Parallel Detection-and-Segmentation Learning for Weakly Supervised Instance Segmentation

Yunhang Shen¹,², Liujuan Cao¹*, Zhiwei Chen¹, Baochang Zhang³
Chi Su⁴, Yongjian Wu², Feiyue Huang², Rongrong Ji¹,⁵,⁶
¹Media Analytics and Computing Lab, Department of Artificial Intelligence,
School of Informatics, Xiamen University, 361005, China, ²Tencent Youtu Lab, Shanghai, China
³Institute of Artificial Intelligence, Beihang University, Beijing, China
⁴KingSoft Cloud Co. Ltd., Beijing, China, ⁵Institute of Artificial Intelligence,
Xiamen University, Xiamen, China, ⁶Peng Cheng Laboratory, Shenzhen, China
{odysseyshen, littlekenwu, garyhuang}@tencent.com, {caoliujuan, rrji}@xmu.edu.cn
zhiweichen.xmu@gmail.com, bczhang@buaa.edu.cn, suchi@kingsoft.com

1. Introduction

Instance segmentation [1, 2] is one of the fundamental tasks in computer vision, which aims to simultaneously localize bounding boxes, classify target categories and estimate segmentation masks of object instances in images. Despite its significant progress in recent years, the dominant paradigms require a large number of training images with instance-level pixel-wise human annotations. However, collecting such fully-labelled training data is labor-intensive [3] and restricts applicability of instance segmentation in many downstream high-level vision tasks, ranging from autonomous driving, pose estimation to image synthesis. Thus, it has motivated the exploration of weakly supervised instance segmentation (WSIS), especially the setting where only image-level labels are used during training.

WSIS is an extremely challenging task with only a few attempts in previous literature. The bottom-up WSIS methods [4, 5, 6, 7, 8, 9] used classification networks to identify object instances from discriminative localization cues in the images. While this is a promising line of works, the initial localization cues are quite coarse and not consistent with object boundaries [10], which over-concentrates on discriminative parts of objects and under-estimates small instances [11]. Moreover, those methods suffer from sophisticated and multi-stage training processes to refine in--
intermediate results, which also rely on class activation maps and segmentation proposals. On the other hand, the top-down methods [10, 12] requires weakly supervise object detection (WSOD) to generate pseudo-ground-truth masks by leveraging gradient of image pixels w.r.t. detection results. Such detection-to-segmentation multi-task cascade causes the quality of pseudo masks heavily depending on WSOD, which limits further improvement with large margins. Although obtaining end-to-end pipeline, the top-down approaches have significantly inferior performance compared to the bottom-up ones with sophisticated training procedures prevailing in public benchmarks [13, 14].

To conquer the aforementioned limitations, we propose a unified parallel detection-and-segmentation learning (PDSL) framework to learn instance segmentation with only image-level labels, which draws inspiration from both top-down and bottom-up instance segmentation approaches. Our motivation is to parallel top-down detection and bottom-up segmentation via correlation learning in an end-to-end manner. The proposed PDSL framework has three advantages. First, compared to top-down WSIS methods, PDSL decouples the generation of pseudo-ground-truth masks from detectors and explores bottom-up object cues to learn segmentation, which models class-agnostic to class-specific foreground masks progressively. Second, compared to bottom-up methods, PDSL imposes bounding-box constraint from detection module on segmentation learning, which are encouraged to match up with object boundaries. Third, PDSL further collaborates detection module with segmentation learning by explicitly modelling correlations between them.

To this end, the proposed PDSL consists of three key components: object detection, image segmentation and correlation learning modules, as illustrated in Fig. 1. First, the detection branch is the same as the typical design of any top-down detectors learned from image-level labels. Second, the segmentation branch leverages self-supervised learning to model class-agnostic foreground extraction, which is followed by self-training to learn class-specific object segmentation constrained by bounding boxes. Third, to improve the coherence between detection and segmentation, we further propose instance-activation correlation learning, which impose a high correlation between two branches for activation of the same object instances. Extensive experiments on PASCAL VOC [13] and MS COCO [14] show that the proposed PDSL outperforms baseline models and achieves the state-of-the-art results. For the first time, we show that a top-down approach delivers competitive WSIS results.

The contributions of this work are three folds:

- We propose a cooperative parallel detection-and-segmentation learning framework to learn instance segmentation with only image-level labels. It introduces bottom-up object cues to top-down pipeline and disentangles segmentation supervision from detectors.
- The segmentation branch cooperates self-supervised learning and self-training to model from class-agnostic foreground extraction to class-specific object segmentation progressively, while the detection branch utilizes off-the-shelf WSOD methods to mine object in the form of bounding boxes.
- We further propose instance-activation correlation module to enhance the coherence between detection and segmentation branches.

