MaskBooster: End-to-End Self-Training for Sparsely Supervised Instance Segmentation

Shida Zheng*, Chenshu Chen*†, Xi Yang, Wenming Tan†

Hikvision Research Institute
{zhengshida,chenchenshu,yangxi6,tanwenming}@hikvision.com

Abstract

The present paper introduces sparsely supervised instance segmentation, with the datasets being fully annotated bounding boxes and sparsely annotated masks. A direct solution to this task is self-training, which is not fully explored for instance segmentation yet. In this paper, we propose MaskBooster for sparsely supervised instance segmentation (SpSIS) with comprehensive usage of pseudo masks. MaskBooster is featured with (1) dynamic and progressive pseudo masks from an online updating teacher model, (2) refining binary pseudo masks with the help of bounding box prior, (3) learning inter-class prediction distribution via knowledge distillation for soft pseudo masks. As an end-to-end and universal self-training framework, MaskBooster can empower fully supervised algorithms and boost their segmentation performance on SpSIS. Abundant experiments are conducted on COCO and BDD100K datasets and validate the effectiveness of MaskBooster. Specifically, on different COCO protocols and BDD100K, we surpass sparsely supervised baseline by a large margin for both Mask RCNN and ShapeProp. MaskBooster on SpSIS also outperforms weakly and semi-supervised instance segmentation state-of-the-art on the datasets with similar annotation budgets.

Introduction

The rapid development of instance segmentation is fueled by large annotated datasets. However, obtaining fine-grained segmentation masks is time- and labor-consuming, which limits the practical application of instance segmentation.

To reduce the reliance on fully annotated datasets, different kinds of supervision for instance segmentation have been explored. Weakly supervised instance segmentation (WSIS) includes image-level label (Huang et al. 2018; Joon Oh et al. 2017), point (Cheng, Parkhi, and Kirillov 2022), scribble (Lin et al. 2016; Vernaza and Chandraker 2017) and bounding box supervision (Khoreva et al. 2017; Lee et al. 2021; Hsu et al. 2019; Tian et al. 2021). Note that WSIS in the following paper refers in particular to to bounding box supervision, which only uses ground truth bounding boxes (GT boxes) without any ground truth mask (GT mask). Semi-supervised instance segmentation (SSIS) includes two splits of images, one split with fully annotated bounding boxes and masks, while the other with neither of them. In partial supervision (PSIS), every object has an GT box, but only objects of specific categories have GT masks. Here we introduce *sparsely supervised instance segmentation* (SpSIS), for which all objects have GT boxes, and a few of them have GT masks. The comparison of different supervision for instance segmentation is shown in Fig. 1. An SpSIS dataset can be created by annotating some masks on the basis of an easily-accessible detection dataset and the present study for SpSIS aims to find out algorithms that can outperform WSIS and SSIS under similar annotation budgets.

The pseudo-mask-based method can be a solution for SpSIS. It generates pseudo masks for objects without GT

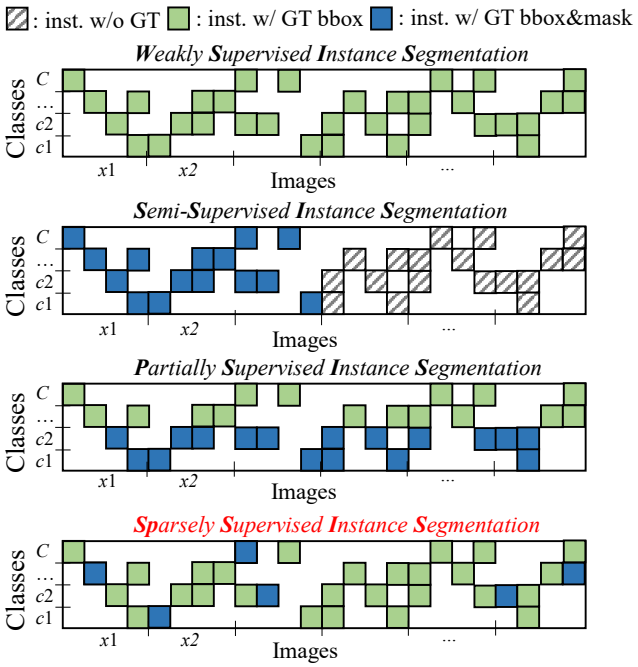

Figure 1: Illustration for instance segmentation datasets in different types of supervision. The x-axis and y-axis respectively represent images and category indexes. For each small square, its position indicates its belonging category and image, with its color representing its annotation information.

*These authors contributed equally.
†Corresponding authors.

masks, and then trains a model in a fully supervised manner. Kulharia et al. (2020), Dai, He, and Sun (2015) and Song et al. (2019) try to generate pseudo masks by classic GrabCut (Rother, Kolmogorov, and Blake 2004), MCG (Pont-Tuset et al. 2016) or CRF (Krähenbühl and Koltun 2011). These complicated methods suffer from the low quality of pseudo masks, leading to dissatisfying performance. Recently, self-training is prevalent in semi-supervised image classification and object detection, yet is not much explored for instance segmentation.

In this paper, we propose a self-training paradigm for SpSIS, named MaskBooster. First, MaskBooster is an end-to-end framework with dynamically updating pseudo masks during training, which avoids the tedious procedure of multi-stage training. When GT masks are too rare, end-to-end training probably results in model collapse, so we design the loss re-weighting for pseudo masks and make MaskBooster converge well under this circumstance. Second, we take advantage of GT boxes to help suppress pixels of binary pseudo masks out of corresponding bounding boxes. Third, we introduce soft pseudo masks, so the model can distill richer information than binary pseudo masks with information lost due to slicing and thresholding. We discover that there is an inherent inter-class relationship between class-aware mask predictions, and apply knowledge distillation (Hinton et al. 2015) to learn this relationship implicitly. Moreover, our method is compatible and can be regarded as a wrapper for most instance segmentation algorithms to boost their segmentation performance, hence termed MaskBooster.

