

# Rethinking Localization Map: Towards Accurate Object Perception with Self-Enhancement Maps

Xiaolin Zhang, Yunchao Wei, Yi Yang, and Fei Wu

**Abstract**—Recently, remarkable progress has been made in weakly supervised object localization (WSOL) to promote object localization maps. The common practice of evaluating these maps applies an indirect and coarse way, *i.e.*, obtaining tight bounding boxes which can cover high-activation regions and calculating intersection-over-union (IoU) scores between the predicted and ground-truth boxes. This measurement can evaluate the ability of localization maps to some extent, but we argue that the maps should be measured directly and delicately, *i.e.*, comparing the maps with the ground-truth object masks pixel-wisely. To fulfill the direct evaluation, we annotate pixel-level object masks on the ILSVRC [1] validation set. We propose to use IoU-Threshold curves for evaluating the real quality of localization maps.

Beyond the amended evaluation metric and annotated object masks, this work also introduces a novel self-enhancement method to harvest accurate object localization maps and object boundaries with only category labels as supervision. We propose a two-stage approach to generate the localization maps by simply comparing the similarity of point-wise features between the high-activation and the rest pixels. Based on the predicted localization maps, we explore to estimate object boundaries on a very large dataset. A hard-negative suppression loss is proposed for obtaining fine boundaries. We conduct extensive experiments on the ILSVRC and CUB [2] benchmarks. In particular, the proposed Self-Enhancement Maps achieve the state-of-the-art localization accuracy of 54.88% on ILSVRC. The code and the annotated masks are released at <https://github.com/xiaomengyc/SEM>.

**Index Terms**—Weakly Supervised Learning, Object Localization Maps, Convolutional Neural Network

## 1 INTRODUCTION

LOCATION maps (*a.k.a* class activation maps) are profoundly studied in recently years [3]–[5]. It is originally proposed to visualize the high-level activations of classification networks [3], where object regions are expected to have higher activation scores while background regions have lower scores. Afterward, localization maps are found to be greatly useful in Weakly Supervised Object Localization (WSOL) [3]–[7] tasks. Generally, WSOL employs a post-processing step to infer object bounding boxes from the generated object localization maps. To be specific, the post-processing step first binarizes the localization maps with fixed thresholds and then obtains the object bounding box by drawing rectangles over the largest connected area. To achieve superiority performance on WSOL, researchers seek methods for increasing the quality of localization maps and conducting the evaluation by calculating the accuracy of the inferred bounding boxes [4]–[6], [8]. Therefore, the quality of the generated localization maps is actually measured via an indirect way with the post-processed bounding boxes.

However, such an indirect measurement is not perfect for evaluating the real accurateness of localization maps. Figure 1a, 1b and 1c show three situations where the current indirect measurement fails to rightly evaluate the acquired localization maps. Ground-truth and inferred bounding boxes are shown in red and green rectangles, respectively. First, localization maps correctly highlight the target regions, but the indirect metric considers them as false positives, as shown in Figure 1a; Second, the maps are successful in highlighting the most important parts of the target

objects, but the localization criterion gives it zero credit as the Intersection-over-Union (IoU) is smaller than 50% as shown in Figure 1b; Third, although the predicted boxes accurately match the ground-truth boxes, the localization maps involve many noises or fail to highlight important areas of target objects as shown in Figure 1c. (We show more such cases in the supplementary material.) These observations reveal a significant problem that *the current indirect evaluation metric is not flawless to truly reflect the correctness of the localization maps generated by WSOL algorithms*. Therefore, we consider that it is necessary to perform a precise measurement for evaluating the quality of localization maps in a *direct* way.

