Boosting Weakly Supervised Object Detection via Learning Bounding Box Adjusters

Bowen Dong[1] Zitong Huang[1] Yuelin Guo[1] Qilong Wang[2] Zhenxing Niu[3] Wangmeng Zuo[1,4]✉
[1]Harbin Institute of Technology [2]Tianjin University [3]Alibaba Damo Academay [4]Pazhou Lab
{cndongsky, zitonghuang99, zhenxingniu}@gmail.com gyl2565309278@163.com qlwang@tju.edu.cn wmzuo@hit.edu.cn

Contents

A. Discussion of EM-like training algorithm

The reason why EM-like training is necessary is that the problem is formulated as a bi-level optimization problem, direct joint training to solve this problem is harmful to generalization ability of LBBA. And EM-like training can keep that of LBBA. Here we state why formulating WSOD problem as bi-level optimization.

In particular, E-step is used to update latent variable $\hat{\mathbf{b}}$,

$$\hat{\mathbf{b}} = \arg\max_{\mathbf{b}_{\text{latent}}} \log P(\mathbf{y}|\mathbf{b}_{\text{latent}}) - \mathcal{L}(\mathbf{b}_{\text{latent}}, f(\mathbf{I}, \mathbb{P}; \theta_f)). \quad (1)$$

For WSOD with box regression, $\mathbf{y}$ is image class labels, $\mathcal{L}$ is defined as box regression loss (*e.g.*, smooth L1 loss [4] for bounding box regression), $\hat{\mathbf{b}}$ means latent bounding box variables, and $P(\mathbf{y}|\mathbf{b}_{\text{latent}})$ is probability of $\mathbf{y}$ with given $\mathbf{b}_{\text{latent}}$ in WSOD training. And $f(\mathbf{I}, \mathbb{P}; \theta_f)$ is bounding box output from WSOD network f with corresponding parameters θ_f. We mainly discuss $\mathcal{L}$ in next paragraphs. Then, M-step is deployed to update the model parameters θ_f.

$$\theta_f = \arg\min_{\theta_f} \mathcal{L}(\hat{\mathbf{b}}, f(\mathbf{I}, \mathbb{P}; \theta_f)), \quad (2)$$

where $\mathcal{L}$ is a combination of weakly supervised object detection loss $\mathcal{L}_{\text{wsod}}$ and bounding box regression loss $\mathcal{L}_{\text{bbr}}$.

As mentioned above, previous methods utilize precomputed proposals as well as pseudo ground-truth mining in E-step, and then update box regression branch of WSOD network in M-step. However, optimizing $P(\mathbf{y}|\mathbf{b}_{\text{latent}})$ in E-step with only image-level supervision to imporve quality of $\hat{\mathbf{b}}$ is difficult. Besides, when optimizing $\mathcal{L}$ in E-step, precomputed proposals are designed for generating region proposals for box regression of object detection, which are not suitable for final object localization. To tackle this problem, we want to use extra well-annotated data to supervise a learnable model, make it generate more precise $\hat{\mathbf{b}}$ in E-step. Therefore, we aim to introduce a class-agnostic Learnable Bounding Box Adjuster (LBBA) $g(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_g)$ trained on a full-annotated auxiliary dataset $\mathbb{X}^{\text{aux}}$. To this end, we suggest to utilize LBBA g to generate latent variable $\hat{\mathbf{b}}^{\text{aux}}$ on $\mathbb{X}^{\text{aux}}$.

$$\begin{aligned} \hat{\mathbf{b}}_{\text{aux}} &= g(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_g) \\ \theta_g &= \arg\min_{\theta_g} \mathcal{L}_{\text{bba}}(\{\mathbf{b}^{\text{aux}}\}, g(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_g)) \end{aligned} \quad (3)$$

After introducing LBBA g into WSOD, our WSOD problem can be transferred into a **bi-level optimization problem**, here we state how to build bi-level optimization.

Lower subproblem. During M-step, WSOD network f is supervised by both image class label $\mathbf{y}$ as well as latent variable $\hat{\mathbf{b}}^{\text{aux}}$, which is output of LBBA network $g(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_g)$. Therefore we update parameters of WSOD network $\theta_{f^{\text{aux}}}$ by minimizing $\mathcal{L}_{\text{wsod}} + \mathcal{L}_{\text{bbr}}$, which is shown as Eq. 4. And Eq. 4 also stands for the lower subproblem of bi-level optimization.

$$\theta_{f^{\text{aux}}} = \arg\min_{\theta_{f^{\text{aux}}}} (\mathcal{L}_{\text{wsod}} + \mathcal{L}_{\text{bbr}})(\hat{\mathbf{b}}^{\text{aux}}, f^{\text{aux}}(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_{f^{\text{aux}}})) \quad (4)$$

Upper subproblem. Thus, taking above equations into consideration, WSOD parameters $\theta_{f^{\text{aux}}}$ can be seen as a function of LBBA parameters θ_g (*i.e.*, $\theta_{f^{\text{aux}}}(\theta_g)$). Thus, in E-step the upper subproblem on θ_g is defined for optimizing $\mathcal{L}_{\text{bba}}$ on the WSOD network $f^{\text{aux}}(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_{f^{\text{aux}}}(\theta_g))$,

$$\theta_g = \arg\min_{\theta_g} \mathcal{L}_{\text{bba}}(\{\mathbf{b}^{\text{aux}}\}, f^{\text{aux}}(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_{f^{\text{aux}}}(\theta_g))) \quad (5)$$

where g generates adjusted bounding box regression for given proposals from WSOD f^{aux}. Thus upper subproblem has transferred into a fully-supervised setting. Furthermore, to ease the training difficulty of the upper subproblem and improve the precision of $\hat{\mathbf{b}}^{\text{aux}}$, we modify the upper subproblem by requiring LBBA accurately predicts the ground-truth boxes, resulting in the following bi-level optimization formulation.

$$\begin{aligned} &\min_{\theta_g} \mathcal{L}_{\text{bba}}(\{\mathbf{b}^{\text{aux}}\}, g(\mathbf{I}^{\text{aux}}, f^{\text{aux}}(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_{f^{\text{aux}}}); \theta_g)) \\ &s.t. \theta_{\text{f}} = \arg\min_{\theta_{\text{f}}} \mathcal{L}_{\text{wsod}} + \mathcal{L}_{\text{bbr}}(\hat{\mathbf{b}}^{\text{aux}}, f^{\text{aux}}(\mathbf{I}^{\text{aux}}, \mathbb{P}^{\text{aux}}; \theta_{f^{\text{aux}}})). \end{aligned} \quad (6)$$

B. Datasets

To illustrate the effectiveness of our method, we conduct experiments on various representative datasets: PASCAL VOC 2007 and 2012 datasets, MS-COCO dataset, and ILSVRC 2013 detection dataset.