Attributed to the ideas above, MaskBooster achieves significantly better performance than sparsely supervised baselines on various datasets, including COCO (Lin et al. 2014) 0.1%/1%/10% protocols and BDD100K (Yu et al. 2020). Mask RCNN (He et al. 2017), ShapeProp (Zhou et al. 2020) and CondInst (Tian, Shen, and Chen 2020) are chosen as the base algorithms to prove our compatibility.

The contributions of this paper are as follows.

- An end-to-end self-training framework known as MaskBooster is proposed. It benefits from teacher-student mutual learning and dynamic pseudo masks. With the help of the loss re-weighting, MaskBooster can tackle with extremely rare GT masks.
- In MaskBooster, both binary and soft pseudo masks are introduced to train the student. We use the bounding box prior to refine the binary pseudo masks, and apply knowledge distillation for soft pseudo masks according to the finding of the inter-class relationship.
- Effectiveness and compatibility of MaskBooster are verified by extensive experiments on various protocols of COCO and BDD100K and on different base models, such as Mask RCNN, ShapeProp and CondInst. MaskBooster also demonstrates superiority over state-of-the-art WSIS and SSIS algorithms under similar annotation budgets.

Related Work

Self-Training in Semi-Supervised Learning

Self-training is popular in semi-supervised learning and has been widely researched in image classification (Xie et al. 2020; Arazo et al. 2020; Sohn et al. 2020a) and object detection (Sohn et al. 2020b; Zhou et al. 2021; Tang et al. 2021). In semi-supervised image classification, FixMatch (Sohn et al. 2020a) generates pseudo labels on weakly augmented images and matches them to the strongly-augmented counterparts. CReST (Wei et al. 2021) proposes a class-rebalancing self-training method, where pseudo labels are specially selected from an estimated class distribution. SSWPL (Taherkhani et al. 2021) provides pseudo labels based on the minimal Wasserstein distance to a labeled-data cluster. In semi-supervised object detection, STAC (Sohn et al. 2020b) puts forward a neat self-training framework and surpasses the consistency-based method (Jeong et al. 2019). And subsequent works improve the self-training method from different aspects, *e.g.*, dynamic pseudo-labeling (Zhou et al. 2021; Liu et al. 2021), soft pseudo labels (Tang et al. 2021), quality estimation for pseudo labels (Li et al. 2022a), end-to-end training (Xu et al. 2021), class-balanced sampling (Zhang, Pan, and Wang 2022) and dual-head denoise training (Zheng et al. 2022). However, there are few self-training methods for instance segmentation, which motivates us to find out a competitive self-training solution for SpSIS.

Instance Segmentation with Icomplete Supervision

WSIS hereby refers to box-supervised instance segmentation without any mask annotation. SDI (Khoreva et al. 2017) generates pseudo masks by MCG (Pont-Tuset et al. 2016) and refines segmentation results by an iterative training procedure. BBTP (Hsu et al. 2019) employs multiple instance learning to utilize the bounding box annotations. DiscoBox (Singh et al. 2021) incorporates pairwise and cross-image potential among boxes with a structured energy model. BoxInst (Tian et al. 2021) introduces a projection loss for the full usage of box annotation and a color-space pairwise loss for finer mask segmentation. Despite good mask performance, BoxInst heavily relies on CondInst (Tian, Shen, and Chen 2020), thus is less compatible with other algorithms. SSIS pays attention to the training strategy for the union of fully labeled images and raw images without any annotation. Wang, Li, and Wang (2022) propose a noise-tolerant mask head and a boundary-preserving map for learning coarse and fine information, respectively. In PSIS, Mask^X RCNN (Hu et al. 2018) learns a weight transfer function from boxes to masks on a small set of fully annotated data. ShapeMask (Kuo et al. 2019) collects shape prior knowledge from available mask annotations. In CPMask (Fan et al. 2020), shapes and appearance commonalities are learned by predicting the instance boundary. ShapeProp (Zhou et al. 2020) learns saliency propagation by the message passing module. OPMask (Biertimpel et al. 2021) highlights correct foreground areas via class activation maps.

SpSIS introduced in this paper is more appropriate for real-world applications than WSIS, since the cost of annotating small amount of masks is acceptable. SpSIS also enjoys