By annotating pixel-level object masks on the most widely applied dataset, *i.e.*, ILSVRC [1], [9], we propose a direct measurement to pixel-wisely compare the object-related partitions with our annotated ground-truth masks for evaluating the real and deliberate ability in highlighting target objects. In particular, we calculate the object Intersection-over-Union (IoU) *w.r.t.* different thresholds for splitting the object pixels from the background. The highest IoU values are considered to be the best potential in localizing object regions. Figure 1d depicts the IoU-Threshold curves of several recently proposed methods, *i.e.*, CAM [3], HaS [7], ACoL [4], SPG [5] and CutMix [10]. These methods all aim at achieving better WSOL performance by promoting the quality of the produced localization maps. For example, HaS [7] and CutMix [10] propose to hide some pitches of the input images to force classification networks to highlight more robust discriminative regions. ACoL [4] erases the regions with the highest scores in high-level feature maps, and re-learn more object pixels by feeding the erased features into an auxiliary branch. SPG [5] learns a stage-wise binary salient map for each image using a side branch, and injects the pixel-level correlation into the classification network using an auxiliary loss function. According

X. Zhang, Y. Wei, Y. Yang are with the Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia. F. Wu is with the College of Computer Science, Zhejiang University, Hangzhou, China. ( e-mail: Xiaolin.Zhang-3@student.uts.edu.au, Yunchao.Wei@uts.edu.au, Yi.Yang@uts.edu.au, Wufei@cs.zju.edu.cn )

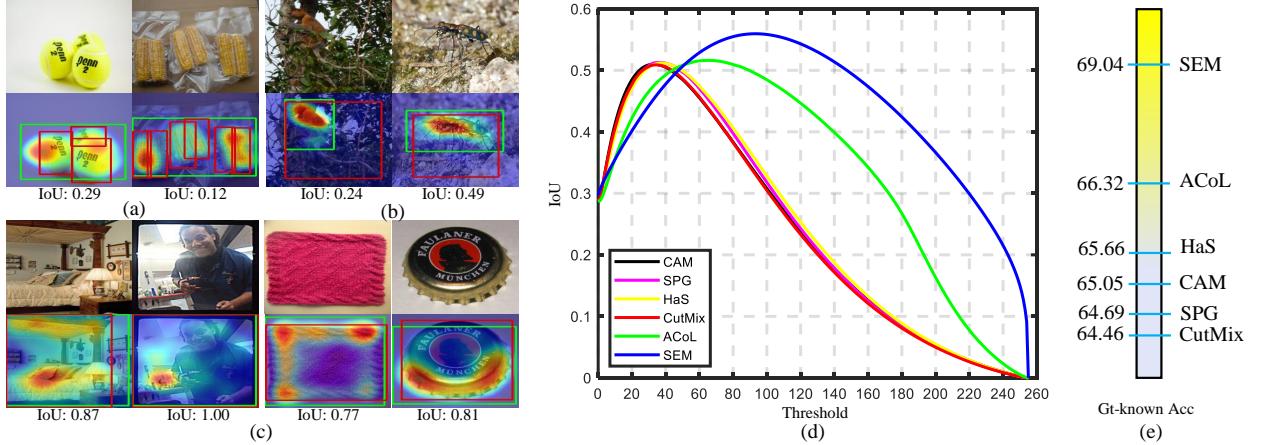


Fig. 1. Three scenarios where the current indirect evaluation metric fails to measure localization maps: (a) Localization maps accurately highlight target objects, while the metric considers them as false positive; (b) main parts of the target objects are highlighted, while the metric gives them zero credit; (c) although the predicted bounding boxes are accurate, the localization maps fail to highlight important areas of the target objects. (Ground-truth boxes are in red, and the predicted are in green.) (d) The proposed IoU-Threshold curves of various approaches for direct pixel-wise measurement on our annotated test set. Our Self-Enhancement Maps (SEM) achieve the best in both IoU and Threshold at the peak point. (e) The Gt-known localization accuracies w.r.t. different approaches. The proposed SEM achieves the best accuracy.

to the proposed direct evaluation metrics, we are surprised to find that these methods based on CAM, *i.e.*, HaS, CutMix, and SPG, which are claimed to improve the quality of localization maps, actually only make marginal progress. It turns out that *the results of the indirect measurement are not consistent with the proposed direct evaluation metric*. Especially, in the proposed pixel-wise measurement, the performance rank of these methods is ACoL > HaS > CAM > SPG > CutMix, while the claimed performance rank *w.r.t.* the bounding box accuracy is HaS > ACoL > CutMix > SPG > CAM, as illustrated in Figure 1d and 1e. This observation supports the idea that the current indirect measurement is not perfect, and we should augment it by applying the finer pixel-wise evaluation.

