN [38] combines a transformer's advantages and CNNs for detection.

2.2. Weakly-Supervised Object Detection (WSOD)

WSOD trains object detectors with image tag supervision. Only with the category annotation, the majority of previous methods treat each image as a bag and candidate proposals as instances. They follow the multiple instance learning (MIL) pipeline [1, 39, 40, 7, 42], which highly depends on category information. However, the MIL loss function leads to a non-convex optimization problem; thus, MIL solutions are usually stuck into the local minima. Context information [23, 44], spatial regularization [1, 9, 42], and optimization strategy [39, 42, 40] are proposed to address the problems. SPE [26] introduces Transformer into WSOD and uses attention to generate proposals. SD-LocNet [51] tackles the initialized noisy object locations in WSOD and proposes a self-directed localization network to identify the noisy object instances. [39, 40, 7] use the pseudo label for classification's iterative refinement. However, we use the pseudo box as a better self-distillation teacher. [46, 36] conduct regression to move the proposals, whereas we conduct regression to distill for better bag construction. In this work, we also formulate box correction as a MIL problem.

2.3. Learning with Noisy Annotations

Training CNNs under noisy labels has been an active research area. Previous research focuses on the classification task, and develops various techniques to deal with noisy labels, such as sample selection [17, 22] for training, label correction [37, 32], and robust loss functions [52, 14] against noisy labels. Recently, many efforts [4, 24, 33, 45, 11, 12] have been devoted to the object detection task. On the one hand, Simon *et al*. [4] first investigates the impact of different types of label noise on object detection. They propose a per-object co-teaching method to alleviate the effect of noisy labels. On the other hand, [45] proposes a meta-learning framework for noisy annotations consisting of noisy category labels and bounding boxes. [11, 12] utilize object-level MIL to refine the inaccurate box. OA-MIL [12] constructs proposal bags through label assignment in a discriminant style. P2BNet [11] originally conducts point-supervised object detection tasks. However, it can be seen as the box correction in its refinement stage. It uses hand-craft anchors to generate proposal bags. Our method inherits the generative style of P2BNet and conducts spatial distillation to mine spatial information.

2.4. Knowledge Distillation

Knowledge distillation (KD) [21] aims to learn compact and efficient student models guided by excellent teacher networks. It is first applied to object detection in [5], in which hint learning and KD are both used for multi-class object detection. Recently, many efforts [25, 43, 8, 16, 47] aim to mimic the feature. [53] shows that localization knowledge is more important and proposed a localization distillation method. We also transfer the spatial knowledge from reliable labeled instances to correct inaccurate bounding boxes (shown in Fig. 2) in a self-distillation manner.

3. Methodology

This work aims to learn a robust detector with inaccurate bounding boxes. Instead of training a detector with the original inaccurate bounding box, we follow most related works [11, 12] that design a branch to refine the inaccurate bounding box and then train the detector head or detector with the refined bounding box. The most important part is how to design a refining policy. We first design a two-stage basic box refiner (gray region in Fig. 3) as a naive solution that modified from [11]. Then, SPSD and SISD are proposed and added to further mine the spatial cues for box refinement, yielding SSD-Det. Therefore, the overall loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{Basic} + \alpha_1 \cdot \mathcal{L}_{SPSD} + \alpha_2 \cdot \mathcal{L}_{SISD} + \alpha_3 \cdot \mathcal{L}_{Det} \quad (1)$$

where α_1, α_2 and α_3 are set as 0.25, 0.25 and 4 respectively. $\mathcal{L}_{Det}$ denotes the loss of detector or detection head. During inference, only the detector or the detection head is used.

3.1. Basic Box Refiner

Motivated by [12] and WSOD [1], we design the basic box refiner (detailed structure figure is in supplementary) that leverages classification confidence to refine the inaccurate box annotation. Then the refine annotation is used to train a detection head or detector. Following [11], we design a two-stream structure as a MIL classifier to select the best proposal for box refinement.

Giving an image with inaccurate box annotation, for each object, $\mathcal{B}$ is a bag of proposals (bounding boxes) that are generated around its inaccurate annotation by a sampler policy (*e.g*., selective search[41], edge box[55], neighborhood sampler in Sec. 3.2). Meanwhile a feature map is extracted with a backbone network. And then through 7×7 RoIAlign [18] and two fully connected (fc) layers, features of proposal in $\mathcal{B}$ are extracted and denote as $\mathbf{F}$. The basic box refiner takes proposal bag $\mathcal{B} \in \mathbb{R}^{P\times 4}$ and features $\mathbf{F} \in \mathbb{R}^{P\times D}$ as inputs, where P, D are denoted as the number of proposals in $\mathcal{B}$, feature dimension respectively.

Following [11] and [1], as Eq. 2 described, we apply the classification branch f_{cls} to $\mathbf{F}$ yields $\mathbf{O}^{cls}$, which is then passed through the $softmax$ function over classification dimension K to obtain the score $\mathbf{S}^{cls} \in \mathbb{R}^{P\times K}$, where K represents the number of instance categories. Likewise, instance selection branch f_{cls} is applied to $\mathbf{F}$ to yield $\mathbf{O}^{ins}$, and instance score $\mathbf{S}^{ins}$ is obtained through $softmax$ function over P proposals. The proposal score $\mathbf{S}$ is obtained by computing the Hadamard product of the classification score and the instance score. The bag score $\widehat{\mathbf{S}}$ is obtained by the summating of the P proposal boxes' proposal scores.

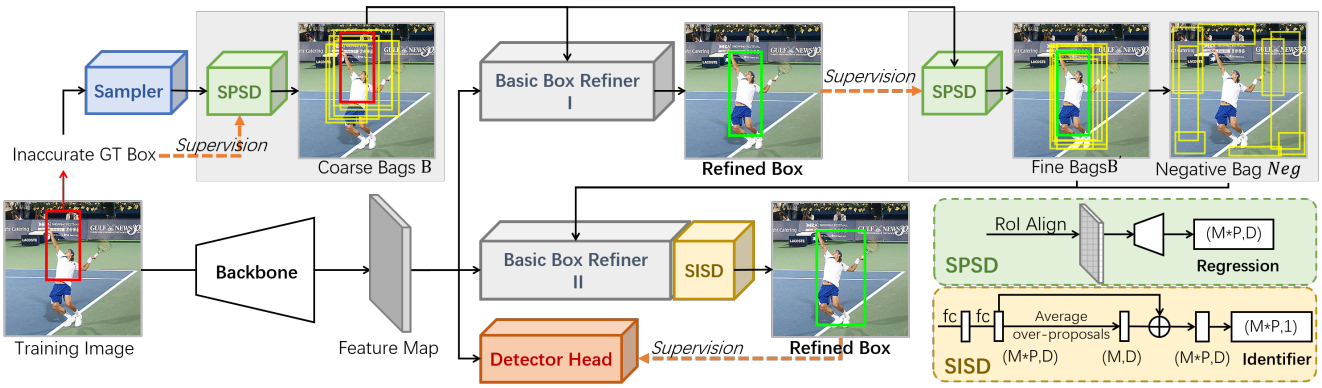


Figure 3: The framework of SSD-Det. It contains basic box refiner, SPSD module, SISD module and a detector head. Neighborhood sampler is adopted around the inaccurate annotation. Then, SPSD module generates better proposal bags which are fed into basic box refiner for MIL training. The selected proposals are average weighted as the refined box and supervise the next SPSD training. Meanwhile, the SISD module predicts the IoU between proposals and the object, and the estimated IoU is multiplied by classification score for better proposal selection to generate the refined box. SPSD shares backbone with the detector.

