

# Rethinking Mask Heads for Partially Supervised Instance Segmentation

Kai Zhao<sup>a,b,\*</sup>, Xuehui Wang<sup>a,c</sup>, Xingyu Chen<sup>a</sup>, Ruixin Zhang<sup>a</sup>, Wei Shen<sup>c</sup>

<sup>a</sup>Tencent YouTu Lab

<sup>b</sup>University of California, Los Angeles

<sup>c</sup>Shanghai Jiaotong University

---

## Abstract

We focus on partially supervised instance segmentation where only a subset of categories are mask-annotated (*seen*) and the model is expected to generalize to *unseen* categories for which only box annotations are provided to eliminate laborious mask annotations. Many recent studies train a class-agnostic segmentation network to distinguish foreground areas in each proposal. However, class-agnostic models behave poorly in complex contexts when the foreground object overlaps with other irrelevant objects. Identifying specific object categories is simpler than distinguishing foreground from background since the definition of the foreground is ambiguous even for a human. However, training class-specific model is unfeasible under the partially supervised setting since the mask annotations of *unseen* categories are absent during training. To overcome this issue, we put forward a teacher-student architecture where the teacher learns general yet comprehensive knowledge and the students, guided by the teacher, delve deeper into specific categories. Concretely, the teacher learns to segment foreground from proposals and the student is devoted to segmenting objects of specific categories. Extensive experiments on the challenging COCO dataset demonstrate our method consistently improve the performance of several recent state-of-the-art methods for the partially setting. Especially, for overlapped ob-

---

\*Corresponding author

Email addresses: [kz@kaizhao.net](mailto:kz@kaizhao.net) (Kai Zhao), [xuehuiwang@sjtu.edu.cn](mailto:xuehuiwang@sjtu.edu.cn) (Xuehui Wang), [harleychen@tencent.com](mailto:harleychen@tencent.com) (Xingyu Chen), [ruixinzhang@tencent.com](mailto:ruixinzhang@tencent.com) (Ruixin Zhang), [wei.shen@sjtu.edu.cn](mailto:wei.shen@sjtu.edu.cn) (Wei Shen)



Figure 1: (a) standalone rackets and (b) rackets overlapped with human body. Class-agnostic partially supervised instance segmentation model performs better on standalone objects than on overlapped objects

jects, our method significantly outperforms the competitors with a clear margin, demonstrating the superiority of our method.

*Keywords:* Instance segmentation, deep learning, mask head.

---

## 1. Introduction

Instance segmentation is a fundamental task in computer vision and autonomous driving. In recent years, the performance of instance segmentation has been improved to an unprecedented level since the use of convolutional neural networks (CNNs) [1, 2, 3, 4, 5]. However, CNNs require a large amount of accurate annotated samples to achieve good performance and avoid overfitting. Annotating object masks for instance segmentation is notoriously laborious and the data becomes a bottleneck to the scalization of deep instance segmentation models. Partially supervised instance segmentation [6] aims at learning from a subset of mask-annotated (*seen*) categories and then generalizing to novel (*unseen*) categories which has only box-level annotations. Since it significantly reduces the workload of data annotation and improves the efficiency of instance segmentation, partially supervised instance segmentation has become a hot research topic in recent years [7, 8, 6, 9, 10].

Hu *et al.* [7] first introduces the prototype concept of partially supervised in-

stance segmentation. They first build a mapping from box classification weights to mask segmentation weights. Then, a model trained on a subset of categories can automatically generate mask segmentation parameters on novel categories with the mapping function. Many follow-up works solve this problem with  
20 class-agnostic segmentation networks. Under the class-agnostic setting, categories degenerate to foreground and background so that missing categories is no longer a problem during training. CPMask [8] learns shared commonalities, *e.g.*, boundary or pixel affinities, that can be generalized from *seen* categories to *unseen* categories. OPMask [9] uses the class activation maps from box  
25 head to enhance localization capacity features for mask prediction. Recently, Zhou *et al.* [6] propose to learn salient maps in an instance and then iteratively propagate the salient maps to the whole object with a message passing module.

The class-agnostic methods overcome the category-missing problem by degenerating categories into foreground and background. However, the ‘foreground’ in class-agnostic models is sometimes ambiguous, especially when there  
30 are overlapped objects. For example, there are many objects overlapping with other *things* – countable objects such as people, animals, tools, as shown in Fig. 1 (a). Under such condition, the model is difficult to segment the correct areas, as shown in Fig. 1 (b). The main issue is that the definition of ‘foreground’ is am-  
35 biguous and the class-agnostic model cannot distinguish foreground object in the current proposal from objects of other categories that are possibly foreground in a different proposal. Recent empirical studies show that the class-specific methods outperform class-agnostic methods even on fully-supervised instance segmentation [11]. Directly training a class-specific network is unfeasible since  
40 the missing of categories. Hu *et al.* [7] train a ‘weight transfer function’ to map box classification weight to segmentation weight. However, the transferred segmentation weight presents unsatisfactory performance on *unseen* categories.