Except for the amendment in evaluating localization quality, the proposed direct measurement, *i.e.*, IoU-Threshold curve, can also measure the visualization effect, which is another crucial property of localization maps. Particularly, in the IoU-Threshold curve, *e.g.*, Figure 1d, the threshold *w.r.t.* the highest IoU point can reflect the visual effect of the target object in the localization maps. The higher threshold values indicate the higher brightness of the object areas with the largest IoU scores, which presents better visual effects. We notice the values of the best thresholds are rather small (<65 in the scale of [1,255]) with the localization maps produced by the existing methods. Such small values can not bring enough brightness when visualizing these localization maps.

In order to overcome the two disadvantages of localization maps, *i.e.*, low accuracy and low brightness of object areas, we propose a simple yet effective enhancement strategy. Intuitively, within one object, pixels with similar appearances usually share similar features in their embedding space. We can find the most discriminative areas and their corresponding features. These features can naturally be applied to capture more object regions, leading to better localization maps accordingly. In particular, given a trained classification network, we first extract object localization maps using the CAM variant proposed in [4]. Then, we identify a set of discriminative seeds by ranking the scores in the obtained localization maps and extract their corresponding feature vectors from high-level feature maps. Finally, by performing a self-enhancement process of calculating the cosine similarity between the seed vectors and the rest, we can simply obtain the refined

localization maps, named Self-Enhancement Maps (SEM). Surprisingly, the generated SEM can not only improve the localization accurateness in IoU (from 50.13% to 55.67%), but also promote the visual effect in increasing threshold of peak points from 33 to 93 after applying SEM to CAM. Our SEM method also significantly promote localization maps from 62.68% to 69.04% in the indirect evaluation method of measuring the inferred bounding boxes, surpassing the runner-up, *i.e.*, HaS by 2.64%. Besides, with the assistance of SEM, we advance the WSOL a step further by exploring to predict much finer details, *i.e.*, object boundaries, in a weakly supervised manner. Concretely, we generate the pseudo object boundaries from SEM, which are further employed as supervision to train the boundary detection models. As far as we know, this is the first feasible attempt to date, which can reversely validate the effectiveness of our SEM in capturing fine details of the target object.

To sum up, we rethink the WSOL from three aspects in this work and make contributions accordingly:

- *Is the current evaluation metrics flawless?* No. We propose a new metric, *i.e.*, IoU-Threshold curve, to augment the evaluation criteria of WSOL for a fairer comparison. To fulfill this, we manually annotate pixel-level masks on the ILSVRC validation set to realize the amended evaluation metrics.
- *Can we enhance localization maps of the existing methods by a simple yet effective way?* Yes. Existing methods based on CAM, *i.e.*, SPG, HaS, CutMix and ACoL, only make marginal progress under our precise pixel-level evaluation metric. Also, the thresholds *w.r.t.* the peak IoU point are small, which will result in the low brightness in visualization. We propose a method *i.e.*, Self-Enhancement Map (SEM), to enhance localization maps by employing feature similarities. The proposed SEM can significantly improve both the object accuracy and brightness in localization maps.
- *Can we apply localization maps to acquire more details of target objects rather than rough locations?* Yes. We explore the potential of our SEM in predicting object boundaries. To the best of our knowledge, this is the first feasible attempt to predict object boundaries using only image-level labels as supervision on a large-scale dataset.