$$\mathbf{O}^{cls} = f_{cls}(\mathbf{F}) \in \mathbb{R}^{P\times K}; [\mathbf{S}^{cls}]_{pk} = e^{[\mathbf{O}^{cls}]_{pk}} / \sum_{k=1}^{K} e^{[\mathbf{O}^{cls}]_{pk}}.$$
$$\mathbf{O}^{ins} = f_{ins}(\mathbf{F}) \in \mathbb{R}^{P\times K}; [\mathbf{S}^{ins}]_{pk} = e^{[\mathbf{O}^{ins}]_{pk}} / \sum_{p=1}^{P} e^{[\mathbf{O}^{ins}]_{pk}}.$$
$$\mathbf{S} = \mathbf{S}^{cls} \odot \mathbf{S}^{ins} \in \mathbb{R}^{P\times K}; \widehat{\mathbf{S}} = \sum_{p=1}^{P} [\mathbf{S}]_p \in \mathbb{R}^{K}. \tag{2}$$

where $[\cdot]_{pk}$ is the value at row p and column k in the matrix.

The basic box refiner has two similar stages. The loss of stage I(termed $\mathcal{L}_I$) adopt the MIL paradigm with the form of cross-entropy (CE) loss, defined as:

$$\mathcal{L}_I = CE(\widehat{\mathbf{S}}, \mathbf{c}) = -\sum_{k=1}^{K} \mathbf{c}_k \log(\widehat{\mathbf{S}}_k) + (1-\mathbf{c}_k)\log(1-\widehat{\mathbf{S}}_k) \tag{3}$$

where $\mathbf{c} \in \{0,1\}^K$ is the one-hot category label. And each object's proposals with the top-k highest proposal score $\mathbf{S}$ are weighted to obtain the refined box.

The stage II takes the refined box of stage I as input and performs fine refining with a similar structure as stage I. Differently, the focal loss is adopted in stage II instead of cross entropy loss. In order to cooperate with focal loss, the classification branch uses the $sigmoid\ \sigma(x)$ instead of $softmax$ function and we sample some negative samples $\mathcal{N}$ to further suppress the background. With the bag score $\widehat{S}$ and the negative sample scores $\mathbf{S}^{cls}_{neg}$, the loss is:

$$\mathcal{L}_{II} = \left\langle \mathbf{c}^{\mathrm{T}}, \widehat{\mathbf{S}}^{*} \right\rangle \cdot \mathrm{FL}(\widehat{\mathbf{S}}, \mathbf{c}) + \sum_{\mathcal{N}} \beta \cdot \mathrm{FL}(\mathbf{S}^{cls}_{neg}, c_{neg}) \} \tag{4}$$

where FL is the focal loss [28], $\widehat{\mathbf{S}}^{*}_{j}$ represents the bag score predicted by stage I. $\left\langle \mathbf{c}^{\mathrm{T}}_{j}, \widehat{\mathbf{S}}^{*}_{j} \right\rangle$ represents the inner product of the two vectors, meaning the predicted bag score of the ground-truth category. β is the average of $\left\langle \mathbf{c}^{\mathrm{T}}_{j}, \widehat{\mathbf{S}}^{*}_{j} \right\rangle$. They are used to weight each object's FL for stable training. The overall loss function of the basic refiner here is:

$$\mathcal{L}_{Basic} = \mathcal{L}_I + \alpha_{II} \cdot \mathcal{L}_{II} \tag{5}$$

where α_{II} are the loss weights of the two stages.

During training, the refined box of stage II is used as supervision for a detection head or detector. After training, the basic box refiner will be removed, leaving a well-trained detection head or detector. In this way, we can train a detector under inaccurate annotations.

3.2. Spatial Position Self-Distillation (SPSD)

Like most MIL paradigm methods, basic box refiner has two main components: bag construction and proposal selection. And the main idea is to use classification information to guide the refining. In this paper, we add spatial information to improve refining. Specifically, SPSD is proposed to use spatial information to enhance bag construction.

Bag construction aims to obtain proposals for each object, while proposal selection is to select the proposals from the object bag. Then, the refined box is averaged over the selected proposals. Therefore, the quality of the proposals in constructed bag determines the upbound of refining. The bag construction can be implemented in a variety of ways. In this paper, the basic box refiner adopts a naive neighborhood sampler for bag construction. basic box refiner adopts a naive neighborhood sampler for bag construction.

Neighborhood Sampler. Proposals around the inaccurate box are sampled to construct an object bag. For each inaccurate box $b^* = (b^*_x, b^*_y, b^*_w, b^*_h)$, its scale and aspect ratio with s and v are adjusted and its positions o_x, o_y are jittered to obtain the diverse proposal $b = (b_x, b_y, b_w, b_h)$:

$$\begin{aligned} b_w &= v \cdot s \cdot b^*_w, \quad b_h = 1/v \cdot s \cdot b^*_h, \\ b_x &= b^*_x + b_w \cdot o_x, \quad b_y = b^*_y + b_h \cdot o_y. \end{aligned} \tag{6}$$

These proposals b are used to construct the positive proposal bag $\mathcal{B}$ to train the MIL classifier. Thanks to the hand-

craft sampling way, the number of proposals in different objects' proposal bags is controllable and balanced. However, the hand-craft neighborhood sampler strategy is difficult to set hyper-parameters, and the sampling space is discrete. For example, when the jitter region is small, the optimization space of refining is limited, while when it is large, more background will be introduced. Hence, we propose the SPSD module to mine spatial information for higher-quality proposal bag construction.

Statistically Guided Adaptive Sampling. Instead of simply using a neighborhood sampler, we adopt a statistically guided adaptive sampling by adding SPSD modules into the basic box refiner. Taking the constructed proposal bag $\mathcal{B}$ of hand-craft neighborhood sampler as input, the RoI features of proposals in $\mathcal{B}$ are extracted and fed into the two shared fc layers to obtain $\mathbf{F}$. Then a regression fc layer f_{dis}, supervised by the inaccurate annotated bounding box b^*, is introduced to predict the adaptive proposal bag $\mathcal{B}^{dis} = f_{dis}(\mathbf{F}) \in \mathbb{R}^{P\times 4}$, in which the proposals are closer to the object. Later, $\mathcal{B}^{dis}$ as the constructed proposal bag is fed into stage I of basic box refiner. In order to combine category and spatial information, we implement an interactive structure by alternately using SPSD to mine the spatial cues and using MIL in basic box refiner to utilize the category information. Specifically, the refined bounding box b^* of stage I that selected by the classification confidence is used to supervision of a new SPSD module for stage II. Similar as stage I, The new SPSD takes proposal bag $\mathcal{B}^{dis}$ of hand-craft neighborhood sampler as input. Through the RoI align and the two shared fc layers, the feature $\hat{\mathbf{F}}$ is extracted. An extra fc layer $f_{\hat{dis}}$ is then utilized to conduct further regression. Different with stage I, the obtained $\mathcal{B}^{\hat{dis}}$ is supervised by the refined $\hat{b^*}$. The loss function of the spatial distillation for adaptive sampling can be defined as $\mathcal{L}_{SPSD}$ in Eq. 7.

$$\mathcal{L}_{SPSD} = \frac{1}{P}\{\sum_{p=1}^{P}\mathbf{L_1}([\mathcal{B}^{dis}]_p, b^*) + \sum_{p=1}^{P}\mathbf{L_1}([\mathcal{B}^{\hat{dis}}]_p, \hat{b^*})\} \quad (7)$$

where the $\mathbf{L_1}$ is the L1 loss function for loose restrictions.