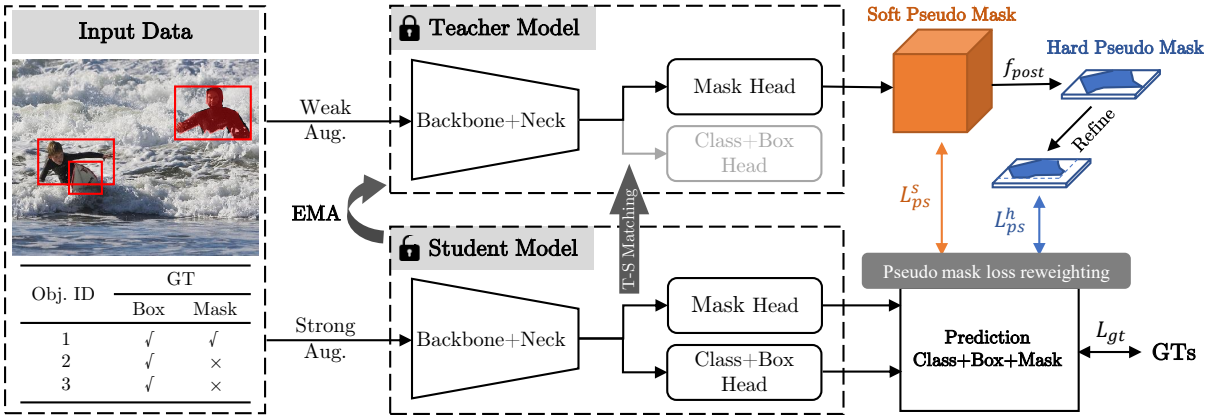

Figure 2: MaskBooster Overview. Weakly augmented data are fed to the EMA teacher for pseudo mask generation and strongly augmented data are fed to the student for training. T-S matching ensures that pseudo masks match corresponding predicted masks. Both soft and binary pseudo masks are involved in loss calculation (L^s_{ps} and L^h_{ps}, respectively). L_{gt} is the loss from ground-truths. Losses from pseudo masks are rescaled by a re-weighting factor. f_{post} represents the activate-slice-threshold procedure. Best viewed in color.

higher cost performance than SSIS under similar annotation budgets. According to Lin et al. (2014) and Papadopoulos et al. (2017), a polygon-based object mask takes 79.2 seconds on average, whereas a bounding box only takes about 7 seconds. Therefore, we can estimate that the time cost of COCO 10% of SSIS in Noisy Boundary (Wang, Li, and Wang 2022) is about 85 days per annotator, which is comparable with COCO 1% of SpSIS in this paper (about 78 days). The experimental section will validate the effectiveness of SpSIS quantitatively and expand more details.

Method

Preliminary

We denote the SpSIS dataset as $\{\mathbf{X}_i, \mathbf{C}_i, \mathbf{B}_i, \mathbf{Q}_i, \mathbf{I}_i\}_i$, where $\mathbf{X}_i \in \mathbb{R}^{H_i \times W_i \times 3}$, $\mathbf{C}_i \in \{1, ..., C\}^{N_i}$, $\mathbf{B}_i \in \mathbb{R}^{N_i \times 4}$ and $\mathbf{Q}_i \in \{0,1\}^{N_i \times H_i \times W_i}$ are the input image, category labels, GT boxes and GT masks. $\mathbf{I}_i \in \{0,1\}^{N_i}$ represents the mask annotation indicator, where $\mathbf{I}_i^{(j)} = 1$ means that the j-th instance has a GT mask. $\mathbf{Q}_i^j$ is regarded as pseudo mask tensor if $\mathbf{I}_i^{(j)} = 0$. C is the number of categories. H_i, W_i and N_i are the height, width and number of instances of the i-th image. Naturally, $N = \Sigma_i N_i$ is the total number of instances and $N' = \Sigma_i \Sigma_j \mathbf{I}_i^j$ is the number of instances with GT masks. We define $\rho = N'/N$ as the ratio of GT masks in a dataset. Both one-stage and two-stage algorithms can be equipped with our method, and analysis below will base on Mask RCNN unless otherwise specified. The subtitle i will be ignored in following notations for the sake of simplicity.

Self-Training: From Static to Dynamic

The pseudo mask is the core of self-training. Off-the-shelf GrabCut (Rother, Kolmogorov, and Blake 2004) and MCG (Pont-Tuset et al. 2016) are alternatives to generate pseudo masks from box annotations. However, these pseudo masks suffer from poor pixel-level precision, and no longer update during self-training (*i.e.*, are static), thus the final performance is unsatisfactory. Though SDI (Khoreva et al. 2017) refines the pseudo masks by an iterative training procedure, the multi-stage algorithm is complicated and pseudo masks are still static during a training period.

In this paper, we change the static pseudo masks into dynamic ones to overcome the above-mentioned problems and propose an end-to-end self-training framework, MaskBooster. The overview is shown in Fig. 2. Specifically, MaskBooster consists of a teacher model θ_t and a student model θ_s. In each iteration, the teacher generates pseudo masks to train the student and the teacher updates its parameters via exponential moving average (EMA). As the student converges during training, the teacher also converges (Ke et al. 2019). The better teacher produces better pseudo masks, which further improves the student. The mutual learning procedure can be formulated as follows:

$$\theta_s \leftarrow \theta_s + \gamma \frac{\partial \mathcal{L}}{\partial \theta_s} \tag{1}$$

$$\theta_t \leftarrow (1-\alpha)\theta_t + \alpha\theta_s, \alpha \in (0,1) \tag{2}$$

where γ is the learning rate and α is the EMA ratio, respectively. $\mathcal{L}$ is the total loss for optimization. It contains supervised losses for classification, localization and segmentation (denoted as $\mathcal{L}^{cls}_{gt}$, $\mathcal{L}^{loc}_{gt}$ and $\mathcal{L}^{seg}_{gt}$, respectively), as well as pseudo-mask losses for those instances without mask annotations (denoted as $\mathcal{L}^{seg}_{ps}$):

$$\mathcal{L} = \mathcal{L}^{cls}_{gt} + \mathcal{L}^{loc}_{gt} + \mathcal{L}^{seg}_{gt} + \mathcal{L}^{seg}_{ps} \tag{3}$$

where $\mathcal{L}^{seg}_{ps}$ includes the binary pseudo mask loss L^h_{ps} and the soft pseudo mask loss L^s_{ps}, which will be elaborated on in the following sections.