In this paper, we propose a teacher-student architecture where the teacher learns general yet comprehensive knowledge, and the students delve deeper into  
45 specific areas. Concretely, we use a class-agnostic head as the teacher to learn a general distinction between foreground and background. Guided by the teacher,

a class-specific head is employed as the student to learn information specific to each category. The teacher is trained with only *seen* categories and its predictions on *unseen* categories are used as supervision to a specific student according to the class label of the instance. Experimental results on the challenging COCO [12] dataset demonstrate that our method consistently outperforms many recent state-of-the-art methods in terms of segmentation quality on *unseen* categories. Our contributions are summarized as below:

1. We observe that class-agnostic methods for partially supervised instance segmentation performs poorly due to ambiguous definition of foreground for overlapped objects. We analyze that this is because these methods cannot distinguish the main object from other objects in the same bounding box.
2. We propose a teacher-student architecture to perform ‘general to class-specific’ knowledge transfer. The teacher learns general and comprehensive knowledge about foreground objects while the students in contrast delve deeper into specific categories, leading to easier learning and better performance.
3. We apply the proposed method to many recent state-of-the-art approaches to partially supervised instance segmentation and observe consistent performance gain on the challenging COCO dataset.

## 2. Related Work

**Instance Segmentation.** Instance segmentation is a classical problem in computer vision. Roughly, instance segmentation methods can be divided into two categories: 1) top-down detection based and 2) bottom-up grouping-based methods. Top-down instance segmentation methods [1, 13, 3, 14, 15, 3] extend the object detection framework, *e.g.*, faster r-cnn [16], with a mask prediction module. Mask R-CNN [1] extends the Faster R-CNN [16] with a FCN [17] branch to segment object area in object bounding boxes. FCIS [3] introduces position-sensitive representations for mask segmentation. Liu *et al.* [15] add

bottom-up path augmentation in addition to the top-down path in FPN [18] to enrich feature representations. Mask Scoring R-CNN [13] calibrates the misalignment between mask quality and mask score by introducing a submodule to explicitly predict quality scores. Shang *et al.* [4] utilize the instance-level contexts to enhance feature representations for instance segmentation. Huang *et al.* [19] use the peak response map from a pretrained classification network for box-supervised instance segmentation. Yang *et al.* [20] propose a one-stage and anchor-free framework for instance segmentation based on boundary point representations. Xiang *et al.* [21] improve the segmentation quality with class-specific semantic feature and instance-specific attributes.

On the other hand, bottom-up instance segmentation methods first obtain a pixel-wise semantic segmentation mask over an image and then group pixels into individual instances. Zhang *et al.* [22, 23] propose to assign instance labels based on local patches and integrate the local results with an MRF. [24] proposes the deep watershed algorithm to segmentation pixels into instances. [25] learns bound-aware representations. Recently, there are many single-stage instance segmentation methods achieving promising performance in both accuracy and speed. InstanceFCN [26] and FCIS [3] use a fully-convolutional network [17] to produce instance-sensitive score maps which contain the relative positions for each objects instances, and then apply an assembling module to output object instances. PolarMask [27] uses rays at constant angle intervals and then describes the contour of an object using the distance between the center and the edge of the object. SOLO [28] assigns categories to each pixel within an instance according to the instance’s location, leading to a simple and fast method for instance segmentation with strong performance. Generally, bottom-up methods are faster than top-down methods, but with the sacrifice of performance.

**Partially supervised Instance Segmentation.** Generalizing instance segmentation models to novel categories with limited or weak annotations are useful and challenging. In the partially supervised instance segmentation setting, only a subset object categories are mask-annotated, all other categories have only

box annotations. The model is trained on mask-annotated images and tested on box-annotated images.

For the past few years, several achievements have been made for partially supervised instance segmentation [7, 8, 9, 6]. Hu *et al.* [7] first introduced the concept of ‘partially supervised instance segmentation. Their method builds up a mapping function between the parameters of the box head and the mask head and therefore generates mask head parameters for *unseen* categories with box head weights. CPMask [8] learns the common low-level clues, *e.g.*, edge and pixel affiliations, from *seen* categories to enhance performance on *unseen* categories. OPMask [9] employs the class activation map (CAM) from the box head as a coarse localization for object segmentation inside proposals. Recently, [6] propose the ‘ShapeProp’ that activates the salient areas in a bounding box and then propagates the area to the whole instance using an iterative message passing module. [29] suggests that using extremely deep mask heads can significantly improve the performance of partially-supervised instance segmentation. Wang *et al.* [30] use pixel-wise contrast learning to improve the feature representations. Most of them have employed the class-agnostic architecture that trains foreground/background segmentation network on *seen* categories and then transfer to *unseen* categories [8, 9, 6]. However, it is challenging to perform binary segmentation under complex contexts where objects are heavily overlapped with others and the background is noisy.

### 3. Observation on Class-agnostic Segmentation

#### 3.1. Performance w.r.t overlaps

Here we quantitatively analyze the impact of object overlaps on the performance of class-agnostic segmentation. We experiment with Mask R-CNN [1] on the COCO [12] dataset. We train a class-agnostic mask head with voc categories of the training set, and then test with non-voc categories from the testing set.

We define the overlap of object  $i$  as:

$$O_i = \max_{j \in \mathbf{I}, j \neq i} \frac{b_i \cap s_j}{b_i}, \quad (1)$$

where  $b, s$  are the bounding box and segmentation mask of an object,  $\cap$  denotes  
 135 the intersection between two areas. Here we use the intersection between  $b_i$  and  
 $s_j$  because the mask annotations of two overlapping objects may not intersect.  
 The definition in Eq. (1) represents the degree of an object being overlapped by  
 other objects. Based on the definition in Eq. (1), we calculate the segmentation  
 performance *w.r.t* the overlaps.

140 In Fig. 2, horizontal axis  $x_i$  means the performance of all objects with over-  
 lap  $x_{i-1} < O \leq x_i$ . The results clearly demonstrate that the class-agnostic  
 model performs better on stand-alone objects and the performance degrades on  
 overlapped objects.