## 2 RELATED WORK

**Weakly supervised object localization** aims to predict object positions given only image-level labels as supervisions. The common practice is to extract localization maps from classification networks [3], [11]–[14], and the localization maps can be applied as an alternative cheaper way for obtaining tight bounding boxes of target objects. CAM [3] is the most widely acknowledged method for generating class-specific localization maps. GradCAM [12], [15] uses the gradients of target concept. The gradients flow into the final convolutional layers to produce a coarse localization map highlighting important regions in the image. MWP [11] takes a top-down strategy and probabilistic winner-take-all process to generate rough object locations. Based on CAM, object erasing methods [4], [7], [16], [17] are proposed to cover more integral object regions by forcing the networks to learn more object patterns. Specifically, HaS [7] is proposed to erase some parts within the input images randomly. CutMix [10] adopts a different strategy of filling the erased area with other image patches and mixes the corresponding classification labels while training to augment the input images. Wei *et al.* [16] adopt an iterative approach to erase a small part of the most discriminative regions discovered by a trained classification network. ACoL [4] improves the efficiency by getting the localization maps online and erasing discriminative regions within high-level feature maps. ADL [6] further promotes the localization maps by applying dropout on multiple intermediate feature maps. Besides the methods based on the erasing operation, SPG [5] tries to learn pixel-level correlations by applying salient masks as the auxiliary supervision to increase the quality of localization maps. DANet [8] adopts a divergent activation method for learning better localization maps.

**Weakly supervised semantic segmentation** (WSSS) aims to predict precise pixel-level object masks using weak annotations which are divided into four groups, *i.e.*, bounding boxes [18]–[20], scribbles [21]–[23], points [24] and image-level labels [25]–[28]. Particularly, segmentation methods based on object bounding boxes basically apply unsupervised methods to obtain pseudo-masks and then train segmentation networks using the inferred pseudo-masks. Methods supervised by scribble lines generally try to improve the segmentation accuracies by mining more pixels that have similar features with the annotated pixels. For example, ScribbleSup [21] adopts an alternating solution to learn a decent segmentation network. It applies a graphical model to expand the discovered pixels and generate better pseudo-masks while uses the masks as supervision to train segmentation networks. Point-based methods use even weaker supervision to learn segmentation models. PointSup [24] performs metric learning upon annotated pixels of the same categories across images so that objects with different content meaning can be better found. Our WSOL shares much similarity with the segmentation methods based on using image-level labels as supervision. Both two tasks train classification networks for getting object localization maps. Differently, WSSS is a final task for pixel-wise classification. Localization maps usually serve as intermediate outputs to provide some initial cues of object locations in WSSS tasks. In the main time, WSSS often applies multiple sophisticated techniques for getting better pseudo masks so that reliable segmentation models can be learned.

**Weakly supervised object detection** (WSOD) aims at drawing tight bounding boxes of the target objects using only image-level supervision. Different from our WSOL task, WSOD considers a more challenging scheme where multiple objects of different semantics and scales may be distributed anywhere in the given

image. These methods generally include two aspects, *i.e.*, 1) estimating pseudo object bounding boxes, and 2) training the object detector using the off-the-shelf object detection approaches. Particularly, these methods can be roughly divided into two types according to the training manners, *i.e.*, alternating training [29]–[32] and end-to-end training [33]–[37]. The alternating training approaches usually need first to mine positive object proposals, and then optimize detection networks towards these bounding boxes. For instance, OM [31] first uses a mask-out strategy to collect class-specific object proposals, and then apply multiple instances learning to mine confident candidates. These candidates are employed as supervision to train regression networks. FV [29] implements a multi-fold multiple instance learning procedure, which prevents alternative training from prematurely locking onto erroneous object locations. HCP [38] develops a self-taught learning method to select more reliable seed positive proposals, and these proposals are further used to learn better detectors [30]. The alternating approach usually updates detectors and object bounding boxes alternately to lift the detection performance progressively. For the end-to-end training approaches, they assemble the pseudo object bounding box generation and object detection into a unified framework. For example, WSDDN [33] implements two branches for classifying object categories and regressing object bounding boxes. The outputs of the two branches are combined to judge the correctness of the predicted boxes and classes. MELM [39] introduces a min-entropy function to measure the randomness of object locations during training, which can reduce the variance of positive instances. A recurrent learning algorithm is further applied to transfer weak supervision information to object locations progressively. WSOD<sup>2</sup> [40] proposes a joint strategy of considering bottom-up and top-down objectness to estimate the objectness scores of the regressed bounding boxes using an adaptive linear combination approach so that the correct bounding boxes and class labels can be selected.