B.1. Auxiliary Datasets

COCO-60 Dataset MS-COCO 2017 [10] is a large-scale object detection dataset. Note that MS-COCO dataset includes 80 different object classes. To eliminate semantic overlap and show the generalization ability of our method, we construct a subset of MS-COCO by excluding PASCAL VOC classes instance annotations and call it COCO-60. As such, COCO-60 dataset contains ∼98K training images and ∼4K validation images, respectively. Construction details are shown as Appendix B.4.

ILSVRC-Source Dataset To prove that our method can be generalized to more categories, we conduct extended experiments on the ILSVRC2013 detection dataset. ILSVRC detection dataset contains 200 categories, which is much more than that for PASCAL VOC or COCO-20. To construct the corresponding auxiliary dataset, we select the first 100 classes sorted in alphabetic order as the source classes in the auxiliary dataset. Construction details are shown as Appendix B.5.

B.2. Target Datasets

PASCAL VOC Dataset PASCAL VOC 2007 and 2012 datasets contain 9,963 images and 22,531 images collected from 20 object classes, respectively. For fair comparison, we use *trainval* set for training WSOD networks and report evaluation results on *test* set. During the training process, only image-level labels are used as supervision.

COCO-20 Dataset To verify the generalization ability of our LBBA, we construct another target dataset from MS-COCO dataset namely COCO-20 dataset. Note that the COCO-20 dataset has the same 20 classes as PASCAL VOC dataset, but containing more complicated scenarios in images. Construction details are shown as Appendix B.5.

ILSVRC-Target Dataset ILSVRC detection dataset contains 200 categories. To construct the target dataset and avoid semantic overlaps with the corresponding auxiliary dataset, we select the last 100 classes sorted in alphabetic order as target classes in our weakly supervised object detection dataset. Construction details are shown as Appendix B.5.

B.3. Auxiliary-Target Pairs

From these datasets, we divide them into four dataset-pair settings, an auxiliary dataset corresponding to a target dataset, to deploy experiments. Table 6 give the dataset-pair settings. Setting 1 and Setting 2 are mentioned in section 4 of main paper and we will introduce details of setting 3 and setting 4 in Appendix B.4 and Appendix B.5. Then we will state more experimental results in Appendix F and Appendix G.

B.4. Construction of COCO-60/COCO-20

To simplify the statement, we define COCO-60 classes as the categories in original COCO classes but excluding PASCAL VOC classes. Then we state how to construct COCO-60 dataset and COCO-20 dataset.

To construct COCO-60 dataset, we first keep annotations of COCO-60 classes in COCO 2017 *train* set, then we select images which contain at least one instance of COCO-60

Table 1. Single model detection per-class results on PASCAL VOC 2007, where $^+$ means the results with multi-scale testing, * means joint training of WSOD models on the auxiliary dataset and weakly-annotated dataset.

Methods	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	TV	AP
WSDDN [2]	39.4	50.1	31.5	16.3	12.6	64.5	42.8	42.6	10.1	35.7	24.9	38.2	34.4	55.6	9.4	14.7	30.2	40.7	54.7	46.9	34.8
OICR$^+$ [18]	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	41.2
PCL$^+$ [17]	54.4	69.0	39.3	19.2	15.7	62.9	64.4	30.0	25.1	52.5	44.4	19.6	39.3	67.7	17.8	22.9	46.6	57.5	58.6	63.0	43.5
Yang *et al.*$^+$ [21]	57.6	70.8	50.7	28.3	27.2	72.5	69.1	65.0	26.9	64.5	47.4	47.7	53.5	66.9	13.7	29.3	56.0	54.9	63.4	65.2	51.5
C-MIDN$^+$ [20]	53.3	71.5	49.8	26.1	20.3	70.3	69.9	68.3	28.7	65.3	45.1	64.6	58.0	71.2	20.0	27.5	54.9	54.9	69.4	63.5	52.6
Arun *et al.* [1]	66.7	69.5	52.8	31.4	24.7	74.5	74.1	67.3	14.6	53.0	46.1	52.9	69.9	70.8	18.5	28.4	54.6	60.7	67.1	60.4	52.9
WSOD2$^+$ [23]	65.1	64.8	57.2	39.2	24.3	69.8	66.2	61.0	29.8	64.6	42.5	60.1	71.2	70.7	21.9	28.1	58.6	59.7	52.2	64.8	53.6
MIST-Full [12]	68.8	77.7	57.0	27.7	28.9	69.1	74.5	67.0	32.1	73.2	48.1	45.2	54.4	73.7	35.0	29.3	64.1	53.8	65.3	65.2	54.9
MSD-Ens$^+$ [9]	70.5	69.2	53.3	43.7	25.4	68.9	68.7	56.9	18.4	64.2	15.3	72.0	74.4	65.2	15.4	25.1	53.6	54.4	45.6	61.4	51.1
OICR+UBBR [7]	59.7	44.8	54.0	36.1	29.3	72.1	67.4	70.7	23.5	63.8	31.5	61.5	63.7	61.9	37.9	15.4	55.1	57.4	69.9	63.6	52.0
Ours	65.4	73.7	53.1	44.8	27.3	73.1	73.7	72.2	29.8	69.2	51.1	68.7	56.4	71.8	20.3	27.1	61.4	60.3	65.5	65.9	**56.5**
Ours$^+$	70.3	72.3	48.7	38.7	30.4	74.3	76.6	69.1	33.4	68.2	50.5	67.0	49.0	73.6	24.5	27.4	63.1	58.9	66.0	69.2	**56.6**
Upper bounds:																					
Faster R-CNN [11]	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6	69.9
Zhong *et al.* (R50-C4)* [24]	64.4	45.0	62.1	42.8	42.4	73.1	73.2	76.0	28.2	78.6	28.5	75.1	74.6	67.7	57.5	11.6	65.6	55.4	72.2	61.3	57.8
Zhong *et al.* (R50-C4)$^{+*}$ [24]	64.8	50.7	65.5	45.3	46.4	75.7	74.0	80.1	31.3	77.0	26.2	79.3	74.8	66.5	57.9	11.5	68.2	59.0	74.7	65.5	59.7