### 3.2. Attribution of mis-segmentations

145 Now we analyze the attribution of mis-segmentations for a class-specific  
 model and a class-agnostic model. We count the *false positive* pixels on the  
 testing set and attribute them into 3 different cases: background, *seen* objects  
 and *unseen* objects. We use the Mask-rcnn [1] and COCO dataset [12] for  
 both fully-supervised and partially-supervised experiments. Under the fully-  
 150 supervised setting, a class-specific mask head is used. For partially-supervised  
 setting, a class-agnostic head is used and the model is supervised with pseudo  
 masks generated by itself on *unseen* categories. The statistics are in Fig. 3.

As shown in Fig. 3, with a class-specific head, most of the false positives occur  
 on the background areas that do not belong to any object. As for the class-

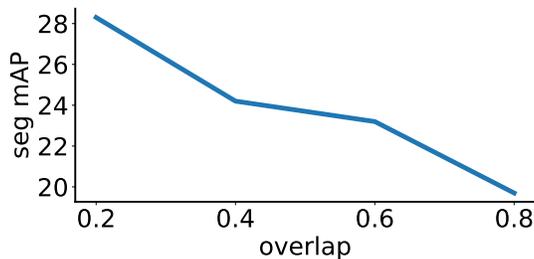


Figure 2: Segmentation performance *w.r.t*. object overlaps. The larger the object overlaps,  
 the lower the performance.

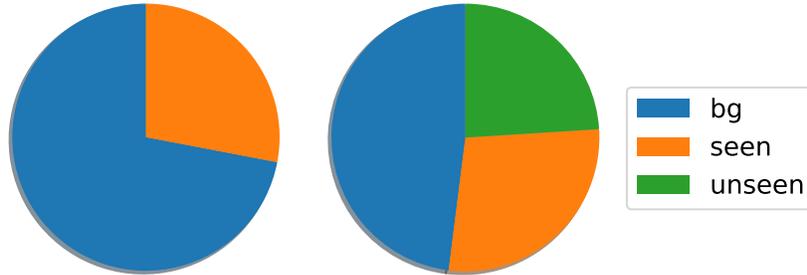


Figure 3: Attribution of false positive of fully supervised (left) and partially supervised (right) Mask R-CNN.

155 agnostic head, more than half of the false positives are on objects. The results suggest that the class-agnostic head has difficulty in recognizing foreground objects from irrelevant objects. This motivates us to develop a class-specific method for partially supervised instance segmentation.

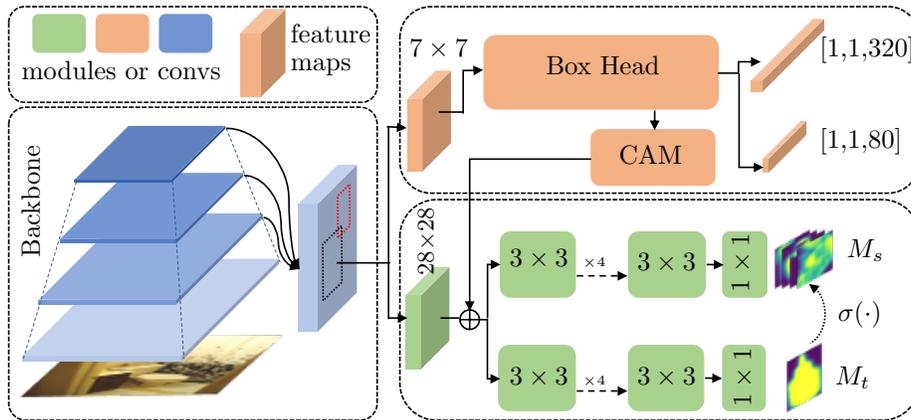


Figure 4: The overall network architecture of our proposed method. The teacher mask head ( $M_t$ , bottom) and the student mask head ( $M_s$ , top) share the same ROI features.  $M_t$  is supervised with *seen* categories and produces pseudo masks on the *unseen* categories as the supervision of  $M_s$ . The box branch (top-right) is supervised by samples of all categories.

#### 4. Methodology

160 In this section, we introduce the proposed method for partially supervised instance segmentation.

We denote the set of categories in a dataset with  $\mathbb{C}$ , and  $C = |\mathbb{C}|$  is the number of classes. We denote  $\mathbb{S}$  as the *seen* categories that are mask-annotated, and  $\mathbb{C} \setminus \mathbb{S}$  is *unseen* categories without mask annotations.

165 *4.1. Teacher and student mask heads*

Our method consists of two mask heads: the teacher mask head and the student head. The teacher head performs binary segmentation in each proposal to distinguish foreground and background areas. During training, the teacher mask head is trained only on objects of *seen* categories and b) the student mask  
170 head performs multi-classification to assign each pixel to a specific category.

Let  $x \in \mathbb{R}^{D \times H \times W}$  be the ROI features and  $F_t, F_s$  be the teacher and student mask heads, where  $D$  is the dimensionality and  $H \times W$  are spatial size. Following many previous works [1, 7, 13],  $F_t, F_s$  consist of four convolutions. The outputs are  $M_t \in \mathbb{R}^{H \times W}$  and  $M_s \in \mathbb{R}^{C \times H \times W}$  are:

$$\begin{aligned} M_t &= F_t(x) \\ M_s &= F_s(x), \end{aligned} \tag{2}$$

175 where  $C$  is the number of categories, *e.g.*,  $C = 80$  for COCO dataset.