It is worth explaining that the proposed mutual learning scheme is non-trivial. Firstly, we design the pseudo mask loss re-weighting strategy in order to avoid the model collapse when GT masks are very limited. It rescales the pseudo

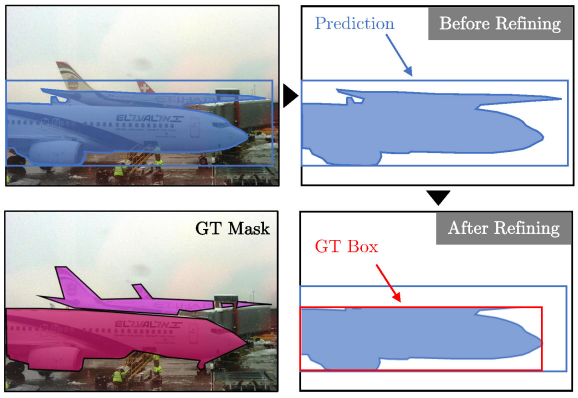

Figure 3: Illustration for refining binary pseudo masks by GT box prior. The foreground pixels outside the GT box are set as background.

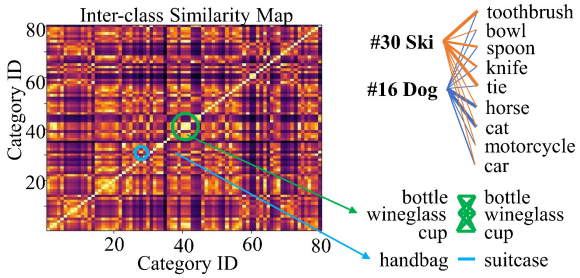

Figure 4: Heatmap of inter-class prediction similarity on COCO *train2017* and its examples. Similarity value in the figure is log-scaled for better visualization. Different line colors indicate different relationships, like appearance, shape and context. A thicker line indicates a stronger relationship between two categories.

mask losses and increases as follows,

$$r(t) = -0.5 \cos \frac{t}{t_e}\pi + 0.5 \tag{4}$$

where t is the training iteration, t_e is the max iteration for loss re-weighting to take effect, and r represents the pseudo mask loss re-weighting factor.

Secondly, we design a T-S Matching strategy that the teacher uses proposals from the student, hence every pseudo mask matches its corresponding student's prediction directly.

$$\mathbf{P}_s = f(x;\theta_s); \mathbf{Q} = f(x;\mathbf{P}_s,\theta_t) \tag{5}$$

x is the input image and $\mathbf{Q}$ is the pseudo mask. $\mathbf{P}_s$ represents the proposals from the student. The proposal is box-like in two-stage algorithms (*e.g.*, Mask RCNN), and point-like in one-stage algorithms (*e.g.*, CondInst). T-S Matching speeds up training period significantly, because the student can directly calulate losses between predictions and the matched targets without GT assignment. Thirdly, we apply asymmetric augmentations to input data. Strongly augmented images are the input of the student, improving the teacher with the diversity of students (Tarvainen and Valpola 2017; Xie et al. 2020). Weakly augmented images are the input of the teacher for reliable and stable pseudo masks.

Box Prior: Binary Mask Refining

Dynamic pseudo labeling improves self-training method in the temporal dimension, and there is still room for the improvement in the space dimension. In MaskBooster, the teacher generates pseudo masks based on the student's proposals due to our T-S Matching mechanism, but the GT boxes are absent. Therefore, we propose a simple yet effective method to well use the GT box prior, so as to refine binary pseudo mask.

In Mask RCNN, there are non-overlapping areas between a proposal and its matched GT box. As the mask prediction comes from the RoI feature extracted by the proposal, the predicted foreground in the non-overlapping area is definitely wrong. Consequently, we force the foreground predicted pixels to be background if they are out of the GT box. Fig. 3 illustrates the refining procedure. For one-stage instance segmentation frameworks, points play the role of proposals, and the proposed box prior still works.

For implementation, we construct a masking tensor as follows:

$$\mathbf{M} = (m_{nhw})_{NHW}, \ m_{nhw} = \begin{cases} 1, & if\ (h,w)\ inside\ \mathbf{B}^{(n)} \\ 0, & otherwise \end{cases} \tag{6}$$

By calculating Hadamard product between the pseudo mask tensor and the masking tensor, we obtain the refined pseudo masks:

$$\mathbf{Q} \leftarrow \mathbf{Q} \circ \mathbf{M} \tag{7}$$

Inter-Class Relationship: From Binary to Soft

For a class-aware mask predictor, there are C neurons responsible for predicting foreground masks for C categories each, but only one mask result is extracted according to the predicted category index. We find that different channels of mask logits from similar categories are likely to activate similar regions. We formulate the mean discrepancy between all pairs of categories in a mask logit tensor $\mathbf{P}$ as follows:

$$\mathbf{D} = (d_{ij}) \in \mathbb{R}^{C\times C}, \ d_{ij} = \frac{1}{K}\sum_{k}^{K}\sum_{h,w}(\mathbf{P}_{kihw} - \mathbf{P}_{kjhw})^2 \tag{8}$$

And the similarity is

$$\mathbf{S} = 1 - Norm(\mathbf{D}) \tag{9}$$

where $Norm(\cdot)$ is the min-max normalization. To make our similarity more statistically significant, we calculate the mean similarity for all instances in all images in COCO *train2017* and finally draw the relationship, as shown in Fig. 4. Both the x-axis and the y-axis represent the category index from 1 to $C(=80)$. The value on the position (x, y) is the mask prediction similarity between category x and category y. The brighter the region, the more similar the mask predictions for the two categories. Obviously, similarity values on the principal diagonal are all 1. Fig. 4 also gives some examples for better illustration. As a thicker line represents higher

similarity, we can see that there are various characteristics of similarity, such as the slim shape ("ski", "tie", "knife", ...), the furry appearance ("dog", "cat") and the context ("bottle", "cup", "wineglass").