2. Related Work

Weakly Supervised Instance Segmentation (WSIS). WSIS can be categorized into two groups according to training supervision, i.e., instance- and image-level labels. The first group mainly utilizes bounding-box annotations to supervise instance segmentation models. Khoreva et al. [15] applied box-driven segmentation techniques for bounding boxes individually to generate pixel-level labels and exploited recursive training as a de-noising strategy. Hsu et al. [16] introduced multiple instance learning to generate pseudo-ground-truth masks, which used tightness prior of bounding boxes to build positive and negative bags. Li et al. [17] extended [15] to iteratively refine pseudo masks with segmentation predictions during training. Arun et al. [18] proposed a joint probabilistic learning objective and conditional distributions of pseudo masks for different levels of weak supervision. Cholakkal et al. [19] constructed object category density maps with the spatial distribution of object-counting information to learn WSIS.

The second group further challenges WSIS problem with only image-level weak supervision. The early work commonly explored bottom-up methods, which contain sophisticated multi-stage training procedures. PRM [4] and IAM [5] utilized class response maps to extract discriminative localization cues via back-propagation, which leveraged segmentation proposals to generate instance masks. IRNet [6] and WISE [7] propagated coarse localization cues from CAM [20] to discover the entire object, which is further regarded as pseudo-ground-truth masks to train fully supervised models. Recent WSIS methods drifted from bottom-up to top-down manner [12, 21, 22], which detects and segments all object instances sequentially. LabelPenet [12] developed multiple cascaded modules with curriculum learning strategy, which also relied on external models, i.e., Excitation BP [23], to compute segmentation masks at each stage. Kim et al. [21] proposed multi-task community learning to construct positive feedback loop and generated pseudo-ground-truth masks from CAM [20].

Unlike prevailing WSIS approaches, we propose a blender PDSL framework to learn top-down detection and
Figure 2: The figure illustrates the overall architecture of PDSL. The proposed PDSL consists of four components: backbone network, object detection, image segmentation and correlation learning modules.

3. The Proposed Method

3.1. Overall Framework

Given an image \( I \) and corresponding image-level labels \( t = [t_1, t_2, \ldots, t_{nc}] \) during training, WSIS aims to estimate segmentation masks for all object instances. Here \( t \) is an binary vector, where \( t_c = 1 \) denotes that image \( I \) contains the \( c^{th} \) target category, and otherwise, \( t_c = 0 \). And \( n_c \) is the number of target categories. In this paper, we propose a unified parallel detection-and-segmentation learning (PDSL) framework to learn instance segmentation with only image-level labels, which draws inspiration from both top-down and bottom-up instance segmentation approaches. As illustrated in Fig 2, the proposed PDSL consists of four modules: backbone network, object detection, image segmentation and correlation learning modules. The backbone network first outputs full-image feature maps of input \( I \). Then we use RoIPool [64] and RoIAlign [65] layers to extract pooled features for pre-computed object proposals.

bottom-up segmentation in parallel, which also captures activation-level interaction between detection and segmentation with correlation learning. The proposed PDSL gets rid of the sophisticated training procedures in the bottom-up approaches while achieving large performance improvement compared to previous top-down WSIS methods.

LIID [8] and S4Net [9] utilized graph partition algorithms to assign pseudo-ground-truth labels for segmentation proposals, which are used to train fully supervised models. However, LIID [8] and S4Net [9] required external instance-level segmentation models to compute salient instances as segmentation proposals, which contains additional ground-truth masks for training such segmentation.

Weakly Supervised Object Detection (WSOD). WSOD aims to predict object instance in the form of bounding boxes with weak supervision. WSDDN [24] selected proposals by parallel detection and classification branches in deep convolutional networks. Many work extended WSDDN [24] and leveraged contextual information [25, 10], attention mechanism [26] to suppress low-quality object proposals. Several different strategies to train the MIL model had been proposed in the literature [27, 28, 29, 30, 31, 10]. Work in [33, 34, 35, 36, 37] treated the top-scoring proposals as supervision to train multiple instance refinement classifiers. Other different strategies [38, 39, 40, 41, 42, 43] are also proposed to generate pseudo-ground-truth bounding boxes and assign labels to proposals. The above framework is further improved by min-entropy prior [44, 45], gradient information [46, 47], continuation MIL [48], utilizing uncertainty [49, 50, 51], generative adversarial learning [52], spatial likelihood voting [53], objectness consistent [54, 55] and deep residual learning [56]. Methods in [57, 58, 59] trained object detection systems from different supervisions.