Let  $W_t \in \mathbb{R}^{D \times 1}, W_s \in \mathbb{R}^{D \times C}$  be the weights of the last convolutional layers in  $F_t, F_s$ , respectively.  $W_t \in \mathbb{R}^{D \times 1}$  can be trained only with *seen* categories and test with *unseen* categories. However, only  $|\mathbb{S}|$  of  $C$  kernels in  $W_s \in \mathbb{R}^{D \times C}$  will be updated during training under the partially supervised setting, meaning that  
180 class-specific models cannot generalize to novel classes. We train both teacher and student mask heads with ground-truth masks of *seen* categories. For *unseen* categories, the predictions from the teacher head are used as pseudo masks to supervise the student head.

Given an arbitrary object with class label  $l \in \{1, 2, \dots, C\}$ , we denote the ground-truth of the student head as  $G \in \{0, 1\}^{C \times H \times W}$ , and  $G_c \in \{0, 1\}^{H \times W}, c = 1, 2, \dots, C$  is the  $c$ -th channel of  $G$ . The values of all unrelated channels are zero:

$$G_c = 0^{H \times W}, c \neq l. \tag{3}$$

If the object is from *seen* categories, *e.g.*,  $l \in \mathbb{S}$ ,  $G_c$  is the human-annotated mask. If the object is from *unseen* categories, *e.g.*,  $l \notin \mathbb{S}$ ,  $G_c$  is assigned according the output of teacher mask head:

$$G_c = \sigma(M_t), \tag{4}$$

190 where  $\sigma$  is the ‘post-processing’ function which will be detailed in the next sub-section.

Consequently, the class-specific student head can be trained on all categories and generalize well to the *unseen* categories.

#### 4.2. Post-processing of pseudo masks

195 The teacher mask head  $F_t$  is only supervised by the mask-annotated categories (*seen*) and is expected to generalize to novel categories (*unseen*). For training sample  $i$ , the optimization objective for teacher mask head is:

$$\mathcal{L}_{M_t}^i = \begin{cases} \mathcal{L}(M_t^i, M^i) & i \in \mathbb{S} \\ 0 & i \notin \mathbb{S}, \end{cases} \tag{5}$$

where  $\mathcal{L}(\cdot)$  is a certain loss function, *i.e.*, cross-entropy loss or GIoU loss, and  $M^i$  is the ground-truth mask of the sample.

200 Student mask head  $F_s$  has to be trained on all categories so that it can perform multi-classification during testing. For *seen* categories, we use the ground-truth mask annotations as the supervision. For *unseen* categories, the teacher mask head will generate pseudo masks for student mask head. Formally, the training objective of student mask head is:

$$\mathcal{L}_{M_s}^i = \begin{cases} \mathcal{L}(M_s^i, M^i) & i \in \mathbb{S} \\ \mathcal{L}(M_s^i, \sigma(M_t^i)) & i \notin \mathbb{S}, \end{cases} \tag{6}$$

205 where  $\sigma(\cdot)$  is the post-processing function for  $M_t$ . We experiment 3 different post-processing functions:

1. threshold:  $\sigma(M_t) = M_t \geq 0.5$ ;

2. threshold after CRF:  $\sigma(M_t) = \text{CRF}(M_t) \geq 0.5$ ;
3. threshold after interaction:  $\sigma(M_t) = \frac{(M_t + M_s)}{2} \geq 0.5$ .

210 Experimental results reveal that simply thresholding  $M_s$  achieves good results. Applying CRF [31] to  $M_t$  slightly improves the performance but brings significant computation. And interacting the two mask heads improves the performance with neglectable computation overhead.

### 4.3. Application to Existing Methods

215 When applying our method to existing methods, *e.g.*, OPMask [9], Mask R-CNN [1] and BoxInst [10]. We keep their original settings unchanged and add the teacher-student architecture to the original models. During training, the only additional loss introduced by our method is the segmentation loss of student mask head, all other loss terms are the same with the baseline methods.

220 For example, with the Mask R-CNN baseline, there are mainly 3 loss terms: 1. the classification and localization losses from the box head; 2. the segmentation loss from teacher mask head (only *seen* categories, cross-entropy loss), 3. the segmentation loss from student mask head (both *seen* and *unseen* categories (cross-entropy loss). The total loss is the sum of these loss terms with equal

225 contribution:

$$\mathcal{L} = L_{cls}^{box} + L_{loc}^{box} + L_T^{seg} + L_S^{seg}, \quad (7)$$

where  $L_{cls}^{box}, L_{loc}^{box}$  are the classification loss and localization loss in the bounding box head, and  $L_T^{seg}, L_S^{seg}$  are segmentation losses in teacher and student mask heads. During inference, the teacher mask head is removed and only the student mask head is preserved.

## 230 5. Experiments

In this section, we introduce implementation details of our method and report the experimental results. We evaluate the proposed method on the challenging COCO [12] dataset under partially supervised setting [7, 8]. The detailed information about the experimental settings can be found in Sec. 5.1.

**Experimental configurations.** All the evaluated methods are implemented based on the MMDetection [11] framework. For a fair comparison, all the evaluated algorithms are trained on the training subset and evaluated on the testing subset under the partially supervised setting. For our proposed method, we use ResNet-50 with the feature pyramid network (FPN) as the backbone. In the inference phase, the network outputs top 512 high confident proposals for each image. Then, we use the non-maximum suppression (NMS) with Jaccard overlap of 0.5 and retain the top 150 high confident detections per image to generate the final results. All the experiments are conducted on a machine with 8 NVIDIA V100 GPUs and a 2.80GHz Intel(R) Xeon(R) E5-1603 v4 processor.