This finding reveals that some useful knowledge, *e.g.*, the inter-class relationship, is missing in the binary pseudo mask. And binary mask comes from mask logits:

$$\mathbf{Q} = Binarize(Slice(Sigmoid(\mathbf{P}))) \triangleq f_{post}(\mathbf{P}) \quad (10)$$

where $\mathbf{P} \in \mathbb{R}^{K\times C\times 28\times 28}$ and $\mathbf{Q} \in \{0,1\}^{K\times 28\times 28}$ are respectively mask logits and binary pseudo masks from the teacher. K is the number of positive proposals. Clearly, $f_{post}(\cdot)$ drops out some information. In order to utilize the missing information, we regard mask logits as soft pseudo masks. Conventionally, soft pseudo masks need saving offline and require considerable storage space, which is unacceptable for a large-scale dataset. Thanks to our framework, mask logits can be produced and calculated losses online without redundant read-write operations.

In order to take advantage of the inter-class relationship, grouping similar categories during training may serve as a feasible approach. For example, mask prediction neurons for "dog", "horse" and "cat" will all be trained if the predicted category is "dog". In fact, this characteristic of mask logits reflects the distribution of foreground confidence for all categories, so we refer to knowledge distillation for optimization. Specifically, we convert logits from both the teacher and the student into probabilities via softmax function with a temperature, which is denoted as $p(\cdot;\mathcal{T})$. A higher value for $\mathcal{T}$ leads to a softer probability distribution. Then, we calculate Kullback-Leibler divergence between the two probabilities as the soft-label loss.

$$\mathcal{L}^s_{ps} = \mathbf{D}_{K||L}(\ p(\mathbf{P};\mathcal{T})\ ||\ p(\tilde{\mathbf{P}};\mathcal{T})\ ) \quad (11)$$

where $\tilde{\mathbf{P}}$ is the mask logits predicted from the student. By combining the soft mask loss with the binary mask loss, the total loss for pseudo masks is:

$$\mathcal{L}^{seg}_{ps} = r * (\lambda_h \mathcal{L}^h_{ps} + \lambda_s \mathcal{L}^s_{ps}) \quad (12)$$

where λ_h and λ_s are respectively the loss weights for balancing the effect of binary and soft pseudo masks.

Wrapper: From Full to Incomplete Supervision

Unlike ShapeProp and Noisy Boundary, MaskBooster does not change the model structure, and is compatible for various instance segmentation algorithms. It can be regarded as a wrapper for most instance segmentation models and can convert a fully supervised algorithm $\mathcal{F}_{sup}$ into a sparsely supervised one $\mathcal{F}_{sparse}$:

$$\mathcal{F}_{sparse} = MaskBooster(\mathcal{F}_{sup}) \quad (13)$$

In the following experiments, we reform the Mask RCNN (He et al. 2017), ShapeProp (Zhou et al. 2020) and CondInst (Tian, Shen, and Chen 2020) to verify our compatibility. Also, MaskBooster can empower PSIS algorithms like OPMask, leading to better novel-class performance.

ratio	Method	AP	AP_{50}	AP_{75}	AP_S	AP_L
	Fully Supervised					
100%	Mask RCNN	37.1	58.7	39.8	18.0	53.8
	Shapeprop	38.4	59.6	41.3	19.1	55.1
	Sparsely Supervised					
0.1%	Mask RCNN	25.4	49.5	23.6	13.6	36.2
	+MaskBooster	**31.1**	54.7	31.4	16.2	43.6
	Shapeprop	28.4	52.7	27.7	14.8	39.9
	+MaskBooster	**33.3**	56.4	34.6	16.8	48.0
1%	Mask RCNN	31.2	53.7	32.3	15.5	44.9
	+MaskBooster	**34.1**	56.9	35.6	17.4	48.6
	Shapeprop	31.7	53.7	32.8	14.3	46.5
	+MaskBooster	**34.7**	56.4	36.8	17.5	49.9
10%	Mask RCNN	34.6	56.1	36.7	16.8	49.8
	+MaskBooster	**36.2**	58.6	38.5	18.9	52.1
	Shapeprop	34.2	55.0	36.3	15.4	50.8
	+MaskBooster	**36.7**	57.9	39.6	18.6	52.5

Table 1: Results on different SpSIS protocols of COCO.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_L
Mask RCNN	27.2	47.0	26.3	14.8	48.0
+MaskBooster	**27.9**	48.4	26.9	15.1	55.4
Shapeprop	27.9	47.7	28.5	15.6	55.9
+MaskBooster	**29.1**	48.4	30.2	16.3	57.5

Table 2: Results on BDD100K ($\rho \approx 11\%$)

Experiments

Implementation Details

We implement MaskBooster on three instance segmentation algorithms: Mask RCNN, ShapeProp and CondInst. The backbone network is ResNet-50 (He et al. 2016). The optimizer we use is SGD with a momentum of 0.9. EMA ratio is set as $\alpha = 1e-3$. The loss weights for pseudo masks are $\lambda_h = 1$ and $\lambda_s = 10$. For COCO 0.1%, due to the extremely limited GT masks, we set $\lambda_h = 0.2$ and apply CopyPaste (Ghiasi et al. 2021). All experiments are under *multi-scale* and *3*$\times$ training schedule. Other settings follow the MMDetection toolbox (Chen et al. 2019).

Note that instance segmentation models are able to train on SpSIS datasets as long as they ignore the pseudo mask loss. We call it *sparsely supervised baseline* in the experiments below for comparison.

Benchmark Datasets

COCO To fully assess an approach of SpSIS, we randomly sample a ratio of ρ instances in COCO *train2017* keeping GT masks while removing GT masks for the rest of the instances. Three datasets are constructed: COCO 0.1%, COCO 1% and COCO 10% with $\rho = 0.1\%/1\%/10\%$, which have 880, 8656 and $\sim$ 86k GT masks, respectively. These datasets can reflect the performance of algorithms given different ratios of mask annotations.