Collaboration mechanism between object detection and semantic segmentation is proposed to take advantages of the complementary interpretations of weakly supervised tasks [60, 61, 62, 54]. However, those approaches aimed to improve detection results with segmentation guidance from full-image masks, which are produced from off-the-shelf segmentation models [63, 6] or additional segmentation branches. Moreover, they neglected the correlation relationship between detection and segmentation, and only reported detection performance. As demonstrated in [21], a straightforward combination of such two techniques failed to achieve competitive results for instance segmentation.
als [66, 67], which are the inputs of object detection and image segmentation branch, respectively. Specifically, a top-down detection branch mines object instances by classifying and refining object proposals, while the image segmentation branch explores bottom-up object cues to learn class-agnostic and class-specific segmentation within bounding boxes via self-supervised learning and self-training. As both branches process the same proposals, we further introduce a correlation learning module to enhance the coherence between detection and segmentation. During training, we have following objective function

\[ L = L_{OD} + L_{IS} + L_{CL}, \]  

where \( L_{OD} \) and \( L_{IS} \) are the loss functions of object detection and image segmentation, respectively. And \( L_{CL} \) is the loss function of correlation learning.

### 3.2. Object Detection Branch

We follow the multiple-instance learning [68] pipeline in deep convolutional networks and utilize two-stream WSDDN [24] algorithms for object detection branch. Given pooled features from RolPool [64] layer, we extract proposal features by two fully-connected layers, each of which is followed by ReLU activation and dropout layer. Then the proposal features are forked into two streams, i.e., classification and detection stream, to produce two score matrices \( C, D \in \mathbb{R}^{n^P \times n^C} \) by another two fully-connected layers, respectively, where \( n^P \) is the number of proposals. Both score matrices are normalized by softmax functions \( \sigma(\cdot) \) over categories and proposals, respectively. Finally, the element-wise product of the output of the two streams is again a score matrix: \( S^0 = \sigma(C) \odot \sigma(D^T)^T \). To acquire image-level multi-label classification scores, a sum pooling is applied: \( y_c = \sum_{p=1}^{n^P} S^0_{cp} \). Then we employ multi-label cross-entropy loss function to utilize image-level labels as

\[ L_{OM} = - \sum_{c=1}^{n^C} \left\{ t_c \log y_c + (1 - t_c) \log(1 - y_c) \right\}. \]  

To further reduce mis-localizations, we tweak the similar idea from OICR [35] to refine detection results via multiple detection refinement heads. To this end, each head has proposal classification and bounding-box regression subnets, which enables to refine both bounding-box scores and coordinates. In details, it produces classification scores \( S^r \in \mathbb{R}^{n^P \times (n^C + 1)} \) and new bounding boxes \( B^r \in \mathbb{R}^{n^P \times n^C \times 4} \) for the \( r \)th refinement head, where \( n^C + 1 \) indicates \( n^C \) object categories and 1 background category.

During training, for the \( r \)th head and the \( c \)th category that \( t_c = 1 \), the highest-score bounding box from previous prediction is selected as pseudo-ground-truth boxes and assigns positive/negative labels for each proposal. Thus, the corresponding objective function is

\[ L_{OR}^r = \sum_{p=1}^{n^P} y_{tp} \cdot \max\left(0, 1 - 1_{t_p \geq 0} \right) + \sum_{p=1}^{n^P} \max\left(0, 1 - 1_{t_p \geq 1} \right), \]  

where \( t^r \in \mathbb{R}^{n^P} \) and \( B^r \in \mathbb{R}^{n^P \times 4} \) are the classification and regression targets for object proposals in the \( r \)th head, respectively. \( L_{CE} \) is the softmax cross-entropy loss, and \( L_{smoothL1} \) is the smooth L1 loss [64]. The Iverson bracket indicator function \( [t^r_{tp} \geq 1] \) evaluates to 1 when \( t^r_{tp} \geq 1 \) and 0 otherwise. With above definition, the overall objective function for object detection module is

\[ L_{OD} = L_{OM} + \sum_{r=1}^{n^r} L_{OR}^r, \]  

where \( n^r \) is the number of detection refinement heads. During testing, the average output of all heads is used.