Table 2. Single model detection results on PASCAL VOC 2012, where $^+$ means the results with multi-scale testing, * means joint training of WSOD models on the auxiliary dataset and weakly-annotated dataset.

Methods	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	TV	AP
OICR$^+$	67.7	61.2	41.5	25.6	22.2	54.6	49.7	25.4	19.9	47.0	18.1	26.0	38.9	67.7	2.0	22.6	41.1	34.3	37.9	55.3	37.9
PCL$^+$ [17]	58.2	66.0	41.8	24.8	27.2	55.7	55.2	28.5	16.6	51.0	17.5	28.6	49.7	70.5	7.1	25.7	47.5	36.6	44.1	59.2	40.6
Yang *et al.*$^+$	64.7	66.3	46.8	28.5	28.4	59.8	58.6	70.9	13.8	55.0	15.7	60.5	63.9	69.2	8.7	23.8	44.7	52.7	41.5	62.6	46.8
WSOD2$^+$ [23]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	47.2
Arun *et al.* [1]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48.4
C-MIDN$^+$ [20]	72.9	68.9	53.9	25.3	29.7	60.9	56.0	78.3	23.0	57.8	25.7	73.0	63.5	73.7	13.1	28.7	51.5	35.0	56.1	57.5	50.2
MIST (Full)$^+$ [12]	78.3	73.9	56.5	30.4	37.4	64.2	59.3	60.3	26.6	66.8	25.0	55.0	61.8	79.3	14.5	30.3	61.5	40.7	56.4	63.5	52.1
Ours	77.0	71.0	62.0	40.0	37.5	67.4	62.5	68.3	23.6	71.4	25.6	78.4	71.9	74.3	6.7	29.2	62.8	50.6	47.8	62.1	**54.5**
Ours$^+$	78.6	71.5	62.7	41.3	38.6	68.8	64.1	71.0	23.2	70.5	24.2	79.1	74.1	75.3	6.5	29.7	63.4	51.8	50.2	63.9	**55.4**
Upper bounds:																					
Faster R-CNN [11]	82.3	76.4	71.0	48.4	45.2	72.1	72.3	87.3	42.2	73.7	50.0	86.8	78.7	78.4	77.4	34.5	70.1	57.1	77.1	58.9	67.0

classes in COCO 2017 *train* set to construct our COCO-60 *train* set. Next we keep the same steps to build up our COCO-60 *val* set.

Besides, we also follow Zhong *et al*. [24] to define a COCO-60-clean dataset. Particularly, we select images which **only contain instances of COCO-60 classes** in COCO 2017 *train* set to construct COCO-60-clean *train* set, and obtain only 21987 training images. Compared to COCO-60 dataset, COCO-60-clean dataset does not exist objects of VOC classes in the background of images, such that this dataset is cleaner than our COCO-60 dataset and easier to learn. We will discuss the difference between our method and Zhong *et al*. [24] based on COCO-60 and COCO-60-clean datasets.

As for COCO-20 dataset, we select images which only contain instances of 20 PASCAL VOC classes in COCO 2017 *train* set to construct our COCO-20 *train* set. Next we keep annotations of 20 PASCAL VOC classes in COCO 2017 *val* set, and then select images which contain at least one instance of 20 PASCAL VOC classes in COCO 2017 *val* set to construct our COCO-20 *val* set.

B.5. Construction of ILSVRC-Source/Target

The original ILSVRC dataset contains a training set and a validation set. Firstly, We split the validation set into val1 validation set and val2 validation set. Then we state how to construct ILSVRC-Source dataset and ILSVRC-Target dataset.

To construct ILSVRC-Source training set, we keep images of the first 100 categories sorted in alphabetic order from val1 and sample 1000 images per category in the same 100 categories from ILSVRC training set as data augmentation.

To construct ILSVRC-Target training set, we keep images of the latter 100 categories sorted in alphabetic order from val1 and sample a maximum of 1000 images per category in latter categories from ILSVRC training set to augment it,while keeping only image-level labels. And to construct ILSVRC-Target test set, we keep images of the same 100 categories from val2.

C. Implementation Details

C.1. Overall Implementation Details

For LBBAs, we apply Faster R-CNN [11] with backbone of ResNet-50 [5] and we adopt class-agnostic bounding box adjusters to eliminate potential semantic information leak in bounding box refinement. For WSOD network, we apply OICR [18] with a backbone of VGG-16 [16] and introduce a class-agnostic bounding box regression branch. Following the settings of [2, 18, 17, 21, 12, 1, 24], we initialize backbone models of two networks with ImageNet [3] pre-trained weights while other layers are randomly initialized. As suggested in [12, 21, 17, 18, 23], we use MCG

Table 3. Single model correct localization (CorLoc) results on PASCAL VOC 2007, where $^+$ means the results with multi-scale testing, * means joint training of WSOD models on the auxiliary dataset and weakly-annotated dataset.