The idea behind SPSD is that the dataset with inaccurate annotation still has many reliable, high-quality boxes and inaccurate boxes. Supervised by the high-quality boxes statistically, the network can guide those proposals sampled around the inaccurate bounding box to regress to the ground truth. With the self-distillation mechanism, SPSD learns the semantic-spatial correspondence knowledge from the reliable samples in the dataset and then propagates the knowledge to produce high-quality proposals.

Adaptive Negative Sampling. Negative samples are introduced in Stage II to better suppress the background. With the sampled $\mathcal{B}^{\hat{dis}}$, we can adaptively sample the negative samples with a small IoU (set smaller than 0.3 by default) with all positive proposals in all bags, to compose the negative sample set $\mathcal{N}$ for stage II.

3.3. Spatial Identity Self-Distillation (SISD)

The basic box refiner selects the proposals only depending on classification confidence during proposal selection. To select the proposal which has high classification confidence and is also spatially close to the object from the bag, we propose a SISD module to predict the IoU between proposals and their corresponding object. Afterwards, through the combination between the IoU and the classification confidence, top-k proposals are selected. In SISD, we design an Object Relevance Enhancement (ORE) module to distinguish different objects' features with the same RoI region. And an identity predictor is designed to predict each proposal's IoU with the object.

Object Relevance Enhancement (ORE). ORE enhances object-relevant features, making SISD an object-relevant IoU predictor. ORE allows the predicted IoU to be different for the same proposal in other objects' bags. In addition, we integrate the feature of the bag's corresponding object into the proposal feature, making the feature of different bags' proposals distinct. That is the so-called ORE. For a proposal bag $\mathcal{B}$, the feature $\mathbf{F}$ is obtained through the RoI align and two fc layers. It is worth mentioning that the two fcs do not share the parameters with those in the refiner since the optimization goals are contradictory. To represent the feature of the bag's corresponding object, $\mathbf{F}^+ \in \mathbb{R}^{1\times D}$ is calculated by averaging features of P proposals in proposal bag $\mathcal{B}$. The object feature $\mathbf{F}^+$ is broadcast into $\mathbb{R}^{P\times D}$, and then added to the proposal features to obtain the object-relevant features $\mathbf{F}^* = \mathbf{F} + \mathbf{F}^+$.

Spatial Identity Prediction. By a following identity fc layer, $U \in \mathbb{R}^{P\times 1}$ is predicted. The pseudo label $T \in (0, 1)$ is IoU between proposals in $\mathcal{B}$ and the merged box $\hat{b^*}$ of stage I. For better optimization, the linear normalized $T' = (T - 0.5)/0.5 \in (-1, 1)$ is utilized as supervision. The object function of the identity predictor is identified in:

$$\mathcal{L}_{SISD} = smooth_{L1}(U, T') \quad (8)$$

where the $smooth_{L1}$ represents the smooth L1 loss. The predicted spatial confidence U' is obtained by normalizing the U. Finally, $S^* = U' \cdot S$ is used to select the top-k proposals for merging as the refined boxes.

4. Experiment

4.1. Experimental Settings

Datasets and Evaluation Metrics. For experimental comparisons, two publicly available datasets are used for object detection with inaccurate bounding boxes: MS-COCO [29] and PASCAL VOC 2007 [13]. **MS-COCO** (2017 version) has 118k training and 5k validation images with 80 common object categories. **PASCAL VOC** 2007 is one of the most popular benchmarks in generic object detection with 20 classes.

Method	Backbone	20% Box Noise Level						40% Box Noise Level					
		AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l
Val Set													
Clean-FasterRCNN [35]	ResNet-50	37.9	58.1	40.9	21.6	41.6	48.7	37.9	58.1	40.9	21.6	41.6	48.7
Clean-FasterRCNN [35]	ResNet-101	39.4	60.1	43.1	22.4	43.7	51.1	39.4	60.1	43.1	22.4	43.7	51.1
Clean-Retinanet [28]	ResNet-50	36.7	56.1	39.0	21.6	40.4	47.4	36.7	56.1	39.0	21.6	40.4	47.4
Noisy-FasterRCNN [35]	ResNet-50	30.4	54.3	31.4	17.4	33.9	38.7	10.3	28.9	3.3	5.7	11.8	15.1
Noisy-Retinanet [28]	ResNet-50	30.0	53.1	30.8	17.9	33.7	38.2	13.3	33.6	5.7	8.4	15.9	18.0
FreeAnchor[50]	ResNet-50	28.6	53.1	28.5	16.6	32.2	37.0	10.4	28.9	3.3	5.8	12.1	14.9
Co-teaching[17]	ResNet-50	30.5	54.9	30.5	17.3	34.0	39.1	11.5	31.4	4.2	6.4	13.1	16.4
SD-LocNet[51]	ResNet-50	30.0	54.5	30.3	17.5	33.6	38.7	11.3	30.3	4.3	6.0	12.7	16.6
KL loss[20]	ResNet-50	31.0	54.3	32.4	18.0	34.9	39.5	12.1	36.7	3.7	6.2	13.0	17.4
OA-MIL[12]	ResNet-50	32.1	55.3	33.2	18.1	35.8	41.6	18.6	42.6	12.9	9.2	19.9	26.5
SSD-Det	ResNet-50	33.6	57.3	35.3	19.5	37.2	43.3	27.6	53.9	26.0	16.0	31.0	34.9
SSD-Det	ResNet-101	34.3	57.6	36.7	19.1	38.1	44.3	28.4	54.3	27.2	16.5	31.9	36.4
SSD-Det+FR	ResNet-50	34.4	57.3	36.8	20.0	38.2	44.0	29.3	54.8	29.0	17.1	32.9	36.9
SSD-Det+FR	ResNet-101	36.2	59.1	39.2	20.9	40.2	47.1	30.6	56.7	30.7	18.1	34.5	39.0
Test Set													
Clean-FasterRCNN [35]	ResNet-50	37.7	58.7	40.8	21.7	40.6	46.7	37.7	58.7	40.8	21.7	40.6	46.7
Noisy-FasterRCNN [35]	ResNet-50	30.7	54.9	31.3	18.0	33.7	37.7	10.4	29.0	3.3	6.0	11.3	14.6
OA-MIL[12]	ResNet-50	32.3	55.8	33.7	18.5	35.0	40.2	18.5	42.3	12.8	9.3	19.1	25.1
SSD-Det	ResNet-50	33.5	57.3	35.5	19.1	36.0	41.9	28.0	54.1	26.5	16.5	30.0	34.5
SSD-Det+FR	ResNet-50	34.7	57.9	37.2	20.0	37.7	42.7	29.7	55.6	29.3	17.5	32.4	36.2

Table 1: Performance comparison on COCO. FR is Faster R-CNN. Clean-* and Noisy-* means original and noisy annotation.