All the experiments are conducted on the COCO dataset [12]. There are totally 80 categories in COCO dataset and they are splited into “*voc*” and “*non-voc*” according to whether they are included by the PASCAL VOC dataset [32]. There are 20 categories in the *voc* subset and 60 categories in the *nonvoc* subset. We mainly conduct experiments on these two settings: “*nonvoc*  $\rightarrow$  *voc*” and “*voc*  $\rightarrow$  *nonvoc*”. “*nonvoc*  $\rightarrow$  *voc*” indicates that “*nonvoc*” categories are regarded as *seen* and “*voc*” as *unseen*”, and vice versa. We use images in COCO-*train2017* for training and those in COCO-*val2017* for evaluation. Typical metrics for instance segmentation, *i.e.*, mask AP, including mAP, AP<sub>50</sub>, AP<sub>75</sub>, AP<sub>S</sub>, AP<sub>M</sub> and AP<sub>L</sub>, are used for evaluation. The performance is only evaluated on the *unseen* categories.

The batch size is set to 16 in the training phase with each GPU processing 2 images in an iteration. The whole network is trained using the stochastic gradient descent (SGD) algorithm with the 0.9 momentum and  $1e^{-4}$  weight decay. All experimental results are obtained using the 1x schedule under which models are trained for 12 epochs. The initial learning rate is set to 0.02 decreases by 0.1 after training 8 and 11 epochs, respectively. We warm up the learning rate for the first 500 iterations.

**Partially supervised training details.** During training, the teacher mask

265 head is trained with only *seen* categories and the student mask head is trained with all instances with either ground-truth masks or pseudo-masks. If a pseudo mask intersects with any mask of *seen* category, the intersection is marked as background, and all areas outside the ground-truth bounding boxes are regarded as background.

**Class Activation Map.** OPMask [9] uses the class-activation map (CAM) [33]

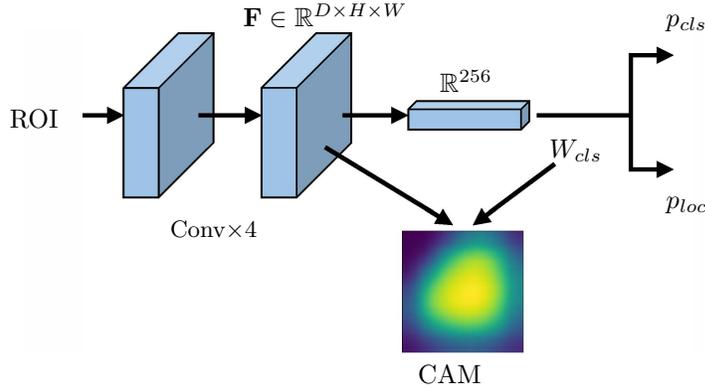


Figure 5: Illustration of the class activation map (CAM).  $W_{cls}$  is the classification weights,  $p_{cls}$  and  $p_{loc}$  are the classification and localization predictions, respectively.

270

to enhance the localization ability of mask features. In our implementation, the CAM is generated by applying the classification weight  $W_{cls}$  of the box head to features before global average pooling. The generation of CAM is illustrated in Fig. 5.

275 Let  $\mathbf{F} \in \mathbb{R}^{7 \times 7 \times 256}$  be the feature map of the last convolutional layer in the bounding-box head. Then the classification and regression predictions are:

$$p_{cls} = \text{GAP}(\mathbf{F}) \cdot W_{cls} + b_{cls} \quad (8)$$

$$p_{reg} = \text{GAP}(\mathbf{F}) \cdot W_{reg} + b_{reg}$$

where ‘GAP’ denotes the ‘global average pooling’ operation and  $W_{cls} \in \mathbb{R}^{D \times C}$  is the classification weight and  $b \in \mathbb{R}^{80}$  the bias. Then we apply the classification parameters to each position of feature  $\mathbf{F}$  before ‘GAP’ to calculate the class-activation map  $\text{CAM} \in \mathbb{R}^{H, W}$

280

$$\text{cam}_{i,j} = \mathbf{F}_{i,j} \cdot W_{cls} + b_{cls} \quad (9)$$

## 5.2. Comparison with SOTA methods

We report quantitative comparisons of our method against Mask R-CNN [1] and many recent state-of-the-art methods including: Mask<sup>X</sup> R-CNN [7], Mask GrabCut [7], CPMask [8], OPMask [9], ShapeProp [6] and BoxInst [10]. We report the performance under two-fold validation: 1) training on voc categories and testing on non-voc categories (denoted as voc→non-voc) and 2) training on non-voc categories and testing on voc categories (denoted as non-voc→voc). The voc→non-voc setting is much more challenging since there are more novel categories. We also present the visual results in Fig. 6.