Method	voc → non-voc						non-voc → voc					
	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
OPMask†	28.5	48.0	29.6	14.5	31.4	41.5	36.2	60.8	37.4	17.2	35.4	49.3
+MaskBooster†	**30.2**	50.1	30.7	16.4	32.2	43.2	**37.6**	61.6	39.0	18.2	37.1	51.3
Shapeprop	33.0	53.2	34.9	16.4	37.1	47.6	37.7	62.8	39.2	17.7	36.7	51.9
+MaskBooster	**34.4**	54.7	36.7	18.3	38.6	49.6	**38.2**	63.9	39.6	19.1	37.5	52.2

Table 3: Results on partial-COCO. †: The training schedule follows OPMask (Biertimpel et al. 2021).

Method	Budget	AP	AP_{50}
Weakly Supervised			
CondInst+BoxInst	7.8	31.3	54.1
SoloV2+DiscoBox	7.8	31.4	52.6
Li et al. (2022b)	7.8	31.4	-
Semi-Supervised			
Noisy Boundary (5%)	4.7	24.9	46.1
Noisy Boundary (10%)	9.4	29.2	51.7
Sparsely Supervised			
CondInst (0.1%)	7.9	28.7	52.7
+MaskBooster (0.1%)	7.9	**33.1**	**55.4**
Mask RCNN (0.1%)	7.9	25.4	49.5
+MaskBooster (0.1%)	7.9	**31.1**	**54.7**
Mask RCNN (1%)	8.7	31.2	53.7
+MaskBooster (1%)	8.7	**34.1**	**56.9**

Table 4: Comparison with WSIS and SSIS algorithms. Note that Noisy Boundary is based on Mask RCNN.

In COCO 0.1%, the only difference from box-supervised WSIS and SpSIS is the 880 GT masks, which enables the comparison between WSIS and limited-budget SpSIS. And in our COCO 1%, the annotation budgets is similar with those of COCO 10% in SSIS (Wang, Li, and Wang 2022), which is also a fair comparison.

BDD100K BDD100K has about 6,887 images with about 89,098 GT masks, and 66,445 images with 821,020 GT boxes, which is a SpSIS dataset defined in aforementioned sections. We benchmark our MaskBooster upon BDD100K and compare the mask performance with sparsely supervised baseline of Mask RCNN and ShapeProp (Zhou et al. 2020).

Partial-COCO We also evaluate MaskBooster on PSIS datasets, which is called partial-COCO for convenience. We follow the setup in the existing PSIS works (Hu et al. 2018; Zhou et al. 2020; Biertimpel et al. 2021) and split the mask annotations of the 80 classes into a 20-class subset ("voc") of the Pascal VOC and a remaining 60-class subset ("non-voc") for training, and COCO *val2017* for evaluation.

Main Results

Comparison with sparsely supervised baseline In Tab. 1, we compare three different GT mask ratios and two different models on COCO SpSIS datasets. MaskBooster achieves significantly higher performance than sparsely supervised baseline with all three GT mask ratios. Especially for COCO 0.1%, MaskBooster gets +5.7 and +4.9 mask AP higher than the baseline Mask RCNN and ShapeProp, respectively. Given the SpSIS dataset with the 10% GT mask ratio, MaskBooster narrows the performance gap of Mask RCNN and ShapeProp to 0.9 and 1.7 AP compared with the fully supervised upper bound. In Tab. 2, we evaluate MaskBooster on another dataset, BDD100K. MaskBooster still gains extra performance with different baseline models. It shows that MaskBooster works well on datasets with different domains and scenes, such as the autonomous driving dataset.

Experiments on PSIS datasets In Tab. 3, two state-of-the-art PSIS algorithms are chosen and wrapped with MaskBooster, the results show that MaskBooster can also gain extra performance for the PSIS task.

Comparison with WSIS and SSIS Tab. 4 compares the SpSIS algorithm, MaskBooster, with other WSIS and SSIS algorithms. We choose the recent BoxInst as a WSIS baseline and Noisy Boundary as a SSIS baseline. Before analyzing the results, we first clarify what the "budget" means. According to former studies (Lin et al. 2014; Papadopoulos et al. 2017; Cheng, Parkhi, and Kirillov 2022), a mask is annotated about 11 times slower than a bounding box, so we define the time cost for annotating $10k$ masks as 1 unit time (*i.e.*, 1 budget), and the time cost for annotating $10k$ bounding boxes is 1/11 unit time. For COCO *train2017* with 860k objects, the total budgets are $86+86/11 = 93.8$, which is reduced to $86*0.001+86/11 = 7.9$ for COCO 0.1% of SpSIS, as listed in Tab. 4. Compared with WSIS algorithm BoxInst, CondInst+MaskBooster achieves +1.8 AP higher with only 0.1 extra annotation budgets. The improvement mostly comes from MaskBooster, because the mask AP drops from 33.1 to 28.7 without MaskBooster. As for SSIS, the annotation budget of COCO 10% of SSIS is similar to COCO 1% of SpSIS. MaskBooster trained on COCO 1% of SpSIS (with budget=8.7) outperforms Noisy Boundary trained on COCO 10% of SSIS (with budget=9.4) by +4.2 mask AP. Even when the GT mask ratio is only 0.1% (880 GT masks), MaskBooster is still better than Noisy Boundary.

Ablation Study

Pseudo mask loss re-weighting Tab. 5 shows the experimental results with the re-weighting length t_e varying from $1k$ to $40k$ and the GT mask ratio varying from 0.1% to 10%.

Dataset	$1k$	$10k$	$20k$	$40k$
COCO 0.1%	18.1	28.8	31.1	31.2
COCO 1%	33.6	34.1	34.0	34.0
COCO 10%	36.3	36.2	36.2	36.2

Table 5: Results on different re-weighting lengths t_e and datasets with different ratios of GT masks.