Method	Backbone	Box Noise Level			
		10%	20%	30%	40%
Clean-FasterRCNN [35]	ResNet-50	77.2 *for clean*			
Clean-RetinaNet [28]	ResNet-50	73.5 *for clean*			
Noisy-FasterRCNN [35]	ResNet-50	76.3	71.2	60.1	42.5
Noisy-RetinaNet [28]	ResNet-50	71.5	67.6	57.9	45.0
KL loss[20]	ResNet-50	75.8	72.7	64.6	48.6
Co-teaching[17]	ResNet-50	75.4	70.6	60.9	43.7
SD-LocNet[51]	ResNet-50	75.7	71.5	60.8	43.9
FreeAnchor[50]	ResNet-50	73.0	67.5	56.2	41.6
OA-MIL[12]	ResNet-50	77.4	74.3	70.6	63.8
SSD-Det	ResNet-50	77.1	74.8	71.5	66.9

Table 2: Performance comparison on the VOC 2007 test set. The evaluation metric is AP_{50}. The Clean-* and Noisy-* means original annotation and noisy annotation.

Evaluation Metric. We use mean average precision mAP@[.5,.95] and (mAP@.5) for MS-COCO and VOC. The $\{AP, AP_{50}, AP_{75}, AP^{Small}, AP^{Middle}, AP^{Large}\}$ is reported for MS-COCO and AP_{50} for VOC.

Synthetic Noisy Dataset. Following [12], We simulate noisy bounding boxes by perturbing clean boxes from the original annotations. The details are in the appendix. We simulate various box noise levels ranging from 10% to 40% for the VOC and $\{20\%, 40\%\}$ for the MS-COCO.

Implementation Details. We implement our method on FasterRCNN [35] with ResNet50-FPN [19, 27] backbone, based on MMDetection [6]. Similar to the default setting of object detection on MS-COCO, the stochastic gradient descent [2] algorithm is used to optimize on 1x training schedule. The batch size is two images per GPU on 8 GPUS. For the VOC dataset, the batch size is two images per GPU on 2 GPUS. The performance we report is on a single scale (1333 * 800 for MS-COCO and 1000 * 600 for VOC).

4.2. Comparison with State-of-the-Art

We compare our method with several state-of-the-art approaches [20, 17, 51, 50, 12] on MS-COCO and VOC 2007 datasets. We denote Clean-FasterRCNN and Noisy-FasterRCNN as FasterRCNN models trained under clean (original annotations) and noisy annotations with the default setting, respectively.

MS-COCO Dataset. Table 1 shows the comparison results on the MS-COCO. Inaccurate bounding box annotations significantly deteriorate the vanilla Faster R-CNN's detection performance. Co-teaching and SD-LocNet only slightly improve the detection performance, especially under 40% box noise. That indicates that small-loss sample selection and sample weight assignment can not tackle noisy box annotations well. KL Loss slightly improves the performance under 20% and 40% box noise. By treating an object as a bag of instances, OA-MIL is somehow robust to noisy bounding boxes and performs better than other methods. Nevertheless, the previously-mentioned label assignment bag construction limits its ability to handle heavy noise. Our approach is more robust to noisy bounding boxes. It outperforms other methods by a large margin under high box noise levels and significantly boosts the detection performance across all metrics. For example, under 40% box noise, the end-to-end SSD-Det achieves 27.6 AP and 53.9 AP_{50}, 9.0 and attains 11.3 point improve-

2-Ref	SPSD	SISD	Re-Train	20% Box Noise Level								40% Box Noise Level							
				AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l	AP^{test}	AP^{test}_{75}	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l	AP^{test}	AP^{test}_{75}
				30.0	57.1	29.0	16.9	33.1	39.8	-	-	22.8	51.1	16.1	13.3	25.0	30.4	-	-
✓				31.2	56.7	31.6	17.8	34.5	41.0	31.4	32.0	24.6	52.0	20.1	14.3	28.2	31.9	25.0	20.5
✓	✓			33.0	56.9	34.8	18.7	35.5	42.2	33.1	34.8	27.2	53.7	24.7	15.9	30.3	**35.2**	27.6	25.6
✓	✓	✓		**33.6**	**57.3**	**35.3**	**19.5**	**37.2**	**43.3**	**33.5**	**35.5**	**27.6**	**53.9**	**26.0**	**16.0**	**31.0**	34.9	**28.0**	**26.5**
✓			✓	31.8	56.8	33.1	18.4	35.7	40.8	32.3	33.7	26.5	54.0	23.3	15.7	30.3	33.8	26.8	23.3
✓	✓		✓	34.1	**57.6**	36.4	19.0	37.7	43.8	34.3	36.6	29.0	**55.1**	27.8	17.0	32.5	36.7	29.3	28.4
✓	✓	✓	✓	**34.4**	57.3	**36.8**	**20.0**	**38.2**	**44.0**	**34.7**	**37.2**	**29.3**	54.8	**29.0**	**17.1**	**32.9**	**36.9**	**29.7**	**29.3**

Table 3: Modules ablation of SPSD, SISD and Re-Train on MS-COCO validation set (without) and test set (with test). The Re-Train means we generate the pseudo label by SSD-Det and re-train a Faster R-CNN detector.

Methods (w/o SISD)	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l
Neighborhood Sampler	24.6	52.0	20.1	14.3	28.2	31.9
SPSD (II) w/o weighted	26.0	53.3	22.5	15.6	29.4	33.4
SPSD (II) $w/$ weighted	26.3	53.4	22.5	15.6	29.3	33.8
SPSD (I+II) $w/$ weighted	**27.2**	**53.7**	**24.7**	**15.9**	**30.3**	**35.2**

Table 4: Different setting of SPSD.

ORE Strategies of SISD	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l
w/o SISD	27.2	53.7	24.7	15.9	30.3	**35.2**
SISD w/o ORE	27.3	53.3	25.7	16.9	30.2	35.0
+ subtract	27.2	53.8	24.6	15.7	30.4	35.0
+ concatenate	27.2	54.0	24.5	16.1	30.2	34.8
+ add	**27.6**	**53.9**	**26.0**	**16.0**	**31.0**	34.9
+ add $w/$ shared fcs	23.0	49.9	17.6	12.6	25.3	30.4

Table 5: Different ORE strategies of SISD.