It can be seen in Tab. 1 that our method significantly improves the performance based on Mask R-CNN, and presents consistent performance improvement on many strong baselines such as OPMask [9] and BoxInst [10]. The

| Backbone | method                  | voc→non-voc |                  |                  |                 |                 |                 | non-voc→voc |                  |                  |                 |                 |                 |
|----------|-------------------------|-------------|------------------|------------------|-----------------|-----------------|-----------------|-------------|------------------|------------------|-----------------|-----------------|-----------------|
|          |                         | AP          | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> | AP          | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
| R50-FPN  | Mask R-CNN              | 19.2        | 36.4             | 18.4             | 11.5            | 23.3            | 24.4            | 23.9        | 42.9             | 23.5             | 11.6            | 24.3            | 33.7            |
|          | Mask R-CNN + ours       | 21.6        | 37.1             | 19.1             | 12.7            | 16.5            | 33.1            | 25.1        | 43.4             | 23.9             | 11.7            | 25.9            | 34.3            |
|          | Mask <sup>X</sup> R-CNN | 23.7        | 43.1             | 23.5             | 12.4            | 27.6            | 32.9            | 28.9        | 52.2             | 28.6             | 12.1            | 29.0            | 40.6            |
|          | Mask GrabCut            | 19.7        | 39.7             | 17.0             | 6.4             | 21.2            | 35.8            | 19.6        | 46.1             | 14.3             | 5.1             | 16.0            | 32.4            |
|          | CPMask                  | 28.8        | 46.1             | 30.6             | 12.4            | 33.1            | 43.4            | 34.9        | 61.1             | 35.0             | 14.9            | 34.3            | 48.3            |
|          | OPMask                  | 29.7        | 48.7             | 31.7             | 13.3            | 33.8            | 42.3            | 35.0        | 60.6             | 36.1             | 15.4            | 34.8            | 48.1            |
|          | OPMask + ours           | 30.9        | 50.4             | 32.8             | 14.5            | 35.3            | 44.6            | 35.4        | 61.0             | 36.1             | 15.9            | 34.3            | 48.8            |
|          | BoxInst                 | 30.4        | 51.2             | 31.8             | 14.3            | 34.2            | 44.7            | 33.9        | 59.6             | 34.8             | 13.5            | 32.9            | 48.6            |
|          | BoxInst + ours          | 31.4        | 50.6             | 32.8             | 14.7            | 35.1            | 45.8            | 34.4        | 60.3             | 34.6             | 14.6            | 33.3            | 47.7            |
|          | Oracle                  | 37.5        | 63.1             | 38.9             | 15.1            | 36.0            | 53.1            | 33.0        | 53.7             | 35.0             | 15.1            | 37.0            | 49.9            |
| R101-FPN | OPMask                  | 31.9        | 51.7             | 33.8             | 14.7            | 36.2            | 46.4            | 35.0        | 59.7             | 35.7             | 16.9            | 34.7            | 47.3            |
|          | OPMask + ours           | 32.2        | 52.0             | 34.0             | 14.3            | 36.9            | 47.0            | 35.6        | 60.6             | 36.1             | 15.4            | 34.8            | 48.1            |
|          | BoxInst                 | 31.9        | 52.1             | 33.7             | 14.2            | 35.9            | 46.5            | 35.5        | 60.5             | 36.7             | 15.6            | 33.8            | 50.3            |
|          | BoxInst+ours            | 32.6        | 52.4             | 33.7             | 14.0            | 36.1            | 46.9            | 35.9        | 61.0             | 36.9             | 15.6            | 34.0            | 50.6            |
|          | Oracle                  | 34.3        | 54.7             | 36.3             | 18.6            | 39.1            | 47.9            | 38.5        | 64.4             | 40.4             | 18.9            | 39.4            | 51.4            |

Table 1: Quantitative comparisons on the challenging COCO dataset. All the models are trained with the standard 1x schedule on 8 GPUs. Our proposed method significantly outperforms the baseline (Mask R-CNN) and present superior performance against strong competitors.

performance improvements under the voc→non-voc setting are more significant than the non-voc→voc setting, suggesting that the proposed method is especially suitable for partially supervised instance segmentation.

The segmentation results in Fig. 6 show that our class-specific method

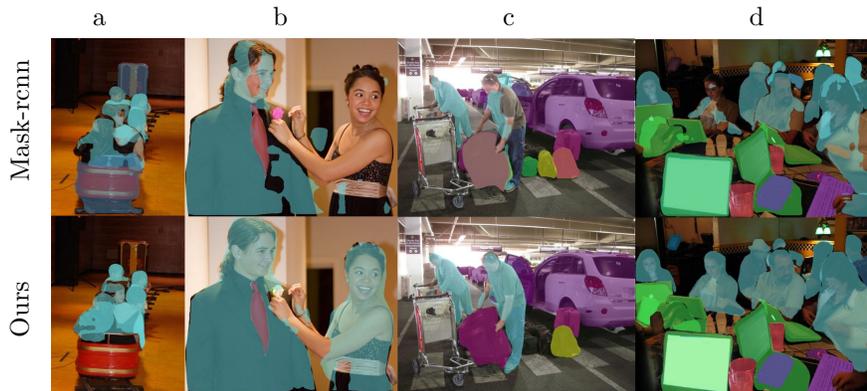


Figure 6: Example instance segmentation results of the Mask-rcnn baseline and our method.

performs better than the class-agnostic foreground/background segmentation method under different scenarios. Fig. 6 (a, c and d) demonstrate that our method performs better with overlapped objects. In Fig. 6 (b), the girl has low contrast against the background hence the foreground/background segmentation method cannot distinguish and segment well. While our class-specific method learns to recognize humans from a all human images from the dataset and can segment well under low contrast.

### 5.3. Experiments with overlapped objects

Since our method are designed to overcome the object overlapping problem in class-agnostic segmentation, in this section, we quantitatively evaluate the performance of our method against baseline on segmenting object that are overlapping with other objects. Under the ‘voc→non-voc’ setting, we collect all testing objects with overlap  $O \geq 0.3$  for evaluation. The object overlap was defined in Eq. (1). The quantitative evaluation results of these ‘overlapped objects’ are in Tab. 2, some segmentation results are shown in Fig. 7. Our method presents clear advantages in distinguishing foreground objects from other overlapped objects.