Methods	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	TV	CorLoc
WSDDN [2]	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
OICR$^+$	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	60.6
PCL$^+$ [17]	79.6	85.5	62.2	47.9	37.0	83.8	83.4	43.0	38.3	80.1	50.6	30.9	57.8	90.8	27.0	58.2	75.3	68.5	75.7	78.9	62.7
Li$^+$ [8]	85.0	83.9	58.9	59.6	43.1	79.7	85.2	77.9	31.3	78.1	50.6	75.6	76.2	88.4	49.7	56.4	73.2	62.6	77.2	79.9	68.6
C-MIL$^+$ [19]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65.0
Yang *et al.*$^+$	80.0	83.9	74.2	53.2	48.5	82.7	86.2	69.5	39.3	82.9	53.6	61.4	72.4	91.2	22.4	57.5	83.5	64.8	75.7	77.1	68.0
WSOD2$^+$ [23]	87.1	80.0	74.8	60.1	36.6	79.2	83.8	70.6	43.5	88.4	46.0	74.7	87.4	90.8	44.2	52.4	81.4	61.8	67.7	79.9	69.5
Arun *et al.*[1]	88.6	86.3	71.8	53.4	51.2	87.6	89.0	65.3	33.2	86.6	58.8	65.9	87.7	93.3	30.9	58.9	83.4	67.8	78.7	80.2	70.9
MIST (Full)$^+$ [12]	87.5	82.4	76.0	58.0	44.7	82.2	87.5	71.2	49.1	81.5	51.7	53.3	71.4	92.8	38.2	52.8	79.4	61.0	78.3	76.0	68.8
WSLAT-Ens [13]	78.6	63.4	66.4	56.4	19.7	82.3	74.8	69.1	22.5	72.3	31.0	63.0	74.9	78.4	48.6	29.4	64.6	36.2	75.9	69.5	58.8
MSD-Ens$^+$ [9]	89.2	75.7	75.1	66.5	58.8	78.2	88.9	66.9	28.2	86.3	29.7	83.5	83.3	92.8	23.7	40.3	85.6	48.9	70.3	68.1	66.8
OICR+UBBR [7]	47.9	18.9	63.1	39.7	10.2	62.3	69.3	61.0	27.0	79.0	24.5	67.9	79.1	49.7	28.6	12.8	79.4	40.6	61.6	28.4	47.6
Ours	89.6	82.0	73.6	55.3	48.9	86.3	87.3	83.1	45.3	87.7	48.3	82.3	80.6	90.8	36.3	52.0	88.7	66.1	81.7	80.3	**72.3**
Ours$^+$	89.2	82.0	74.2	53.2	51.2	84.8	87.5	83.7	46.2	87.0	48.3	84.7	79.9	92.4	40.3	47.6	88.7	65.6	81.0	81.7	**72.5**
Upper bounds:																					
Faster R-CNN [11]	99.6	96.1	99.1	95.7	91.6	94.9	94.7	98.3	78.7	98.6	85.6	98.4	98.3	98.8	96.6	90.1	99.0	80.1	99.6	93.2	94.3
Zhong *et al.* (R50-C4)* [24]	86.7	62.4	87.1	70.2	66.4	85.3	87.6	88.1	42.3	94.5	32.3	87.7	91.2	88.8	71.2	20.5	93.8	51.6	87.5	76.7	73.6
Zhong *et al.* (R50-C4)$^{+*}$ [24]	87.5	64.7	87.4	69.7	67.9	86.3	88.8	88.1	44.4	93.8	31.9	89.1	92.9	86.3	71.5	22.7	94.8	56.5	88.2	76.3	74.4

Table 4. Single model correct localization (CorLoc) results on PASCAL VOC 2012, where $^+$ means the results with multi-scale testing, * means joint training of WSOD models on the auxiliary dataset and weakly-annotated dataset.

Methods	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	TV	CorLoc
OICR$^+$ [18]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	62.1
PCL$^+$ [17]	77.2	83.0	62.1	55.0	49.3	83.0	75.8	37.7	43.2	81.6	46.8	42.9	73.3	90.3	21.4	56.7	84.4	55.0	62.9	82.5	63.2
Shen [15]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	63.5
Li$^+$ [8]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	67.9
C-MIL$^+$ [19]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	67.4
Yang *et al.*$^+$ [21]	82.4	83.7	72.4	57.9	52.9	86.5	78.2	78.6	40.1	86.4	37.9	67.9	87.6	90.5	25.6	53.9	85.0	71.9	66.2	84.7	69.5
Arun *et al.*[1]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	69.5
WSOD2$^+$ [23]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	71.9
MIST (Full)$^+$ [12]	91.7	85.6	71.7	56.6	55.6	88.6	77.3	63.4	53.6	90.0	51.6	62.6	79.3	94.2	32.7	58.8	90.5	57.7	70.9	85.7	70.9
Ours	91.9	87.4	81.9	66.7	58.5	91.2	79.9	67.3	50.0	91.9	49.6	80.3	89.6	91.8	15.6	58.8	88.7	67.1	70.2	85.0	**73.2**
Ours$^+$	91.9	87.2	81.0	66.9	61.3	90.7	81.2	66.8	51.2	91.9	50.4	81.0	90.5	91.4	16.1	58.5	89.9	67.8	70.0	86.7	**73.7**

boxes as precomputed proposals for COCO-60 and use Selective Search boxes as precomputed proposals for PASCAL VOC. During training, both two networks are optimized by stochastic gradient descent (SGD) with the batch size of 1 and initialized learning rate of 0.001. In each stage, LBBA is trained with 4 epochs, and the learning rate is decayed by 0.1 after 3 epochs. Analogously, WSOD network is trained within 20 epochs and learning rate is decayed by 0.1 after 10 epochs. All programs are implemented by PyTorch toolkit, and all experiments are conducted on a single NVIDIA RTX 2080Ti GPU.

For the multi-label image classifier, we adopt the ADD-GCN [22], which builds a Dynamic Graph Convolutional Network (D-GCN) to model the relation of content-aware category representations generated by a Semantic Attention Module(SAM). During training, the ADD-GCN is optimized by SGD with batch size of 16. The learning rate is initially set to 0.05 for training 40 epoch and decayed by 0.1 to train the latter 10 epoch. The best threshold τ is set to -3.0. By the way, the setting of the τ is based on the implementation of multi-label image classifier. Too high or too low will be detrimental to the final result, and we will give the results and analysis in the next section.