Num.	0	1	2	3
AP/AP_{50}	24.6 / 52.0	26.3 / 53.4	**27.2 / 53.7**	27.0 / 53.1

Table 6: Number of SPSD module.

ment compared with state-of-the-art method OA-MIL, respectively. Also, through re-training on FasterRCNN, the performance further reaches 29.3 AP and 54.8 AP_{50}. With the backbone of ResNet-101, the performance achieves consistent improvement. On MS-COCO test set, our method also achieves state-of-the-art performance.

VOC 2007 Dataset. Table 2 shows the comparison results on the VOC 2007 test set. Co-teaching, SD-LocNet and KL Loss, can not address inaccurate bounding box annotations well. OA-MIL improves the performance on different noisy datasets. Our approach obtains further improvements to 77.10, 74.80, 71.50, 66.90 AP_{50} on 10%, 20%, 30 % and 40 % noisy box datasets, respectively.

4.3. Ablation Study and Analysis

To further analyze SSD-Det's effectiveness and robustness, we conduct more experiments on COCO val set if there are no other instructions. Except for Table 3, the noise level of these experiments is 40%.

Ablation of Modules. Ablation study of each component in our approach is given in Table 3, including: (i) Different stages of our basic box refiner. *i.e.* training object detector without the stage II (2-Ref), where the pseudo boxes predicted by the stage I are served as the supervision for training a parallel detector. (ii) SPSD, *i.e.* training without SPSD, where the object-bag is constructed directly

Methods	Box Refiner+Re-Train	SSD-Det	SSD-Det+Re-Train
AP/AP_{50}	29.0 / 54.4	27.6/53.9	**29.3 / 54.8**

Table 7: Comparisons of end-to-end and Re-Train.

Detectors	Clean-supervised		Noise-supervised			
					($w/$ ours)	
	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}
FasterRCNN	37.9	58.1	**10.3**	**28.9**	29.3	54.8
SparseRCNN [38]†	45.0	64.1	6.0	20.3	34.3	60.2
De-DETR [54]†	**46.8**	**66.3**	5.0	16.9	**35.2**	**60.9**

Table 8: Experiments on advanced detectors. De-DETR is Deformable DETR. † uses multi-scale data augment. '$w/$ ours' means using our method under noisy supervision.

by neighborhood sampling around the noisy ground-truth or the predicted pseudo boxes of the stage I. (iii) SISD. (iv) Re-Train with FasterRCNN (Re-Train).

Effectiveness of SPSD. SPSD further improves the detection performance on the MS-COCO, especially under high box noise levels, *e.g.* under 40% box noise level, SPSD boosts the performance from 24.6 to 27.2, as shown in Table 3 (row 3). In Table 4, we conduct further ablation on SPSD. With SPSD bag construction only in stage II, the performance increases by 1.4 AP. The performance further improves with the proposal score of stage I as weights. With SPSD in all stages, the AP reaches 27.2. Fig. 5 shows the bag quality. With SPSD, the mean IoU increases from 40.3 to 58.7 and the max and top-10 IoU increase to 78.3 and 75.1, which indicates a better upper bound of proposal selection. More high-quality proposals bring better optimization and easier proposals selection.

Number of SPSD. As shown in Table 6. When adding 3 SPSD, performance drops slightly, probably due to the accumulation of errors outweighing the performance gain from extra stages. Hence, 2 SPSD is our default setting.

Effectiveness of SISD. SISD is designed to select object-aware proposals in box selection. Under 40% and 20% box noise, the detection performance improves from 27.2 to 27.6 and 33.0 to 33.6, which verifies the effectiveness of the module, shown in Table 3. We also study the strategies of ORE in SISD (Table 5). The minus or concat on object feature $\mathbf{F}^+_j$ and proposal feature $\mathbf{F}_j$ do not work. With add strategy, the performance is 27.60. If SISD shares the two fc layers, the performance drops to 22.99 since the

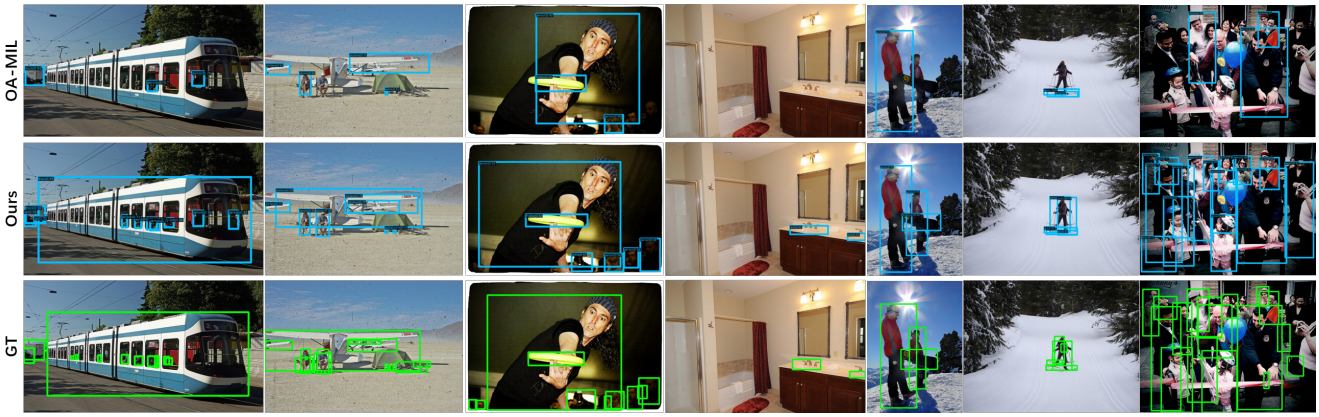


Figure 4: Qualitative detection results on COCO validation. Previous methods miss objects and face part prediction problems. Our method misses fewer objects, and the bounding box quality is better, especially for small or overlapped objects.

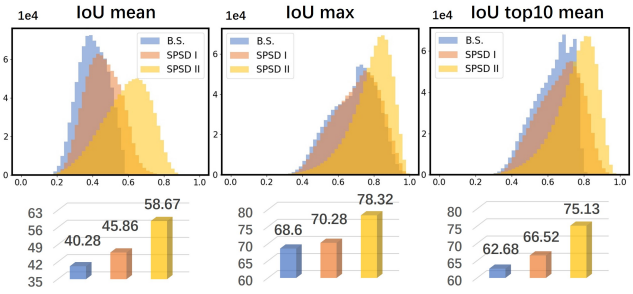


Figure 5: Bag quality (IoU of proposals with GT) of construction in SSD-Det. B.S. (blue) means neighborhood sampler. SPSD I (orange) denotes single SPSD adopted. SPSD II (yellow) is two SPSD and interactive structure adopted. SPSD II significantly improves the quality.

Methods	Drift Rate (%)				Group Rate (%)				Part Rate (%)			
	all	s	m	l	all	s	m	l	all	s	m	l
OA-MIL[12]	15.1	17.8	17.1	7.4	6.7	2.8	3.4	1.4	2.8	3.4	2.7	2.3
Ours	**1.5**	1.0	1.3	1.4	**1.7**	1.2	0.5	0.7	**1.0**	0.5	1.1	1.3

Table 9: Breakdown of different problems during refinement (COCO under 40% noise level). s, m and l mean small, middle and large scale.

optimization goals are contradictory (Identity distinguishes objects in the same category). If we directly use the RoI feature without ORE, the performance drops to 27.32 AP, verifying the effectiveness of the object relevance strategy.