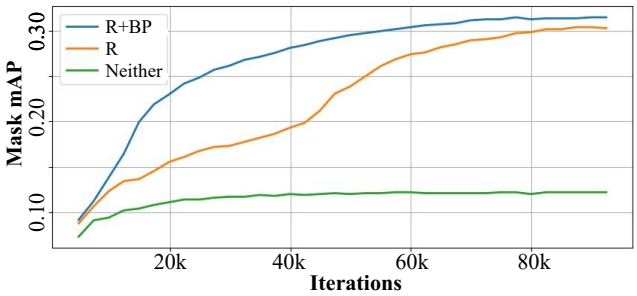

Figure 5: The mask AP curve during training. BP: with box prior. R: with pseudo mask loss re-weighting. All three curves are from the training logs on COCO 0.1%.

EMA+Aug	Binary	Soft	0.1%	1%	10%
			25.4	31.2	34.6
✓			26.4	31.8	34.9
✓	✓		28.0	33.9	36.0
✓		✓	28.8	32.3	35.1
✓	✓	✓	**31.1**	**34.1**	**36.2**

Table 6: Ablate three components in MaskBooster. Each ablation is on 0.1%, 1% and 10% protocols of SpSIS.

Figure 6: Visualization results on COCO *val2017*. The top row is from baseline model and the bottom row is from MaskBooster. The training dataset is COCO 0.1%.

For COCO 0.1%, the performance drops a lot when the re-weighting length is not long enough (*i.e.* $1k$ and $10k$). For COCO 1%, $t_e = 10k$ is a proper setting yet $1k$ leads to -0.5 mAP. We can conclude that an appropriate re-weight length is related to the ratio of GT masks. The fewer the GT masks, the more important the re-weighting. Besides, $t_e = 40k$ is suitable for all of the three datasets, so it is reasonable that t_e can be set long enough by default for training a new dataset.

Fig. 5 shows the performance trend during training on COCO 0.1% of SpSIS. Without loss re-weighting (the green curve), the poor quality of pseudo mask interferes the convergence, thus the mask AP increases slowly and ends with a bad result. With the proposed pseudo mask loss re-weighting (the orange curve shows), the model converges faster and gets better final performance.

GT box prior The effectiveness of the proposed GT box prior is showed In Fig. 5, whose AP curve is in blue. When using box prior, the training becomes more steady and the final result reaches higher performance.

Decompose MaskBooster We decompose MaskBooster into three components: EMA+Aug strategy, binary pseudo masks with box prior and soft pseudo masks. Tab. 6 gives the results for the ablation study. The first row is the result of sparsely supervised baseline. The experiment of the second row adds EMA and strong augmentation (EMA+Aug) to the baseline. The last three rows are of different combinations of pseudo masks based on our framework. We can find that the EMA+Aug achieves some improvement, but the optimized binary pseudo mask and the optimized soft pseudo mask boost the performance considerably. With the union of all the three components in MaskBooster, the performance respectively increases +5.7, +2.9 and +1.6 mask AP for the 0.1%, 1% and 10% GT mask ratios, and validates the effectiveness of our method. Moreover, MaskBooster can significantly improve the performance from 25.4 to 31.1 mask AP, when the annotated masks are extremely rare, which is a challenge to the SpSIS task.

Visualization

We visualize the predictive bounding boxes and masks for both the baseline Mask RCNN and its MaskBooster-wrapped version, as shown in Fig. 6. The baseline Mask RCNN only uses 880 GT masks (COCO 0.1%) to train its mask head. In contrast, MaskBooster uses progressive pseudo masks, thus obtaining finer segmentation results and recalling more objects than the baseline Mask RCNN.

Conclusion

In this paper, we introduce SpSIS, a new type of supervision for instance segmentation. To enhance pseudo-mask-based self-training method on SpSIS datasets, we propose a strong and universal end-to-end framework, MaskBooster. In MaskBooster, we consider three aspects of pseudo masks: progressive, binary and soft. Each aspect corresponds with a carefully designed approach pushing MaskBooster a step further, including dynamically pseudo labeling with the indispensable pseudo mask loss reweighting, binary pseudo mask refining via the box prior, and soft pseudo masks via knowledge distillation. Extensive experiments with different base segmentation models validate our strong performance and compatible framework. Compared with WSIS and SSIS, MaskBooster trained on SpSIS demonstrates superiority given similar annotation budgets. In the future work, a more universal supervision can be investigated, which has incomplete annotations for both bounding boxes and masks. And we believe our method has the potential to address it.

References

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Biertimpel, D.; Shkodrani, S.; Baslamisli, A. S.; and Baka, N. 2021. Prior to Segment: Foreground Cues for Weakly Annotated Classes in Partially Supervised Instance Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2824–2833.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C. C.; and Lin, D. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*.

Cheng, B.; Parkhi, O.; and Kirillov, A. 2022. Pointly-supervised instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2617–2626.

Dai, J.; He, K.; and Sun, J. 2015. BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Fan, Q.; Ke, L.; Pei, W.; Tang, C.-K.; and Tai, Y.-W. 2020. Commonality-parsing network across shape and appearance for partially supervised instance segmentation. In *European Conference on Computer Vision*, 379–396. Springer.

Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.-Y.; Cubuk, E. D.; Le, Q. V.; and Zoph, B. 2021. Simple Copy-Paste Is a Strong Data Augmentation Method for Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2918–2928.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Hsu, C.-C.; Hsu, K.-J.; Tsai, C.-C.; Lin, Y.-Y.; and Chuang, Y.-Y. 2019. Weakly supervised instance segmentation using the bounding box tightness prior. *Advances in Neural Information Processing Systems*, 32: 6586–6597.