All the source code and pre-trained models will be made publicly available.

C.2. Structure of LBBA

Here we briefly introduce the structure of LBBA. In our solution, we adopt Faster R-CNN [11] with backbone of ResNet-50 [5] as our LBBA. And LBBA is designed to be a class-agnostic bounding box regressor to eliminate potential semantic information leak in bounding box refinement. Note that the inside RPN [11] is only used during EM-like LBBA training to improve the training stabilization and generalization ability of LBBA, and will not be used during the inference stage. We argue that using Faster R-CNN as adjuster has two merits. (i) For the initialization of LBBA training, Faster R-CNN exhibits better performance than Fast R-CNN. (ii) By combining precomputed proposals and proposals from RPN, box regression branch of LBBA can generalize better to various proposals, resulting in more precise box refinement results.

D. More Ablation Studies

D.1. Evaluating LBBA Module Separately

In our solution, LBBA module is designed to be class-agnostic, making that the learned box regressors can be shared among different object classes and transfered to newly added classes. Though we have shown the positive effect of LBBA module in terms of mAP metric, we still evaluate it separately in a manner of proposal evaluation.

Table 5. Detailed comparison of different methods on COCO-20.

Methods	mAP	AP50	AP75	AP_S	AP_M	AP_L	AR_{100}	AR_S	AR_M	AR_L
OICR	9.5	22.8	6.8	2.4	9.4	17.5	24.2	8.0	21.8	38.9
OICR+REG	10.4	23.9	8.1	3.9	9.5	17.8	22.3	7.5	19.3	35.1
Ours LBBA	13.0	27.5	11.2	4.1	12.5	21.4	25.1	8.6	23.3	38.4
Ours LBBA+masking	13.7	29.9	11.5	4.2	13.0	22.1	25.8	8.8	23.9	39.7

Table 6. Experimental settings on auxiliary datasets and target datasets.

Data Settings	Auxiliary Datasets	Target Datasets
Setting 1	COCO-60	PASCAL VOC 2007
Setting 2	COCO-60	PASCAL VOC 2012
Setting 3	COCO-60	COCO-20
Setting 4	ILSVRC-Source	ILSVRC-Target

Therefore we calculate mean IoU between refined proposals from LBBA module and GT boxes. As a comparison, we also calculate mIoU between precomputed proposals and GT boxes as a baseline. IoU performance of LBBA is shown as Table 7. It is clear to conclude that our LBBA module obtains more precise box refinement ability after EM-like LBBA training.

D.2. Performance with ideal LBBA

Our observation is that localization attribute is shared among all kinds of objects, such that a fully supervised box refinement network trained on an auxiliary dataset can be utilized during transfer learning. Therefore, to verify our observation, we build another LBBA-boosted WSOD experiment. During this experiment, we replace pretrained LBBA network by ground-truth bounding box and keep using image class labels to supervise MIL branch, because ground-truth boxes can be seen as an ideal LBBA network to supervise box regression branch of WSOD network during LBBA-boosted WSOD. And then we execute such LBBA-boosted WSOD with the same training schedule. Detection performance of WSOD with ideal LBBA on PASCAL VOC 2007 *test* set is shown as Table 8. Compared to baseline OICR +[12] as well as our proposed LBBA, LBBA-boosted WSOD with ideal LBBA ourperforms by 7.0% on mAP and 2.6% on mAP, respectively. This improvement verifies our observation, and also encourages us to develop more effective adjusters.

D.3. Effect of Masking Strategy for Proposal Classification

Improving the performance of proposal classification usually benefits to improving the overall detection performance of WSOD. Therefore, we also explore the effect of our masking strategy in our LBBA-boosted WSOD network. To demonstrate the effect of the masking strategy, we compared LBBA method with masking strategy with pure LBBA. Table 12 shows the effect of the masking strategy of proposal classification. Compared to pure LBBA with OICR and OICR +[12], our masking strategy improves detection performance by 1.3% and 0.7% mAP on PASCAL VOC 2007 *test* set. We also explore the effect of τ in masking strategy, experimental result is shown as Table 13, we found that $\tau = -3.0$ is the best selection during our masking strategy. Above results indicate that classification predictions from multi-label image classifier are able to select categories with high scores. By suppressing the bounding box scores of non-appearing categories, the proportion of false positives in the final test results is reduced, which is beneficial to improving the overall detection performance of WSOD.

D.4. Is One-class Adjuster Necessary?

During our experiments, to simplify overall experimental settings, we adopt conventional Faster R-CNN [11] with class-agnostic box regression branch as our LBBA fundamental structure, and keep the original RoI classification branch (*e.g.*, 60 classes on COCO-60 dataset). But how the performance of LBBA-boosted WSOD will be changed if we use class-agnostic detector as our LBBA? To solve this question, we train another LBBA whose box regression branch and RoI classification branch are both class-agnostic. And then we execute EM-like LBBA training as well as LBBA-boosted WSOD sequentially using one-class LBBA mentioned above. Performance of LBBA-boosted WSOD supervised by one-class LBBA on PASCAL VOC 2007 is shown as Table 9. Compared to WSOD with our proposed standard LBBA, LBBA with one-class LBBA achieves a slight performance improvement (56.2% mAP vs. 55.8% mAP) on PASCAL VOC 2007 *test* set. However, using conventional LBBA during our experiment is convenient and flexible because each pretrained object detection network can be utilized as a pretrained LBBA directly. Based on this observation, we keep using conventional Faster R-CNN [11] as our LBBA.

D.5. How to update θ_f?

During our LBBA-boosted WSOD in Sec. 3, we use $\{g_0 \dots g_T\}$ with corresponding parameters $\{\theta_g^0 \dots \theta_g^T\}$ to supervise our WSOD network f with θ_f progressively. And to construct a simpler training pipeline, we can directly use the last g_T to supervise f with θ_f. Therefore we are curious about the performance gap between updating θ_f progressively and updating θ_f directly. Corresponding evaluation results are shown as Table 11. The WSOD network up-

Table 7. Per-class mIoU and average mIoU of our LBBA with precomputed proposals. It is clear to conclude that LBBA obtains more precise box refinement ability.

Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	TV	mIoU
Precomputed Proposals	46.1	45.7	45.3	45.3	44.6	46.1	45.7	47.1	45.8	45.6	48.6	46.2	45.8	46.1	45.5	45.0	45.0	47.8	46.9	45.0	45.9
LBBA Module	63.0	54.6	65.5	60.8	60.5	68.3	68.3	69.4	57.4	69.8	57.8	69.0	65.6	58.7	59.3	52.0	66.7	64.5	66.0	68.5	63.2

Table 8. Does ideal LBBA improve performance of WSOD?

Methods	mAP (VOC07)
baseline OICR+[12]	51.4
Ours LBBA	55.8
Ours LBBA (ideal)	58.4

Table 9. Does one-class LBBA improve performance of WSOD?

Methods	mAP (VOC07)
Ours LBBA	55.8
Ours LBBA (one class)	56.2

Table 10. Detailed comparison of different methods on ILSVRC13 Target

Methods	AP50
OICR	20.5
OICR+REG	22.4
LBBA(OICR)	28.0
LBBA(OICR)+masking	30.1

Table 11. Comparison of updating pipeline of f with θ_f (here we set T=3). Evaluation result shows that updating progressively achieves better performance while updating with last g_T achieves a similar performance with only one training stage.

Methods	Stages	mAP (VOC07)
updating progressively	4	55.8
updating with last g_T	1	55.4

dated progressively achieves better performance, while the WSOD network updated with the last g_T achieves a similar performance (-0.4% in terms of mAP on VOC 2007 dataset) with only one training stage. This result indicates that we can build a lighter LBBA-boosted WSOD training pipeline by only using the last g_T in practice, but training progressively is usually stable and better.

E. Comparison with State-of-the-arts

We compare our method with several state-of-the-art WSOD approaches in terms of detection and localization performance on PASCAL VOC datasets. As suggested in [2, 18, 17, 21, 12, 1, 24], we report detection results on *test* set and localization results on *trainval* set, respectively. Table 1 and Table 2 compares the results of different state-of-the-art WSOD approaches on PASCAL VOC 2007 and 2012 datasets. It can be seen that our LBBA improves OICR and OICR+REG over 15.3% and 5.0% on PASCAL VOC 2007 dataset, respectively. Furthermore, our method performs better than all competing methods, except Zhong *et al.* [24]. Note that [24] uses a stronger backbone model and knowledge transfer strategy by directly incorporating source and target datasets. As shown in Fig. 1, our method has the ability to generate precise bounding boxes. On PASCAL VOC 2012, our LBBA is superior to all competing methods and obtains more than 1% gains over all WSOD approaches. Experimental results show that our method is effective in improving the detection performance of WSOD. As shown in Fig. 2, our method also has the ability to generate precise bounding boxes on PASCAL VOC 2012 dataset.

We further evaluate the localization performance of our method. Table 3 and Table 4 lists the results of several state-of-the-art WSOD approaches on PASCAL VOC 2007 and 2012. Our LBBA outperforms OICR by 11.7% and also improves the baseline OICR+REG over 4.3% on PASCAL VOC 2007 dataset. Besides, our LBBA performs better than all competing methods. Meanwhile, on PASCAL VOC 2012, our LBBA is also superior to all competing methods and obtains 1.3% over WSOD 2[23]. In comparison to Zhong *et al.* [24], our LBBA-based method employs a weaker backbone model and avoids the direct joint use of the source and target datasets, while still achieving competitive CorLoc results under the settings of both single-scale testing and multi-scale testing. Above results show that our LBBA-based method is effective in improving the localization performance of WSOD.

F. Generalization to COCO-20

We verify the generalization ability of our LBBA method using a COCO-20 dataset. To this end, we build COCO-20 dataset by collecting the images that only contain instances belonging to the remain 20 classes from *train* and *val* sets of COCO 2017 [10], and use them as the corresponding *train* and *val* sets. Comparing with PASCAL VOC, COCO-20 is more challenging due to more instances and complex layouts. Here we adopt OICR+REG as WSOD network f, and compare with OICR and OICR+REG as baseline methods. We train all models using exactly the same settings in sec. C, and the results are listed in Table 5. Note that our LBBA method with masking strategy outperforms OICR and OICR+REG by 3.5% (4.7%) and 2.6% (3.6%) in terms of mAP and AP50, clearly demonstrating the generalization ability of our LBBA method. After adding masking strategy, our LBBA method outperforms OICR and OICR+REG by 4.2% (7.1%) and 3.3% (6.0%) in terms of mAP and AP50, which demonstrates the effectiveness of our masking strategy.

G. Generalization to ILSVRC-Target

To illustrate that our method can be generalized to more categories, we build the ILSVRC-Target dataset following Appendix B.5 and conduct experiments on it. The base-

Table 12. Effect of Masking Strategy, where *+masking* means our LBBA with masking strategy.

Methods	mAP (VOC07)
LBBA(OICR)	55.1
LBBA(OICR)+masking	56.4
LBBA(OICR+[12])	55.8
LBBA(OICR+[12])+masking	56.5

Table 13. Varying τ for Multi-Label Image Classifier. We evaluated τ on LBBA-Boosted WSOD with OICR head.

τ	mAP (VOC07)
+0.5	55.4
-0.5	55.7
-1.5	56.1
-3.0	**56.4**
-6.0	56.3
-10.0	56.1
-12.0	55.8
-20.0	55.3

line models setting is same as Appendix F and results are listed in Table 10. Note that our LBBA method outperforms OICR and OICR+REG by 7.5% and 5.6% in terms of AP50, which proves that our method can withstand the test of scenes containing more categories of objects. Furthermore, with the enhancement of masking strategy, the performance of WSOD network further outperforms pure LBBA-boosted WSOD by 2.1% in terms of AP50, which shows that masking strategy is able to improve quality of proposal classification and can be generalized to more categories simultaneously.

