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[31] 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 11 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 11: 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 12: 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 13 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

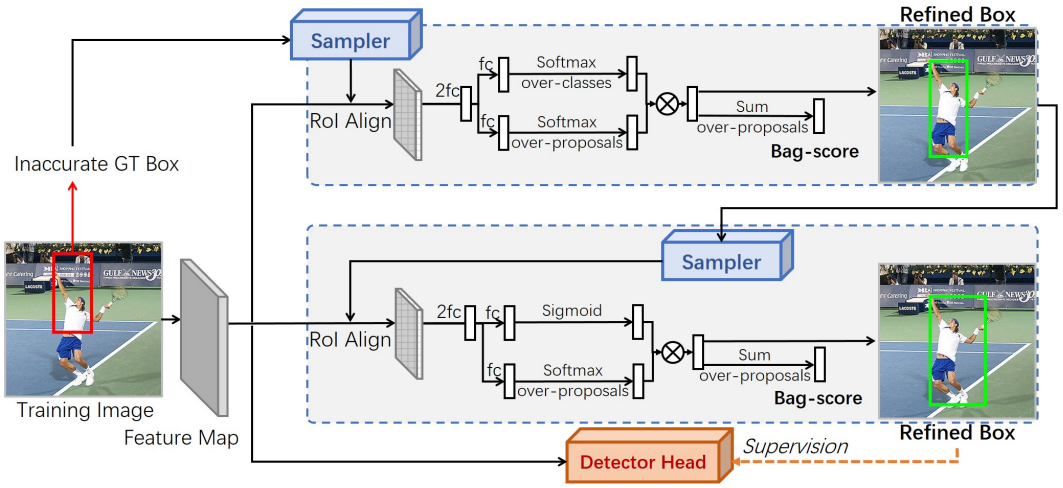

Figure 6: The basic box refiner.

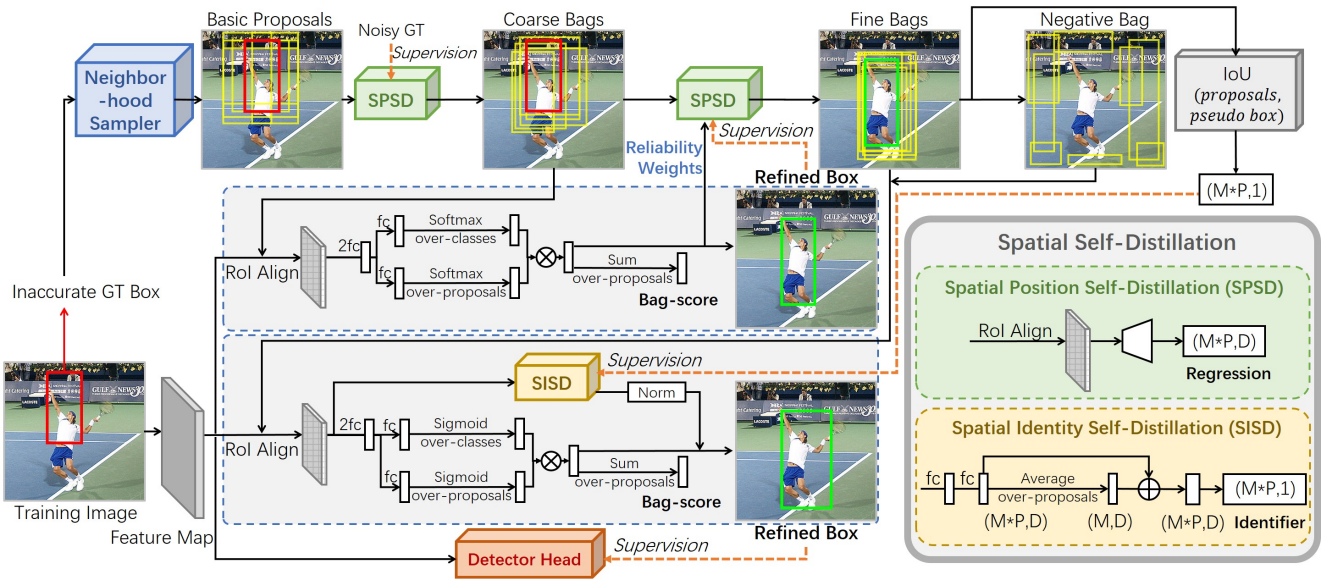

Figure 7: SSD-Det (SPSD shares backbone with the detector).

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

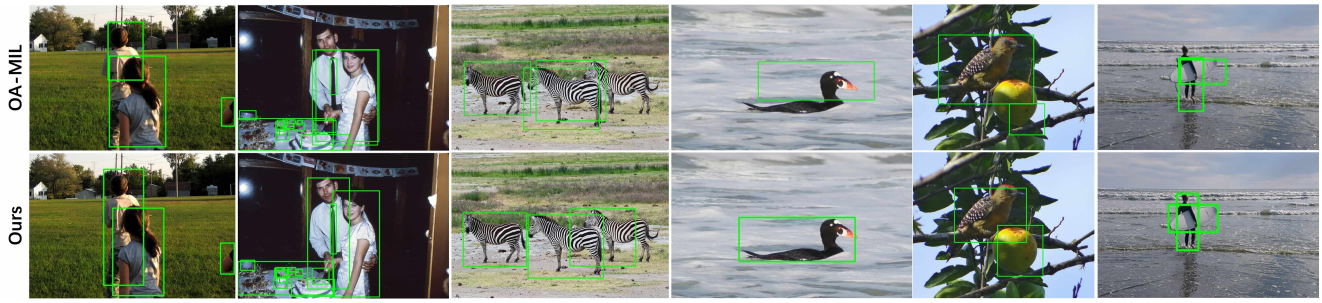

Figure 8: Examples of the refined instances (MS-COCO train set under 40% noise level).

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