**Object boundary detection** is a fundamental problem in understanding images. Various edge detectors have been proposed and achieve many successes, including supervised methods, weakly supervised methods, and unsupervised methods. Canny [41] is the most widely used unsupervised detector because of its efficiency. Based on the recent convolutional networks, HED [42] applies multiple intermediate feature maps for predicting edges. RCF [43] and BDCN [44] adopt similar ideas to incorporate multiscale feature maps for the detectors. Khoreva *et al.* [45] propose a weakly supervised method for detecting object boundaries. They infer the object boundaries with the help of segmentation masks and object bounding box proposals produced by unsupervised segmentation methods and object detectors. However, all of these methods target on small datasets. SOBD [46] proposes to predict object boundaries on a very large dataset, *i.e.*, ILSVRC, by learning many detectors for different situations, *e.g.*, predicting one edge detector for each category. Different from the above methods, we propose a unified approach to predict the boundaries on ILSVRC with only image-level labels as the supervision. To the best of our knowledge, this approach is the first method trying to predict object boundaries for such a large dataset to date.

## 3 METHODOLOGY

In this section, we first provide the details of the new evaluation metric, which facilitates localization maps to be evaluated in a *direct* manner (Section 3.1). We then propose a simple yet

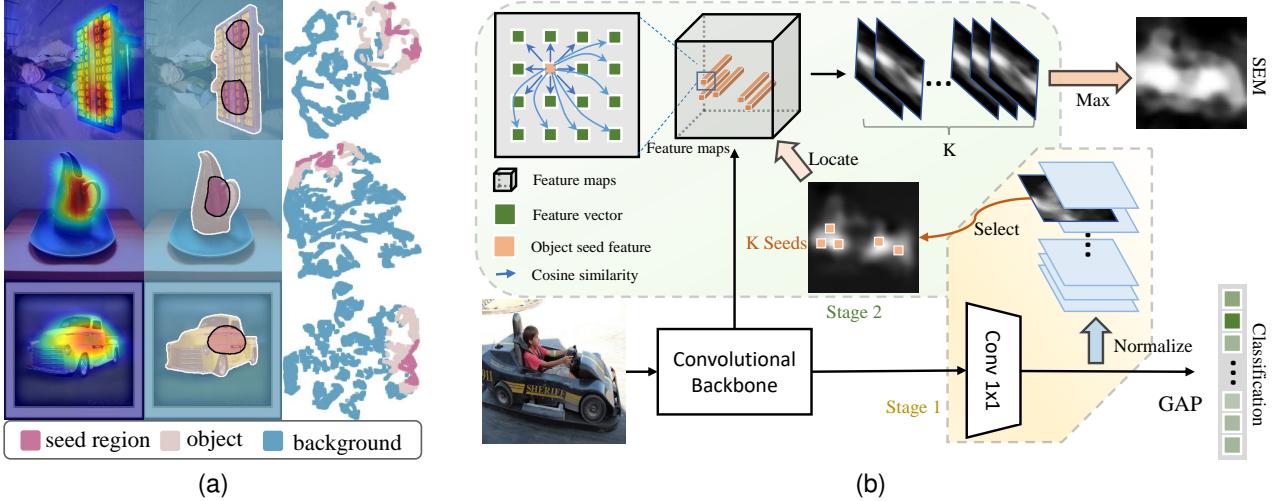


Fig. 2. (a) t-SNE [47] visualization of high-level features. Features of seed regions are close to the rest object regions while distant to the background; (b) The framework of SEM. In Stage 1, localization maps are extracted from the last-layer feature maps in a classification network. In Stage 2,  $K$  seeds with the highest scores are firstly selected. Then, we calculate the similarity between the seeds and the rest pixels for getting  $K$  similarity maps. We finally obtain the enhanced maps by computing element-wise maximum values among the  $K$  maps.

effective self-enhanced approach to promote the quality of localization maps (Section 3.2). Benefiting from the high-quality object localization maps produced by the proposed approach, we further explore to harvest more details of detecting object boundaries in a weakly supervised manner, which is the first attempt on a large dataset (Section 3.3).