H. Discussion

In this section, we will discuss our proposed LBBA as well as some modern weakly supervised object detection algorithms in different aspects.

H.1. Discussion of our LBBA

Here we discuss several potential merits of the problem setting and our proposed method. In LBBA-boosted WSOD, the auxiliary well-annotated dataset is not needed and only a smaller amount (*e.g.*, 3) of LBBAs are required. Thus, our problem setting allows deploying LBBAs to versatile weakly annotated datasets for boosting detection performance while avoiding the leakage of well-annotated dataset. In terms of memory consumption, LBBAs are much more economical than the storage of well-annotated dataset.

For the sake of generalization ability, we adopt class-agnostic LBBAs. In comparison to the universal bounding box regressor [7], stage-wise LBBAs are specifically learned to adjust the region proposals generated by WSOD towards the ground-truth bounding boxes, and thus are more effective. To show the generalization ability, the LBBAs learned from well-annotated dataset can be readily deployed to the weakly-annotated dataset with non-overlapped object classes. Nonetheless, LBBAs also work well when the weakly-annotated dataset has the overlapped object classes.

Table 14. Some analysis of Zhong *et al*. in iteration 0. We keep auxiliary dataset and weakly annotated dataset isolated to evaluate performance of Zhong *et al*. fairly.

Methods	mAP (VOC07)
Zhong *et al*. [24] iter 0	54.4
Zhong *et al*. [24] w/o Test-Time Aug iter 0	41.8
Zhong *et al*. [24] w/ COCO-60-full iter 0	∼45

Furthermore, the two subtasks, *i.e*., learning bounding box adjusters and LBBA-boosted WSOD, can be respectively regarded as a kind of knowledge extraction and transfer. With learning bounding box adjusters, we extract the knowledge from the auxiliary well-annotated dataset. Consequently, the extracted knowledge, *i.e*., LBBAs, will be transferred to the WSOD models for improving detection performance. In comparison to directly incorporating auxiliary dataset with weakly-annotated dataset, we argue that the separation of knowledge extraction and transfer is practically more natural, convenient, and acceptable.

H.2. Discussion of *ResNet-WS*

Shen *et al.* [14] proposed a novel residual network backbone architecture, which combines the advantage of residual blocks for feature extraction as well as redundant adaptation neck like *fc6-fc7* of VGG, and leads to better detection performance of the residual network with the weakly supervised setting.

Due to hardware limitations, we did not employ ResNet-WS backbone in our experiments. However, such improvements mainly focus on the backbone of WSOD networks and are able to easily plug into our framework to improve the overall performance of our proposed method. We believe that such method is compatible with ours.

H.3. Discussion of *CASD*

Recently we noticed that Huang *et al.* [6] proposed a novel *Comprehensive Attention Self-Distillation* approach to further improve performance of weakly supervised object detection. This approach obtains higher detection performance than ours and lower localization performance than ours. Similarly, as mentioned in the ablation study, our approach is compatible with various WSOD heads. Naturally, CASD is also compatible. We also believe that the detection performance of WSOD can be better when we apply CASD to our proposed method.

H.4. Discussion of Zhong *et al*.

Zhong *et al*. proposed a novel transfer learning based weakly supervised object detection framework, which utilizes a progressive knowledge distillation training procedure and builds up a universal object proposal generator as well as the corresponding WSOD network.

This method achieves the state-of-the-art detection performance on PASCAL VOC dataset. However, ththisese method exists some difference with our proposed method, which can be listed as follows. First, the Method of Zhong *et al*. proposed a kind of proposal generator while our proposed method is a kind of box refinement network. Second, during EM-like Multi-stage LBBA training as well as LBBA-boosted WSOD, we keep auxiliary dataset and weakly annotated dataset isolated to avoid information leakage of weakly annotated dataset. Finally, after LBBA-boosted WSOD, our WSOD network can generate object detection results individually without help from LBBA.

Besides, the approach of Zhong *et al*. also suffers from *three fundamental limitations during applications*. **First**, when training OCUD in iteration 1 or 2, ground-truth data from auxiliary dataset and pseudo labels from weakly annotated detection dataset are mixed and fed into the OCUD network jointly. As we discussed in Section H.1, this mixture might introduce information leakage of weakly annotated dataset and longer training time in practice.

Second, to improve detection performance during evaluation, predictions from the MIL network of Zhong *et al*. are augmented by adding corresponding objectness scores from OCUD. When removing *Test-Time-Augmentation* (same with using MIL network individually), the performance of Zhong *et al*. drops to 41.8% mAP.

Finally, Zhong *et al*. [24] trains the OCUD on COCO-60-clean dataset which is mentioned in Sec. B.4, and this dataset is easier to learn. Different from [24], we optimize our LBBAs on COCO-60 dataset. For a fair comparison, we evaluate both two methods with the same COCO-60 dataset (containing 98K images) as the auxiliary dataset. When training on our COCO-60 dataset (only removing annotations of VOC classes in COCO dataset) in iteration 0, performance of Zhong *et al*. drops to $\sim$ 45% mAP on PASCAL VOC 2007 *test* set (shown in Table 14). A possible reason is that *the regions with the annotation removed are treated as background in OCUD, which will reduce the recall rate for COCO-60-full*. Compared to Zhong *et al*., our LBBA-boosted WSOD is much more stable with data with noise (see Table 1 for quantitative results).

In conclusion, our method is different from Zhong *et al*., but can be compatible with each other. We believe that the detection performance of WSOD can be better when we apply the method of Zhong *et al*. into our proposed method.