Affect of Re-Train. As most WSOD methods do, we re-run the experiments by training a fully supervised detector for better performance. We find that if the SSD-Det only trains the refiner and uses the pseudo label to train the FasterRCNN, the result is good but lower than re-train after the end-to-end training given in Table 7 (row 1). This is because joint training is beneficial for box refinement.

Experiments on Advanced Detectors. We re-train recent detectors, *e.g.* SparseRCNN and Deformable DETR, under the boxes refined by our method. Table 8 verifies that our method achieves consistency improvement.

4.4. Visualization and Discussion.

Fig. 4 shows that OA-MIL faces missing instances and grouping instances issues for small or overlapped objects (as mentioned in [12]), while our method still works well. For a better intuitive understanding of SISD and SPSD, we visualize the bag construction quality in Fig. 5 Then, we makes noise types breakdown of 'Drift', 'Group' and 'part dominance' issues. We give the definition of IoU, IoG and IoD:

$$IoU = \frac{A(I)}{A(D) + A(G) - A(I)}, IoG = \frac{A(I)}{A(G)}, IoD = \frac{A(I)}{A(D)} \quad (9)$$

where A(*) is area of box *, D and G are refined box and gt box respectively, and I is insertion between D and G. We statistically count the proportion of three noise types of 'bad' refined boxes (having small IoU with gt) in Table 9: (i) Drift: 'bad' refined box has a higher IoU with another nearby object. (ii) Group: 'bad' refined box has high IoG with multiple objects. (iii) Part: 'bad' refined box has a high IoD. Table 9 shows quantitative results for each noise type of baseline and ours. The drift, group, part problems reduce from 15.1%, 6.7%, 2.8% to 1.5%, 1.7%, 1.0% , respectively, demonstrating our improvement.

5. Conclusion

This paper investigates problems during refinement caused by solely using category information to select proposals. We also propose SSD-Det to mine spatial information in a self-distillation fashion. SSD-Det introduces the SPSD module to learn semantic-spatial correspondence knowledge with neighborhood sampler and an interactive structure to combine spatial information and category information, thus producing a high-quality proposal bag. SISD in SSD-Det is utilized to improve the proposal selection procedure by integrating object-relevant spatial confidence. Complete ablations on multiple datasets verify the effectiveness of SSD-Det.

References

[1] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 1, 3

[2] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade - Second Edition*. Springer, 2012. 6

[3] Nicolas Carion, Francisco Massa, and Gabriel Synnaeve *et al*. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[4] Simon Chadwick and Paul Newman. Training object detectors with noisy data. In *IV*, 2019. 1, 3

[5] Guobin Chen, Wongun Choi, and Xiang Yu *et al*. Learning efficient object detection models with knowledge distillation. In *NeurIPS*, 2017. 3

[6] Kai Chen, Jiaqi Wang, and Jiangmiao Pang *et al*. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[7] Ze Chen, Zhihang Fu, and Rongxin Jiang *et al*. SLV: spatial likelihood voting for weakly supervised object detection. In *CVPR*, 2020. 2, 3

[8] Xing Dai, Zeren Jiang, and Zhao Wu *et al*. General instance distillation for object detection. In *CVPR*, 2021. 3

[9] Ali Diba, Vivek Sharma, and Ali Mohammad Pazandeh *et al*. Weakly supervised cascaded convolutional networks. In *CVPR*, 2017. 3

[10] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 1997. 2

[11] Chen. *et al*. Point-to-box network for accurate object detection via single point supervision. In *ECCV*, 2022. 1, 2, 3

[12] Liu. *et al*. Robust object detection with inaccurate bounding boxes. In *ECCV*, 2022. 1, 2, 3, 6, 8, 11

[13] Mark Everingham, Luc Van Gool, and Christopher K. I. Williams *et al*. The pascal visual object classes (VOC) challenge. *IJCV*, 2010. 5, 11

[14] Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017. 3

[15] Ross B. Girshick. Fast R-CNN. In *ICCV*, 2015. 2

[16] Jianyuan Guo, Kai Han, and Yunhe Wang *et al*. Distilling object detectors via decoupled features. In *CVPR*, 2021. 3

[17] Bo Han, Quanming Yao, and Xingrui Yu *et al*. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018. 3, 6

[18] Kaiming He, Georgia Gkioxari, and Piotr Dollár *et al*. Mask R-CNN. In *ICCV*, 2017. 1, 3

[19] Kaiming He, Xiangyu Zhang, and Shaoqing Ren *et al*. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[20] Yihui He, Chenchen Zhu, and Jianren Wang *et al*. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, 2019. 1, 6

[21] Geoffrey Hinton and Oriol *et al*. Vinyals. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 3

[22] Lu Jiang, Zhengyuan Zhou, and Thomas Leung *et al*. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018. 3

[23] Vadim Kantorov, Maxime Oquab, and Minsu Cho *et al*. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2016. 3

[24] Junnan Li, Caiming Xiong, Richard Socher, and Steven C. H. Hoi. Towards noise-resistant object detection with noisy annotations. *CoRR*, abs/2003.01285, 2020. 1, 3

[25] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *CVPR*, 2017. 3

[26] Mingxiang Liao, Fang Wan, and Yuan Yao *et al*. End-to-end weakly supervised object detection with sparse proposal evolution. In *ECCV*, 2022. 3

[27] Tsung-Yi Lin, Piotr Dollár, and Ross B. Girshick *et al*. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[28] Tsung-Yi Lin, Priya Goyal, and Ross B. Girshick *et al*. Focal loss for dense object detection. In *ICCV*, 2017. 1, 2, 4, 6

[29] Tsung-Yi Lin and Michael *et al*. Maire. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 5, 11

[30] Wei Liu, Dragomir Anguelov, and Dumitru Erhan *et al*. SSD: single shot multibox detector. In *ECCV*, 2016. 2

[31] Ze Liu, Yutong Lin, and Yue Cao *et al*. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[32] Xingjun Ma, Yisen Wang, and Michael E. Houle *et al*. Dimensionality-driven learning with noisy labels. In *ICML*, 2018. 3

[33] Jiafeng Mao, Qing Yu, and Kiyoharu Aizawa. Noisy localization annotation refinement for object detection. *TIS*, 2021. 1, 3

[34] Joseph Redmon, Santosh Kumar Divvala, and Ross B. Girshick *et al*. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[35] Shaoqing Ren, Kaiming He, and Ross B. Girshick *et al*. Faster R-CNN: towards real-time object detection with region proposal networks. *TPAMI*, 2017. 1, 2, 6

[36] Zhongzheng Ren, Zhiding Yu, and Xiaodong Yang *et al*. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *CVPR*, 2020. 2, 3

[37] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: refurbishing unclean samples for robust deep learning. In *ICML*, 2019. 3

[38] Peize Sun, Rufeng Zhang, and Yi Jiang *et al*. Sparse R-CNN: end-to-end object detection with learnable proposals. In *CVPR*, 2021. 2, 7