### 3.3. Image Segmentation Branch

Image segmentation branch aims to predict instances masks given bounding boxes in images. In fully supervised learning, image segmentation can directly learn ground-truth masks within positive bounding boxes [65]. However, recent top-down WSIS methods, i.e., WSJDS [10] and Label-PENet [12], back-propagated detection results to images to generate object heatmaps, which are then post-processed as supervision for segmentation learning. Thus, the quality of pseudo-ground-truth masks is strongly tied to the performance of object detection, which limits further improvement with large margins.

In this paper, we formulate image segmentation as foreground extraction via a progressive class-agnostic to class-specific strategy. Particularly, we first leverage self-supervised learning to learn class-agnostic foreground segmentation from bottom-up object cues within bounding boxes, which disentangles the pseudo-ground-truth mask generation from detection module. Then, the class-agnostic mask predictions are treated as supervision to learn class-specific segmentation branches via self-training. To this end, image segmentation module consists of multiple mask prediction heads with the same structure. In detail, image segmentation module has 4 convolutional layers with \( 3 \times 3 \) kernels and 256 channels to extract feature maps, which followed by a deconvolutional layer with \( 2 \times 2 \) kernels and \( n^m \) final prediction layer with \( 1 \times 1 \) kernels. Given a set of
object proposals, the objective function $L_{IS}$ is defined as

$$L_{IS} = \sum_{p=1}^{n^p} L_{BCE}(M_p^0, \hat{M}_p^0) + \sum_{c=1}^{n^c} \sum_{p=1}^{n^p} [t_{p}^{c}] \log(1 + \exp(-\tau M_{p}^{c} - \tau M_{p}^{c} - 1)) \quad (5)$$

where the first term and the last two terms are loss function of class-agnostic and multiple class-specific mask heads, respectively. And $L_{BCE}(M, \hat{M}) = -\sum_{i,j} M_{ij} \log \hat{M}_{ij} - (1 - \hat{M}_{ij}) \log(1 - M_{ij})$ is the binary cross-entropy loss. Specifically, $M_0^p$ and $M_0^c$ are the mask prediction and pseudo-ground-truth targets for the $p^{th}$ proposal in class-agnostic mask head. And $M_m^{pc}$ denotes the prediction masks for the $p^{th}$ proposal and the $c^{th}$ category in class-specific mask heads. Different to class-agnostic mask head, class-specific mask heads only compute losses for the categories existed in the images, which are weighted by the image-level classification scores $y_{t^p}$. To acquire initial pseudo-ground-truth masks $\hat{M}_0^0$ for class-agnostic mask head, we use unsupervised GrabCut [69] methods to extract bottom-up object cues as foreground segmentation given bounding boxes. We are not restricted with the algorithms that generate object cues from input images. However, extract foreground segmentation for all proposals is computationally inefficient during training. As a large number of proposals are necessary to achieve a reasonable recall rate and good performance. Recall that object proposals are always redundant and highly overlap each other, which makes their masks shareable. Thus, we introduce a seed sample acquisition strategy. Specifically, we first pick the highest-score object proposal for each category that appears in the category-label. We then sort the object proposals according to IoU overlaps with the selected proposal. After that, the top $n^\text{seed}$ proposals are sampled as seeds to estimate foreground segmentation for pseudo-ground-truth masks, where $n^\text{seed} \ll n^p$. Finally, the pseudo-ground-truth masks of the rest proposals are the same as that of seed proposals with the highest box IoU overlap. Despite its simplicity, our experiment shows that, even with a small $n^\text{seed} = 10$ for each category, the generated pseudo-ground-truth masks still enable class-agnostic mask head to learn high-quality foreground segmentation.

3.4. Correlation Learning Module

Theoretically, the loss functions of detection and segmentation task lead to complementary knowledge [10]. MIL-based WSOD explicitly penalizes all false positives, and counts a prediction as correct as long as it has IoU with ground truth over a threshold. This brings clean background with few false positives, but also lacks sensitivity to fine tune the localizer. On the other hand, for segmentation, the lack of explicit penalty on false positives often results in noisy background. But the fine granularity gives it better precision on ambiguous regions to guide the object localizer. So these two tasks complement well with each other. Our motivation of correlation learning is to exploit complementary knowledge learned from individual tasks by enhancing the coherence between detection and segmentation. Our ablation study also demonstrates that correlation learning is vital to achieve high performance for parallel detection-and-segmentation learning.

Although PDSL learns detection and segmentation in parallel, both modules are applied to feature maps cropped by the same proposals from fully-image feature maps. Thus, we design an instance-activation correlation learning with overall objective function formulated as

$$L_{ACL} = \sum_{m=1}^{n^m} L_{ACL}^m, \quad (6)$$

where $L_{ACL}^m$ is the loss functions of instance-activation correlations for the $m^{th}$ segmentation branch.