### 5.4. Stronger teacher heads

Our teacher-student mask heads naturally benefits from stronger teacher mask heads since stronger teachers provide better supervision for the student.



Figure 7: Segmentation results of a) vanilla Mask R-CNN (with mask-agnostic mask head) and b) our method on overlapped objects.

| Method            | AP   | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
|-------------------|------|------------------|------------------|-----------------|-----------------|-----------------|
| Mask R-CNN        | 16.2 | 31.2             | 15.4             | 9.9             | 14.3            | 24.4            |
| Mask R-CNN + ours | 17.3 | 44.1             | 15.8             | 10.4            | 14.9            | 32.1            |
| OPMask            | 27.4 | 39.7             | 37.5             | 12.9            | 28.4            | 38.6            |
| OPMask + ours     | 28.7 | 41.1             | 38.1             | 13.3            | 29.2            | 39.8            |

Table 2: Segmentation performance on overlapped objects.

Here we test two stronger teachers: **deeper** and **higher-resolution**. The deeper teacher mask head has 12 convolutional layers (the original mask head has 4), and the higher-resolution teacher head outputs  $56 \times 56$  masks. According

| Teacher  | voc $\rightarrow$ non-voc | non-voc $\rightarrow$ voc |
|----------|---------------------------|---------------------------|
| Original | 21.6                      | 25.1                      |
| High-res | 22.1                      | 25.7                      |
| Deeper   | 26.3                      | 29.4                      |

Table 3: Performance with stronger teacher heads. Higher-resolution teacher slightly improves the performance, and deeper teacher improves the performance with significant margin.

320 to the results in Tab. 3, higher-resolution teacher head slightly improves the performance, and deeper teacher head significantly improve the performance with a very clear margin. The results reveal that the resolution is not a bottleneck in the partially-supervised instance segmentation setting, while the generation

ability of identifying ‘foreground objects’ is the key to this task.

325 *5.5. Experiments with various number of seen categories*

In addition to the voc *v.s.* non-voc split, we conduct experiments on various number of *seen/unseen* splits to verify the effectiveness of our method. Starting from the ‘voc→non-voc’ setting, we gradually add non-voc categories to *seen* categories. We compare our method with the baseline methods Mask R-CNN [1] and OPMask [9]. The results in Fig. 8 reveals that our method consistently

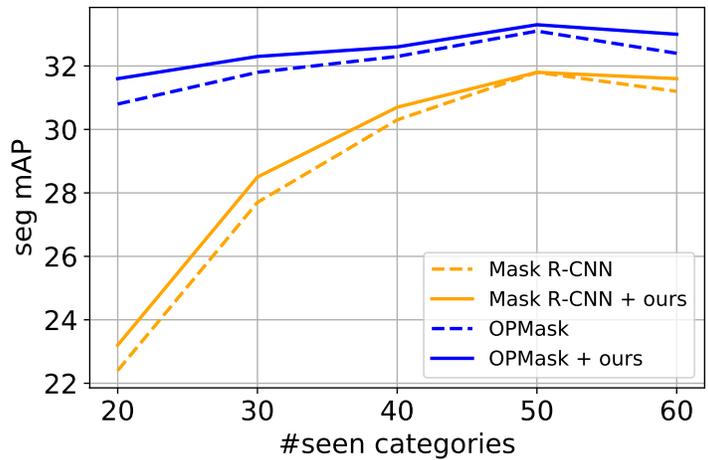


Figure 8: Performance with various *seen/unseen* splits. The horizontal axis indicates the number of *seen* categories. Our proposed method consistently improve the performance of two baseline methods.

330

improves the baselines. The performance gap is even larger then there are more *unseen* categories, demonstrating that our method’s superiority under extreme partially-supervised conditions.

*5.6. Ablation studies*

335

In this section, we conduct ablation studies to verify the design choice in our method. All the experiments in this section are based on the ResNet-50 [34] backbone trained on voc categories and tested on non-voc categories.

**Effectiveness of Components.** We first ablate the components of our method including: 1) the teacher-student architecture, and 2) interaction between teacher

and student mask heads. Results in Tab. 4 demonstrate that our teacher-student

| teacher-student | post-processing | voc→non-voc |
|-----------------|-----------------|-------------|
| ✗               | ✗               | 19.2        |
| ✓               | ✗               | 20.2        |
| ✓               | ✓               | 21.6        |

Table 4: Ablation study on the teacher-student architecture and post-processing of pseudo masks.

340

architecture improves the performance of the baseline method, and the post-processing of pseudo-masks plays a positive role in our method.

**Post-processings of pseudo masks.** We test three difference post-processing methods on the pseudo masks  $M_s$ : CRF<sup>1</sup>, threshold with 0.5 and interaction between  $M_t$  and  $M_s$ . Results are in Tab. 5. CRF is a widely used post-process

|           | voc→non-voc | non-voc→voc | Time (Hours) |
|-----------|-------------|-------------|--------------|
| CRF       | 31.4        | 34.5        | 36           |
| thres=0.5 | 31.1        | 34.2        | 12           |
| Interact  | 31.4        | 34.4        | 12           |

Table 5: The performance of three different post-processing methods. Experiments are conducted using the BoxInst [10]+ResNet50 as baseline.

345

approach for many semantic segmentation methods such as DeepLab [35]. However, CRF is time-consuming and our experiments show that simply binarization with a threshold performs on par with CRF. And interacting  $M_s$  with  $M_s$  achieves the best performance with neglectable computations.