[39] Peng Tang and Xinggang Wang *et al*. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017. 2, 3

[40] Peng Tang, Xinggang Wang, and Song Bai *et al*. PCL: proposal cluster learning for weakly supervised object detection. *TPAMI*, 2020. 3

[41] Koen E. A. van de Sande, Jasper R. R. Uijlings, and Theo Gevers *et al*. Segmentation as selective search for object recognition. In *ICCV*, 2011. 2, 3

[42] Fang Wan, Pengxu Wei, and Zhenjun Han *et al*. Min-entropy latent model for weakly supervised object detection. *TPAMI*, 2019. 3

[43] Tao Wang, Li Yuan, and Xiaopeng Zhang *et al*. Distilling object detectors with fine-grained feature imitation. In *CVPR*, 2019. 3

[44] Yunchao Wei, Zhiqiang Shen, and Bowen Cheng *et al*. TS ^2 2 C: tight box mining with surrounding segmentation context for weakly supervised object detection. In *ECCV*, 2018. 3

[45] Youjiang Xu, Linchao Zhu, and Yi Yang *et al*. Training robust object detectors from noisy category labels and imprecise bounding boxes. *TIP*, 2021. 1, 3

[46] Ke Yang, Dongsheng Li, and Yong Dou. Towards precise end-to-end weakly supervised object detection network. In *ICCV*, 2019. 3

[47] Zhendong Yang, Zhe Li, and Xiaohu Jiang *et al*. Focal and global knowledge distillation for detectors. In *CVPR*, 2022. 3

[48] Xuehui Yu, Pengfei Chen, and Di Wu *et al*. Object localization under single coarse point supervision. In *CVPR*, 2022. 2

[49] Xuehui Yu, Yuqi Gong, and Nan Jiang *et al*. Scale match for tiny person detection. In *WACV*, 2020. 1, 2

[50] Xiaosong Zhang, Fang Wan, and Chang Liu *et al*. Freeanchor: Learning to match anchors for visual object detection. In *NeurIPS*, 2019. 1, 6

[51] Xiaopeng Zhang, Yang Yang, and Jiashi Feng. Learning to localize objects with noisy labeled instances. In *AAAI*, 2019. 3, 6

[52] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018. 3

[53] Zhaohui Zheng, Rongguang Ye, and Ping Wang *et al*. Localization distillation for dense object detection. In *CVPR*, 2022. 3

[54] Xizhou Zhu, Weijie Su, and Lewei Lu *et al*. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2, 7

[55] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 3

Appendix

A. Codes

The code of this paper is also included as a zip file (ssd-det.zip) in the supplementary. The submitted version contains training codes on MS-COCO[29] and VOC[13]. The details are given in README.md in the zip file.

B. Details of SSD-Det Deployment

Structure Details. Fig. 6 depicts the detailed structure of the basic box refiner, while Fig. 8 depicts the detailed structure of our SSD-Det.

Implementation Details. ResNet-50 is used as the backbone network unless otherwise specified, and FPN is adopted for feature fusion. The mini-batch is 16 images; all models are trained with 8/2 GPUs and 2 images per GPU for MS-COCO/VOC. The training epoch numbers are set as 12, and the learning rate is set as 0.02/0.002 and decays by 0.1 at the 8-th and 11-th epoch for MS-COCO/VOC. In default settings, the backbone is initialized with the pre-trained weights on ImageNet and other newly added layers are initialized with Xavier. In 40% noise rate in MS-COCO, the original settings of basic sampling are:$(v \cdot s) \in \{0.7, 0.8, 1, 1.2, 1.3\}$, $(v/s) \in \{0.7, 0.8, 1, 1.2, 1.3\}$ and $(o_x, o_y) \in \{(0,0), (2,0), (0,2), (-2,0), (-2,-2)\}$ is used to jitter the centre position. Those are set the half for the 20% noise rate dataset. The settings in VOC are the same and adaptively changed for other noise rate datasets. In negative sampling, we randomly sample 500 boxes, filter out those which have high IoU (0.3) with all positive proposals and obtain the final negative sample set $\mathcal{N}$. The loss weights are set as $\alpha_1, \alpha_2, \alpha_3$ and α_4 are set as 1, 0.25, 0.25 and 4, respectively, without much hyper-parameter tuning.

Synthetic Noisy Dataset. Following [12], we simulate noisy bounding boxes by perturbing clean boxes from the original annotations. Specifically, cx, cy, w, and h denote an object's the center x coordinate, center y coordinate, width, and height, respectively. We simulate an inaccurate bounding box by randomly shifting and scaling the box as follows:

$$\begin{cases} \hat{cx} = cx + \Delta_x \cdot w, & \hat{cy} = cy + \Delta_y \cdot h \\ \hat{w} = (1 + \Delta_w) \cdot w, & \hat{h} = (1 + \Delta_h) \cdot h \end{cases} \tag{10}$$

where Δx, Δy, Δw, and Δh obey the uniform distribution $U(-r, r)$, and r is the box noise level. For example, when r = 40%, Δx, Δy, Δw, and Δh are in the range of $(-0.4, 0.4)$. We simulate various box noise levels ranging from 10% to 40% for the VOC dataset and $\{20\%, 40\%\}$ for the MS-COCO dataset. Eq. 10 is conducted on every bounding box in the training dataset.

C. Details of Average IoU

Average IoU is the evaluation metric of the performance of dataset refine, and the higher average IoU means the better performance. Table 10 shows that the quality of dataset refinement is greatly improved after OA-MIL solves the drift problem. By simply filtering out the pseudo box with $IoU = 0$, the performance of OA-MIL improves from 47.6 to 54.4. Further, once filtering out the pseudo box with $IoU = 0$, the performance of OA-MIL improves from 47.6 to 54.4. If the pseudo frame with $IoU \leq 0.5$ is filtered out, OA-MIL's refinement performance is close to ours. If only the proposals whose IoU with GT is greater than 1e-5 are counted (second line), the average IoU of OA-MIL is greatly increased, meaning lots of extremely low-quality refined results, while IoU of our SSD-Det remains essentially unchanged.

Methods	Average IoU			
	IoU≥0	IoU >0	IoU >0.3	IoU >0.5
(40% Noise Level)	46.4	-	-	-
OA-MIL[12]	47.6	54.4	57.1	67.5
SSD-Det	65.1	65.1	67.7	72.7

Table 10: The average IoU of different methods' refined boxes with clean GT on MS-COCO under 40% Noise Level.

D. Qualitative Results

Affect of Re-Train. As most WSOD methods do, we re-run the experiments by training a fully supervised detector, *e.g.* Faster R-CNN or RetinaNet, to regress the object locations more precisely. As shown in Table 7, we get a better result of 20.29 AP and 34.37 AP on 40% and 20% noise datasets. We also find that if the SSD-Det only trains the refiner and uses the pseudo label to train the FasterRCNN, the result is good but lower than re-train after the end-to-end training given in Table 7 (row 1). This is because joint training is beneficial for box refinement.