The instance-activation correlation learning requires consistent prediction activation between detection and segmentation for the same proposals. We first compute the proposal activation $a^m_p$ of the $p^{th}$ proposals and the $m^{th}$ mask head using Log-Sum-Exp [70] as

$$a^m_p = \frac{1}{\tau} \log \left( \frac{1}{hw} \sum \exp(\tau M_{pk}^m) \right), \quad (7)$$

where $h, w$ are the spatial size of prediction mask $M^m_{pc}$, and hyper-parameter $\tau = 5$ controls the smooth. The instance activation from detection module is the predicted scores from the last object refinement heads: $\hat{a}_p = S^{\text{ref}}_{p^m}$. Thus, the instance-activation correlation loss $L_{ACL}^m$ for the $m^{th}$ mask refinement head computes $p$-norm distance as

$$L_{ACL}^m = \frac{1}{n^p} \sum_{p=1}^{n^p} ||a^m_p - \hat{a}_p||_2. \quad (8)$$

4. Quantitative Evaluations

4.1. Datasets

We evaluate the proposed method on PASCAL VOC 2012 [13] and MS COCO [14]. PASCAL VOC 2012 consists of 20 target categories as well as a background category. We follow [4, 6, 8] to utilize the main trainval subset, excluding segmentation val images, to train our models. We evaluate our approach and baseline models using the 1,449
segmentation val images. MS COCO dataset consists of 80 target categories. We follow [9] to train on the standard train set and evaluate on the target categories. We follow [9] to train on the standard protocol to report the precision on Pascal VOC, we follow the standard PASCAL VOC average precision (AP) over IoU thresholds. For object detection refinement heads output bounding boxes for segmentation in parallel with detection. Thus, we also show the influence of different learning strategies, i.e., parallel and cascade learning, as plotted in Fig. 3. We compute the AP results for object detection and instance segmentation for each 2,000 iterations with single-scale testing. It is obvious that parallel learning provides superior performance for instance segmentation compared to cascade learning. As detection-to-segmentation multi-task cascade causes the quality of pseudo masks heavily relying on object detection.

4.4. Ablation Study

Before the comparison with other competitors, we perform several ablation studies to evaluate the effectiveness of different design choices and parameter settings. All ablation studies are conducted on the PASCAL VOC 2012 instance segmentation. Here, we use ResNet18-WS [56] as backbone to save time if not mentioned. When tuning each hyper-parameter, other parameters are kept as default.

Parallel vs. cascade learning strategies. Recall that previous top-down methods [10, 12] are based on multi-task cascade and utilized the gradient of detection results with respect to images to generate pseudo masks. The proposed PDSL method leverages correlation learning and self-supervised learning with bottom-up cues to model segmentation in parallel with detection. Thus, we also show the influence of different learning strategies, i.e., parallel and cascade learning, as plotted in Fig. 3. We compute the AP results for object detection and instance segmentation for each 2,000 iterations with single-scale testing. It is obvious that parallel learning provides superior performance for instance segmentation compared to cascade learning.

The number \( n' \) of detection refinement heads. The detection refinement heads output bounding boxes for segmentation during testing, which heavily influence the performance of instance segmentation. The hyperparameter \( n' \) in Eq. 4 controls the number of refinement heads. Different settings of \( n' \) are evaluated in Tab. 1. When we have \( n' = 0 \), the second term of loss function \( L_{OD} \) in Eq. 4 are omitted. We can see that the results of this setting are worse than using more refinement heads, demonstrating that the