References

[1] Aditya Arun, C.V. Jawahar, and M. Pawan Kumar. Dissimilarity coefficient based weakly supervised object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 3, 4, 6

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. 3, 4, 6

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[4] Ross Girshick. Fast r-cnn. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3, 4

[6] Zeyi Huang, Yang Zou, Vijayakumar Bhagavatula, and Dong Huang. Comprehensive attention self-distillationfor weakly-supervised object detection. In *NeurIPS*, 2020. 7

[7] Seungkwan Lee, Suha Kwak, and Minsu Cho. Universal bounding box regression and its applications. In C.V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, pages 373–387, Cham, 2019. Springer International Publishing. 3, 4, 7

[8] Xiaoyan Li, Meina Kan, Shiguang Shan, and Xilin Chen. Weakly supervised object detection with segmentation collaboration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 4

[9] Yan Li, Junge Zhang, Jianguo Zhang, and Kaiqi Huang. Mixed supervised object detection with robust objectness transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 02 2018. 3, 4

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 6

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 3, 4, 5

[12] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G. Schwing, and Jan Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 4, 5, 6, 7

[13] M. Rochan and Y. Wang. Weakly supervised localization of novel objects using appearance transfer. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4315–4324, 2015. 4

[14] Yunhang Shen, Rongrong Ji, Yan Wang, Zhiwei Chen, Feng Zheng, Feiyue Huang, and Yunsheng Wu. Enabling deep residual networks for weakly supervised object detection. In *European Conference on Computer Vision (ECCV)*, 2020. 7

[15] Yunhang Shen, Rongrong Ji, Yan Wang, Yongjian Wu, and Liujuan Cao. Cyclic guidance for weakly supervised joint detection and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

Figure 1. More visualization results of our method on PASCAL VOC 2007, which has the ability to generate precise bounding boxes.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *In-*

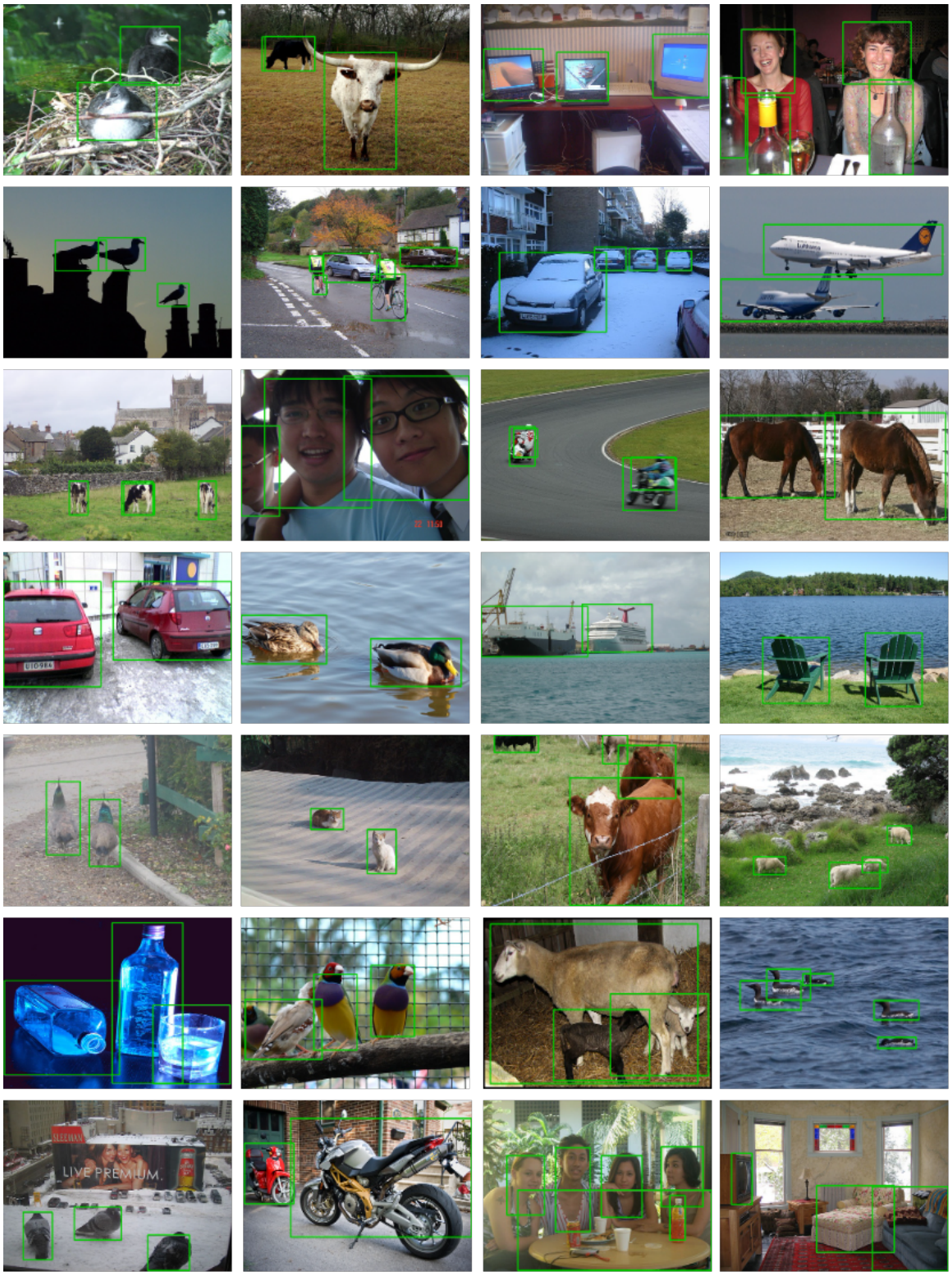

Figure 2. More visualization results of our method on PASCAL VOC 2012, which has the ability to generate precise bounding boxes.

ternational Conference on Learning Representations, 2015.

[17] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):176–191, 2018. 3, 4, 6

[18] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017. 3, 4, 6

[19] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-mil: Continuation multiple instance learning for weakly supervised object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 4

[20] G. Yan, B. Liu, N. Guo, X. Ye, F. Wan, H. You, and D. Fan. C-midn: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9833–9842, 2019. 3

[21] Ke Yang, Dongsheng Li, and Yong Dou. Towards precise end-to-end weakly supervised object detection network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8372–8381, 2019. 3, 4, 6

[22] J. Ye, Junjun He, Xiaojiang Peng, Wenhao Wu, and Y. Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *ECCV*, 2020. 4

[23] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3, 4, 6

[24] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 615–631, Cham, 2020. Springer International Publishing. 3, 4, 6, 7, 8