## 350 6. Conclusion

We observed that the coexistence of multiple objects in a single proposal box is an obstacle to the performance of class agnostic models for partially supervised instance segmentation, since the model cannot distinguish the foreground

<sup>1</sup>Implemented with the pydensecrf <https://github.com/lucasb-eyer/pydensecrf> package with default parameters.

object from other objects in the proposal box. Motivated by the observation, we  
355 proposed a class-specific method to segment specific object categories instead  
of a foreground object. We designed a teacher-student architecture consists of a  
class-agnostic teacher head and a class-specific student head. The teacher head  
is trained with ground-truth masks on *seen* categories and the student head  
is trained with both *seen* and *unseen* categories with ground-truth or pseudo  
360 masks. During testing, the student head can segment foreground objects from  
other objects by identifying the specific category. Extensive experiments on  
the challenging COCO dataset demonstrate that our method consistently im-  
proved the performance of several existing state-of-the-art methods on partially  
supervised instance segmentation.

## 365 **References**

### **References**

- [1] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
- 370 [2] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, et al., Hybrid task cascade for instance segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4974–4983.
- [3] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance-aware se-  
375 mantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2359–2367.
- [4] C. Shang, H. Li, F. Meng, H. Qiu, Q. Wu, L. Xu, K. N. Ngan, Instance-level context attention network for instance segmentation, *Neurocomputing* 472 (2022) 124–137.

- 380 [5] Y. Sun, L. Su, Y. Luo, H. Meng, W. Li, Z. Zhang, P. Wang, W. Zhang, Global mask r-cnn for marine ship instance segmentation, *Neurocomputing* (2022).
- [6] Y. Zhou, X. Wang, J. Jiao, T. Darrell, F. Yu, Learning saliency propagation for semi-supervised instance segmentation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 385 2020, pp. 10307–10316.
- [7] R. Hu, P. Dollár, K. He, T. Darrell, R. Girshick, Learning to segment every thing, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4233–4241.
- 390 [8] Q. Fan, L. Ke, W. Pei, C.-K. Tang, Y.-W. Tai, Commonality-parsing network across shape and appearance for partially supervised instance segmentation, in: *European Conference on Computer Vision*, Springer, 2020, pp. 379–396.
- [9] D. Biertimpel, S. Shkodrani, A. S. Baslamisli, N. Baka, Prior to segment: 395 Foreground cues for weakly annotated classes in partially supervised instance segmentation, *arXiv preprint arXiv:2011.11787* (2020).
- [10] Z. Tian, C. Shen, X. Wang, H. Chen, Boxinst: High-performance instance segmentation with box annotations, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5443–400 5452.
- [11] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, D. Lin, MMDetection: Open mmlab detection toolbox and benchmark, 405 *arXiv preprint arXiv:1906.07155* (2019).
- [12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan,

- P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.
- [13] Z. Huang, L. Huang, Y. Gong, C. Huang, X. Wang, Mask scoring r-cnn, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6409–6418.
- [14] X. Chen, R. Girshick, K. He, P. Dollár, Tensormask: A foundation for dense object segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2061–2069.
- [15] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
- [16] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015) 91–99.
- [17] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
- [18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
- [19] X. Huang, Q. Zhu, Y. Liu, S. He, Weakly supervised segmentation via instance-aware propagation, Neurocomputing 447 (2021) 1–9.
- [20] H. Yang, L. Zheng, S. G. Barzegar, Y. Zhang, B. Xu, Borderpoints-mask: One-stage instance segmentation with boundary points representation, Neurocomputing 467 (2022) 348–359.
- [21] C. Xiang, W. Zou, C. Xu, Cimask: Segmenting instances by class-specific semantic feature extraction and instance-specific attribute discrimination, Neurocomputing 464 (2021) 164–174.

- 435 [22] Z. Zhang, S. Fidler, R. Urtasun, Instance-level segmentation for autonomous driving with deep densely connected mrfs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 669–677.
- [23] Z. Zhang, A. G. Schwing, S. Fidler, R. Urtasun, Monocular object instance  
440 segmentation and depth ordering with cnns, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2614–2622.
- [24] M. Bai, R. Urtasun, Deep watershed transform for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5221–5229.
- 445 [25] Z. Hayder, X. He, M. Salzmann, Boundary-aware instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5696–5704.
- [26] J. Dai, K. He, J. Sun, Instance-aware semantic segmentation via multi-task network cascades, in: Proceedings of the IEEE conference on computer  
450 vision and pattern recognition, 2016, pp. 3150–3158.
- [27] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, P. Luo, Polarmask: Single shot instance segmentation with polar representation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 12193–12202.
- 455 [28] X. Wang, T. Kong, C. Shen, Y. Jiang, L. Li, Solo: Segmenting objects by locations, in: European Conference on Computer Vision, Springer, 2020, pp. 649–665.
- [29] V. Birodkar, Z. Lu, S. Li, V. Rathod, J. Huang, The surprising impact of mask-head architecture on novel class segmentation, in: Proceedings of  
460 the IEEE/CVF International Conference on Computer Vision, 2021, pp. 7015–7025.

- [30] X. Wang, K. Zhao, R. Zhang, S. Ding, Y. Wang, W. Shen, Contrast-mask: Contrastive learning to segment every thing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 11604–11613.
- [31] P. Krähenbühl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, *Advances in neural information processing systems* 24 (2011) 109–117.
- [32] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, *International Journal of Computer Vision* 111 (1) (2015) 98–136.
- [33] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: *Computer Vision and Pattern Recognition*, 2016.
- [34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, *IEEE transactions on pattern analysis and machine intelligence* 40 (4) (2017) 834–848.