---

**Figure 3:** Object detection and instance segmentation performance on VOC 2012 for parallel and cascade learning.

**Table 1:** Ablation study of PDSL on PASCAL VOC 2012 instance segmentation.

<table>
<thead>
<tr>
<th>( n' )</th>
<th>( n'' )</th>
<th>( L_{ASC} )</th>
<th>mAP</th>
<th>( mAP_{50} )</th>
<th>( mAP_{75} )</th>
<th>( mAP_{35} )</th>
<th>( mAP_{50} )</th>
<th>( mAP_{75} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>10</td>
<td>✓</td>
<td>8.7</td>
<td>31.8</td>
<td>2.5</td>
<td>0.6</td>
<td>2.4</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>10</td>
<td>✓</td>
<td>14.3</td>
<td>40.6</td>
<td>8.7</td>
<td>1.8</td>
<td>6.0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>10</td>
<td>✓</td>
<td>19.1</td>
<td>45.4</td>
<td>14.7</td>
<td>2.4</td>
<td>11.1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>10</td>
<td>✓</td>
<td>20.2</td>
<td>47.3</td>
<td>15.5</td>
<td>2.5</td>
<td>12.9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>10</td>
<td>✓</td>
<td>20.8</td>
<td>47.8</td>
<td>15.6</td>
<td>2.3</td>
<td>12.7</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>10</td>
<td>✓</td>
<td>20.8</td>
<td>47.8</td>
<td>15.6</td>
<td>2.2</td>
<td>12.7</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>✓</td>
<td>16.0</td>
<td>45.4</td>
<td>11.7</td>
<td>1.4</td>
<td>10.1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>✓</td>
<td>16.6</td>
<td>45.8</td>
<td>12.1</td>
<td>1.1</td>
<td>10.9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>✓</td>
<td>16.6</td>
<td>46.4</td>
<td>12.2</td>
<td>1.4</td>
<td>11.3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>✓</td>
<td>17.3</td>
<td>46.9</td>
<td>13.0</td>
<td>1.8</td>
<td>12.9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>✓</td>
<td>20.4</td>
<td>47.8</td>
<td>15.6</td>
<td>2.3</td>
<td>12.7</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>10</td>
<td>✓</td>
<td>20.8</td>
<td>47.9</td>
<td>15.0</td>
<td>2.1</td>
<td>13.3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>✓</td>
<td>15.1</td>
<td>38.5</td>
<td>9.3</td>
<td>1.2</td>
<td>7.8</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>✓</td>
<td>20.4</td>
<td>47.8</td>
<td>15.6</td>
<td>2.3</td>
<td>12.7</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>50</td>
<td>✓</td>
<td>21.0</td>
<td>47.8</td>
<td>15.6</td>
<td>2.6</td>
<td>12.9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>100</td>
<td>✓</td>
<td>20.9</td>
<td>48.0</td>
<td>15.0</td>
<td>2.3</td>
<td>12.7</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>10</td>
<td>✓</td>
<td>20.8</td>
<td>47.8</td>
<td>15.6</td>
<td>2.3</td>
<td>12.7</td>
</tr>
</tbody>
</table>

4.2. Evaluation Protocol

For the evaluation metrics of instance segmentation, we report the standard COCO metrics [14], which is mean average precision (AP) over IoU thresholds. For object detection on Pascal VOC, we follow the standard PASCAL VOC protocol to report the mAP at 50% Intersection-over-Union (IoU) of the detected boxes with the ground-truth ones. We also report CorLoc to indicate the percentage of images in which a method correctly localizes an object of the target category according to the PASCAL criterion.

4.3. Implementation Details

We implement our method using the PyTorch framework. All backbones are initialized with the weights pre-trained on ImageNet ILSVRC [71]. We use synchronized SGD training on 4 GPUs. A mini-batch involves 1 images per GPU. We use a learning rate of 0.01, momentum of 0.9, and dropout rate of 0.5. We use a step learning rate decay schema with decay weight of 0.1 and step size of 70,000 iterations. The total number of training iterations is 100,000. We adopt 240,000 training iterations for MS COCO. In the multi-scale setting, we use scales range from 480 to 1216 with stride 32. To improve the robustness, we randomly adjust the exposure and saturation of the images by up to a factor of 1.5 in the HSV space. A random crop with 0.9 of the size of the original images is applied. We use MCG [67] to generate object proposals for all experiments, including our implementation of baseline methods. We set the maximum number of proposals in an image to be 2,000. The test scores are the average of scales of {480, 576, 672, 768, 864, 960, 1056, 1152} and flips. Detection results are post-processed by NMS with threshold of 0.5. We use the following parameter settings in all the experiments, unless specified otherwise. We set the both hyper-parameters \( n' \) and \( n'' \) in Eq. 4 to 4. For the seed sample acquisition strategy, we set the number of sampled proposals \( n_{seed} \) to 10.
The number \( m^\text{seed} \) of sample acquisition strategy. We continue by evaluating the effect of \( m^\text{seed} \). Recall that we only sample \( m^\text{seed} \) proposals to estimate foreground segmentation using unsupervised traditional methods. And generating such pseudo-ground-truth masks is time-consuming during training. Thus, we seek to balance the quality of pseudo-ground-truth masks and training speed by tuning hyper-parameter \( m^\text{seed} \). As shown in Tab. 1, when only the highest-score proposal is used, \( m^\text{seed} = 1 \), the quality of learned masks drop dramatically. We can observe that compared with \( m^\text{seed} = 1 \), even just using only 10 object proposals boosts the performance a lot, which confirms the effectiveness of leveraging object cues for segmentation module. We also find that more seed proposals help to improve pseudo-ground-truth masks for class-agnostic mask head. We set the default values for \( m^\text{seed} \) as 10, which provides a good balance between final performance and training time.