Methods	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l
Box Refiner+Re-Train	29.0	54.4	28.2	**17.7**	32.3	36.4
SSD-Det	27.6	53.9	26.0	16.0	31.0	34.9
SSD-Det+Re-Train	**29.3**	**54.8**	**29.0**	17.1	**32.9**	**36.9**

Table 11: Comparisons of end-to-end and re-train (40% noise).

Experiments on Different Detectors. Experiments are conducted on ResNet50. We re-train the different detectors with corrected labels. Table 12 shows the detection results, verifying the robustness of our method.

Visualization. Fig. 8 shows the refined boxes predicted by OA-MIL and our SSD-Det on the MS-COCO datasets with 40% box noise. We can observe that OA-MIL suffers from object drift, group prediction, part domination problems. Fig. 9 shows the qualitative results of the OA-MIL

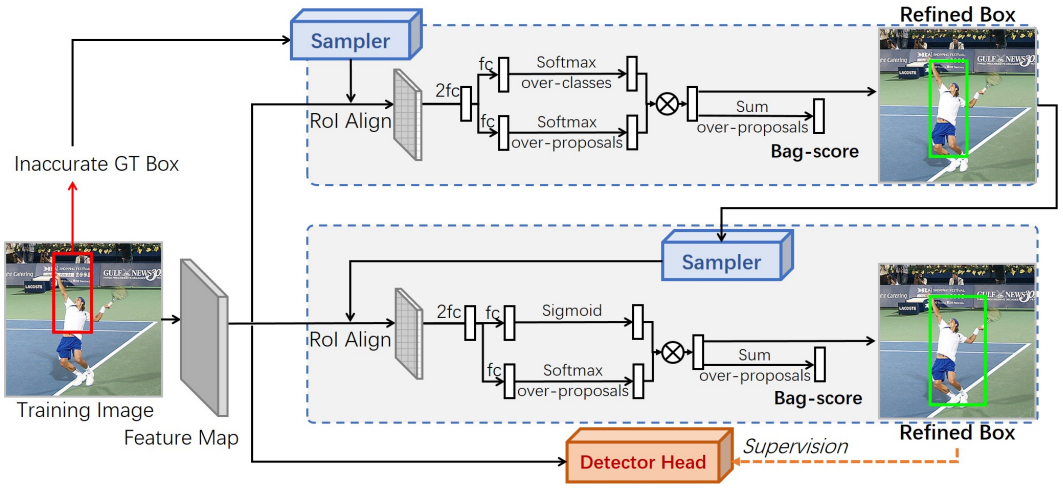


Figure 6: The basic box refiner.

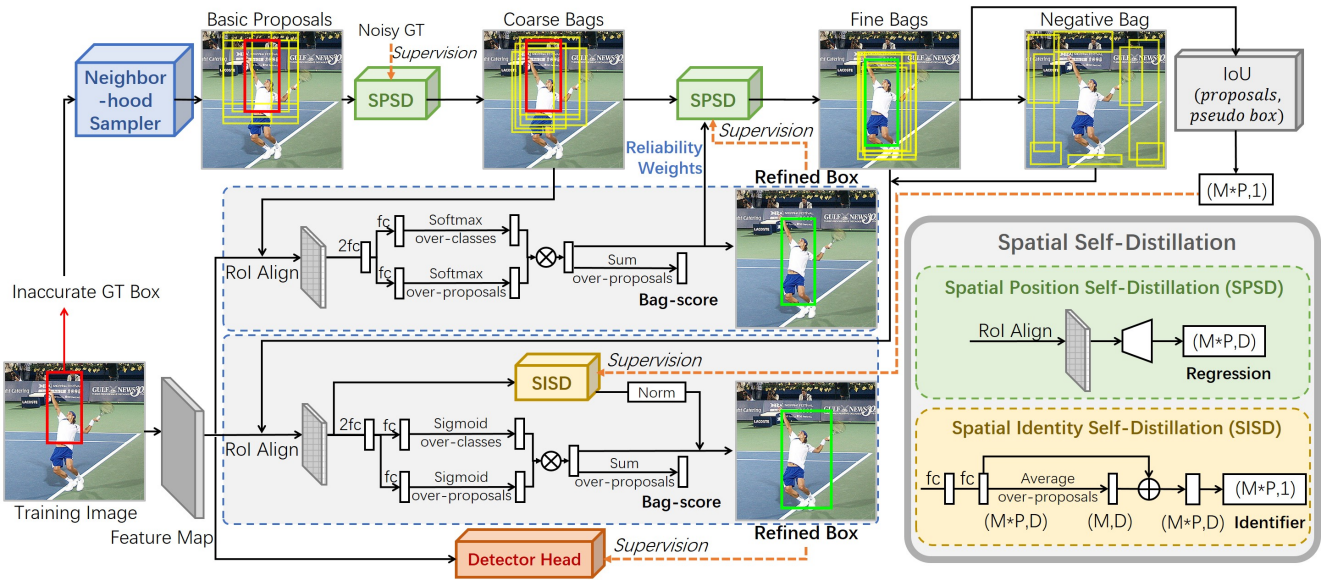


Figure 7: SSD-Det (SPSD shares backbone with the detector).

and our SSD-Det on the MS-COCO datasets with 40% box noise.

Detectors	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l
Faster R-CNN	29.3	54.8	29.0	17.1	32.9	36.9
RetinaNet	28.6	52.8	28.8	17.1	32.3	36.4
RepPoints	28.6	53.7	28.0	16.8	32.0	37.0
Free-Anchor	29.4	54.1	29.6	17.0	32.4	37.6
Sparse R-CNN	34.3	60.2	36.4	22.4	37.5	43.7
Deformable-DETR	35.0	60.7	37.4	23.6	38.1	44.4

Table 12: Different detectors for re-train (40% noise).

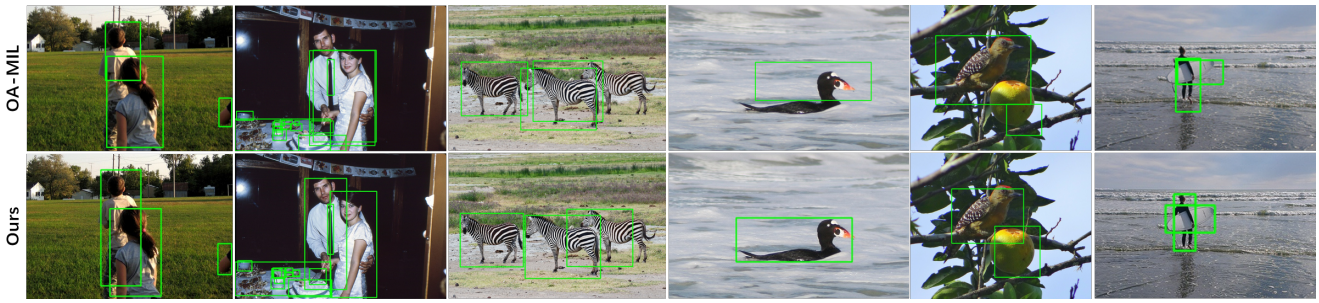


Figure 8: Examples of the refined instances (MS-COCO train set under 40% noise level).

Figure 9: Qualitative results on MS-COCO validation set.