Hu, R.; Dollár, P.; He, K.; Darrell, T.; and Girshick, R. 2018. Learning to Segment Every Thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, Z.; Wang, X.; Wang, J.; Liu, W.; and Wang, J. 2018. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7014–7023.

Jeong, J.; Lee, S.; Kim, J.; and Kwak, N. 2019. Consistency-based semi-supervised learning for object detection. *Advances in neural information processing systems*, 32: 10759–10768.

Joon Oh, S.; Benenson, R.; Khoreva, A.; Akata, Z.; Fritz, M.; and Schiele, B. 2017. Exploiting saliency for object segmentation from image level labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4410–4419.

Ke, Z.; Wang, D.; Yan, Q.; Ren, J.; and Lau, R. W. 2019. Dual Student: Breaking the Limits of the Teacher in Semi-Supervised Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Khoreva, A.; Benenson, R.; Hosang, J.; Hein, M.; and Schiele, B. 2017. Simple Does It: Weakly Supervised Instance and Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Krähenbühl, P.; and Koltun, V. 2011. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In Shawe-Taylor, J.; Zemel, R. S.; Bartlett, P. L.; Pereira, F. C. N.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, 109–117.

Kulharia, V.; Chandra, S.; Agrawal, A.; Torr, P.; and Tyagi, A. 2020. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In *European Conference on Computer Vision*, 290–308. Springer.

Kuo, W.; Angelova, A.; Malik, J.; and Lin, T.-Y. 2019. ShapeMask: Learning to Segment Novel Objects by Refining Shape Priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Lee, J.; Yi, J.; Shin, C.; and Yoon, S. 2021. BBAM: Bounding Box Attribution Map for Weakly Supervised Semantic and Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2643–2652.

Li, H.; Wu, Z.; Shrivastava, A.; and Davis, L. S. 2022a. Rethinking pseudo labels for semi-supervised object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1314–1322.

Li, W.; Liu, W.; Zhu, J.; Cui, M.; Hua, X.; and Zhang, L. 2022b. Box-supervised Instance Segmentation with Level Set Evolution. *arXiv preprint arXiv:2207.09055*.

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J. 2016. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3159–3167.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, Y.-C.; Ma, C.-Y.; He, Z.; Kuo, C.-W.; Chen, K.; Zhang, P.; Wu, B.; Kira, Z.; and Vajda, P. 2021. Unbiased Teacher

for Semi-Supervised Object Detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Papadopoulos, D. P.; Uijlings, J. R.; Keller, F.; and Ferrari, V. 2017. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision*, 4930–4939.

Pont-Tuset, J.; Arbelaez, P.; Barron, J. T.; Marques, F.; and Malik, J. 2016. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 39(1): 128–140.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. " GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3): 309–314.

Singh, A.; Chopra, A.; Garza, E.; Zhang, E.; Vepakomma, P.; Sharma, V.; and Raskar, R. 2021. DISCO: Dynamic and Invariant Sensitive Channel Obfuscation for Deep Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12125–12135.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020a. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 596–608. Curran Associates, Inc.

Sohn, K.; Zhang, Z.; Li, C.; Zhang, H.; Lee, C.; and Pfister, T. 2020b. A Simple Semi-Supervised Learning Framework for Object Detection. *CoRR*, abs/2005.04757.

Song, C.; Huang, Y.; Ouyang, W.; and Wang, L. 2019. Box-Driven Class-Wise Region Masking and Filling Rate Guided Loss for Weakly Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Taherkhani, F.; Dabouei, A.; Soleymani, S.; Dawson, J.; and Nasrabadi, N. M. 2021. Self-Supervised Wasserstein Pseudo-Labeling for Semi-Supervised Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12267–12277.

Tang, Y.; Chen, W.; Luo, Y.; and Zhang, Y. 2021. Humble Teachers Teach Better Students for Semi-Supervised Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3132–3141.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 1195–1204.

Tian, Z.; Shen, C.; and Chen, H. 2020. Conditional convolutions for instance segmentation. In *European Conference on Computer Vision*, 282–298. Springer.

Tian, Z.; Shen, C.; Wang, X.; and Chen, H. 2021. BoxInst: High-Performance Instance Segmentation With Box Annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5443–5452.

Vernaza, P.; and Chandraker, M. 2017. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7158–7166.

Wang, Z.; Li, Y.; and Wang, S. 2022. Noisy Boundaries: Lemon or Lemonade for Semi-Supervised Instance Segmentation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16826–16835.

Wei, C.; Sohn, K.; Mellina, C.; Yuille, A.; and Yang, F. 2021. CReST: A Class-Rebalancing Self-Training Framework for Imbalanced Semi-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10857–10866.

Xie, Q.; Luong, M.-T.; Hovy, E.; and Le, Q. V. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10687–10698.

Xu, M.; Zhang, Z.; Hu, H.; Wang, J.; Wang, L.; Wei, F.; Bai, X.; and Liu, Z. 2021. End-to-End Semi-Supervised Object Detection With Soft Teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3060–3069.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

Zhang, F.; Pan, T.; and Wang, B. 2022. Semi-supervised object detection with adaptive class-rebalancing self-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3252–3261.

Zheng, S.; Chen, C.; Cai, X.; Ye, T.; and Tan, W. 2022. Dual Decoupling Training for Semi-supervised Object Detection with Noise-Bypass Head. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36: 3526–3534.

Zhou, Q.; Yu, C.; Wang, Z.; Qian, Q.; and Li, H. 2021. Instant-Teaching: An End-to-End Semi-Supervised Object Detection Framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4081–4090.

Zhou, Y.; Wang, X.; Jiao, J.; Darrell, T.; and Yu, F. 2020. Learning saliency propagation for semi-supervised instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10307–10316.