The instance-activation correlation learning loss \( L_{\text{ACL}} \). To understand the importance of the correlation

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervision</th>
<th>Backbone</th>
<th>( m_{\text{AP0.25}} )</th>
<th>( m_{\text{AP0.50}} )</th>
<th>( m_{\text{AP0.75}} )</th>
<th>( m_{\text{AP}_{S}} )</th>
<th>( m_{\text{AP}_{M}} )</th>
<th>( m_{\text{AP}_{L}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask-R-CNN [65]</td>
<td>( \mathcal{M} )</td>
<td>ResNet101</td>
<td>76.7</td>
<td>67.9</td>
<td>52.5</td>
<td>44.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Khoreva et al.</td>
<td>( B )</td>
<td>VGG-16</td>
<td>-</td>
<td>44.8</td>
<td>-</td>
<td>16.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cholakkal et al.</td>
<td>( C )</td>
<td>ResNet-50</td>
<td>48.5</td>
<td>30.2</td>
<td>14.4</td>
<td>8.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hsu et al.</td>
<td>( B )</td>
<td>ResNet-101</td>
<td>75.0</td>
<td>58.9</td>
<td>30.4</td>
<td>21.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arun et al.</td>
<td>( B )</td>
<td>ResNet-101</td>
<td>73.8</td>
<td>58.2</td>
<td>34.3</td>
<td>32.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Comparison with the state-of-the-art methods on PASCAL VOC 2012 instance segmentation. The terms \( \mathcal{M}, \mathcal{B}, \mathcal{C} \) and \( \mathcal{I} \) denote pixel-level, bounding-box-level, object-count and image-level labels, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervision</th>
<th>Backbone</th>
<th>( m_{\text{AP0.25}} )</th>
<th>( m_{\text{AP0.50}} )</th>
<th>( m_{\text{AP0.75}} )</th>
<th>( m_{\text{AP}_{S}} )</th>
<th>( m_{\text{AP}_{M}} )</th>
<th>( m_{\text{AP}_{L}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask-R-CNN [65]</td>
<td>( \mathcal{M} )</td>
<td>ResNet101</td>
<td>76.7</td>
<td>67.9</td>
<td>52.5</td>
<td>44.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Khoreva et al.</td>
<td>( B )</td>
<td>VGG-16</td>
<td>-</td>
<td>44.8</td>
<td>-</td>
<td>16.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cholakkal et al.</td>
<td>( C )</td>
<td>ResNet-50</td>
<td>48.5</td>
<td>30.2</td>
<td>14.4</td>
<td>8.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hsu et al.</td>
<td>( B )</td>
<td>ResNet-101</td>
<td>75.0</td>
<td>58.9</td>
<td>30.4</td>
<td>21.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arun et al.</td>
<td>( B )</td>
<td>ResNet-101</td>
<td>73.8</td>
<td>58.2</td>
<td>34.3</td>
<td>32.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Comparison with the state-of-the-art methods on MS COCO instance segmentation. The terms \( \mathcal{M}, \mathcal{B}, \mathcal{C} \) and \( \mathcal{I} \) denote pixel-level, instance saliency and image-level labels, respectively.

- The number \( m^\text{m} \) of class-specific mask heads. To analyze how AP varies with the number of segmentation refinement, we range hyper-parameter \( m^\text{m} \) in Equ. 5 from 0 to 5 When we have \( m^\text{m} = 0 \), the second and third terms of loss function \( L_{\text{IS}} \) in Equ. 5 are omitted, as we only have class-agnostic mask head. In Tab. 1, the performances are improved consistently for all metrics with hyper-parameter \( m^\text{m} \) increasing. For \( m^\text{m} \geq 4 \), the performance improvement is relatively small. Therefore, we set \( m^\text{m} \) to 4 by default.

- The number \( n^\text{seed} \) of sampled proposals for seed sample acquisition strategy. We continue by evaluating the effect of hyper-parameter \( n^\text{seed} \). Recall that we only sample \( n^\text{seed} \) proposals to estimate foreground segmentation using unsupervised traditional methods. And generating such pseudo-ground-truth masks is time-consuming during training. Thus, we seek to balance the quality of pseudo-ground-truth masks and training speed by tuning hyper-parameter \( n^\text{seed} \). As shown in Tab. 1, when only the highest-score proposal is used, \( n^\text{seed} = 1 \), the quality of learned masks drop dramatically. We can observe that compared with \( n^\text{seed} = 1 \), even just using only 10 object proposals boosts the performance a lot, which confirms the effectiveness of leveraging object cues for segmentation module. We also find that more seed proposals help to improve pseudo-ground-truth masks for class-agnostic mask head. We set the default values for \( n^\text{seed} \) as 10, which provides a good balance between final performance and training time.

- The instance-activation correlation learning loss \( L_{\text{ACL}} \). To understand the importance of the correlation
learning, we evaluate the influence of different loss functions in this module. The instance-activation correlation loss $L_{ACL}$ punishes the activation diversity of the same proposals between detection and segmentation modules. As we can see in Tab. 1, results can be improved by instance-activation correlation learning.

### 4.5. Comparison with State of the Arts (SOTAs)

We comprehensively evaluate our method with three ResNet-WS [56] backbones in our experiments, which are variation of ResNet [73]. It should be noted that our default hyper-parameters are not the best setting according to Tab. 1. Comparisons with recent state-of-the-art methods on PASCAL VOC 2012 and MS COCO are listed in Tab. 2 and 3. Some previous methods achieve high performance, thanks to the specially designed inter-pixel relation module [6], graph partition algorithm [8, 9] and salient detector [9, 8]. Unlike previous methods [7, 6, 4, 5], our PDSL does not rely on sophisticated and multiple sequential training process, i.e., fully-supervised model retraining [7, 6] and class activation map module [6, 7] and segmentation proposals [4, 5]. Instead, we utilize a powerful unified parallel detection-and-segmentation learning framework and correlation learning module to capture intra-modular and inter-modular dependencies across separate branches. Thus, we achieve consistent accuracy gain over existed methods and set the new state-of-the-art results.

The qualitative results on the PASCAL VOC 2012 val are shown in Fig. 4. As can be observed in the first five columns, our approach outputs semantically meaningful and precise predictions despite the existence of complex object appearances and challenging background contents. It demonstrated the effectiveness of the proposed unified parallel detection-and-segmentation learning framework. We further visualize our failure mode in the last column, mainly resulting from confusion with similar objects, localization error and failing to distinguish multiple instances.

Each iteration of PDSL with ResNet50 takes about 1 second for forward-backward propagation on GTX 1080Ti GPUs and several seconds on CPUs to extract bottom-up object cues. Thus, the total training times are about 12 days for PASCAL VOC. Noted that parallel GPU computing of bottom-up cues can further reduce training time. We can also use pre-computed GrabCut masks, segmentation proposals and attention maps as bottom-up object cues. During inference, PDSL runs at 539 ms per image including NMS.

### 5. Conclusion

In this paper, we propose a unified parallel detection-and-segmentation learning (PDSL) framework to learn instance segmentation with only image-level labels, which draws inspiration from both top-down and bottom-up instance segmentation approaches. In order to improve the coherence between detection and segmentation branches, we further propose instance-activation correlation learning, which impose a high correlation on activation between two branches for the same object instance. For the first time, we show a top-down WSIS approach could deliver the state-of-the-art results on both PASCAL VOC and MS COCO.

### 6. Acknowledgment

This work is supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No.U1705262, No.62072386, No.62072387, No.62072389, No.62002305, No.61772443, No.61802324 and No.61702136), Guangdong Basic and Applied Basic Research Foundation (No.2019B1515120049), the Fundamental Research Funds for the central universities (No.20720200077, No.20720200090 and No.20720200091) and the Beijing Nova Program under Grant Z201100006820023.
References


[31] Xinggang Wang, Zhuotun Zhu, Cong Yao, and Xiang Bai. Relaxed Multiple-Instance SVM with Application to Object Discovery. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3


[57] Linpu Fang, Hang Xu, Zhili Liu, Sarah Parisot, and Zhenguo Li. EHSOD: CAM-Guided End-to-end Hybrid-Supervised Object Detection with Cascade Refinement. In AAAI Conference on Artificial Intelligence (AAAI), 2020. 3


[61] Xiaoyan Li, Meina Kan, Shiguang Shan, and Xilin Chen. Weakly Supervised Object Detection with Segmentation Collaboration. In IEEE International Conference on Computer Vision (ICCV), 2019. 3


[64] Ross Girshick. Fast R-CNN. In IEEE International Conference on Computer Vision (ICCV), 2015. 3, 4

[65] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In IEEE International Conference on Computer Vision (ICCV), 2017. 3, 4, 7