### 3.1 Evaluation Metric Amendment

Due to the lack of object masks, the current bounding-box-based evaluation metric alternatively adopts an *indirect* way to examine the quality of localization maps. However, as shown in Figure 1a, 1b and 1c, such an *indirect* metric is not perfect to truly reflect the real quality of localization maps. The concurrent work [48] also notice this drawback of the indirect measurement, but [48] only applies precision and recall properties to reflect the quality of localization maps. The most accurate manner for evaluating localization maps is to compare the maps to the ground-truth masks at every pixel in a *direct* way. To fulfill the purpose of the accurate and direct evaluation, we annotate object masks on the ILSVRC [1] validation set.

The value of each pixel on localization maps represents the probability of belonging to target objects. The probability scores are continuous values, and thereby, we adopt a dense binarization strategy to evaluate the IoU scores at every binarized threshold. Particularly, we first use many equally spaced thresholds to binarize the localization maps. Then, we calculate the IoU scores between the binarized maps and their ground-truth masks. To this end, a high-quality localization map should meet two requirements: 1) the entire object region can be accurately extracted with a specific threshold; 2) brightness values of pixels belonging to object and the background should differ greatly so that the objects can be well visualized. To present the two properties, we propose to apply IoU-Threshold curves for evaluation, as shown in Figure 1d. Peak-IoU  $\in [0, 1]$  and Peak-T  $\in [0, 255]$  denote the best IoU score and its corresponding threshold, respectively. If a model generates better localization maps, the peak point in the corresponding curve should drift towards the upper-right corner of the figure. In other words, higher Peak-IoU and Peak-T values indicate better localization maps.

Based on this evaluation metric, we find that the recent efforts [4]–[6], [10] built on CAM [3] for boosting the quality of localization maps have only made marginal progress even with more sophisticated structures and extra training resources. Another disadvantage of these methods is that the maps perform poorly in visualization. The object regions do not have high brightness and cannot show enough distinctions between objects and background regions. In order to increase localization ability and boost the visualization effect, we find a more promising way to enhance these methods in a training-free way, which will be introduced in the next subsection.

### 3.2 Self-Enhancement Map

Object localization maps can highlight the object regions of interest while suppressing the values of background pixels [3], [11]. The best-known method, CAM [3], applies a category-wise fully connected layer to aggregate high-level feature maps so that class-specific localization maps can be harvested. These maps are typically employed as a coarse indication of the target object positions. Nevertheless, these localization maps can only identify several small and sparse object regions while hard to highlight the entire object, which deviates from the requirement of WSOL. Most previous efforts [4]–[6] reach a consensus that this issue is caused by the inconsistent discriminative ability of features from objects. However, even many promising solutions [4]–[6] are proposed to tackle this issue, the achievements are actually very limited based on the *direct* evaluation metric shown in Figure 1d.

To improve the quality of localization maps, we conduct the following analysis. First, we employ CAM [3] to produce localization maps and divide all pixels into three groups, *i.e.*, seed regions, the rest object regions *wrt* ground-truth masks, and background regions. The seed regions are pixels with scores greater than 0.7 (value chosen in SPG [5]). Then, we extract the corresponding high-level features and visualize their distributions using t-SNE [47]. As shown in Figure 2a, although some object regions do not have high activation values in the localization maps, they are still more similar to the discovered seed regions than the background regions in the embedding feature space. This observation motivates us that high-quality localization maps can

be simply obtained by considering the pixel-wise similarity in the embedding space. To this end, we propose Self-Enhancement Map (SEM), which can truly improve the localization ability in a much simpler yet more effective way.

Concretely, the proposed SEM is a two-stage method, which can be used in conjunction with the current state-of-the-art WSOL approaches [3], [5], [7], [10]. For simplicity, we adopt CAM [3] as the first stage method of our SEM. Given a trained classification network, we extract the first-stage localization maps using the method in ACoL [4], which is a simple variant of CAM. Since the convolutional operations for generating the localization maps can relatively preserve positions of input pixels, the feature vectors with high activation scores are often the most discriminative and best features representing the target objects. Therefore, in the second stage, we first identify the most discriminative seeds by ranking the scores in the obtained localization maps. Then, the corresponding feature vectors of the seeds are extracted from high-level feature maps. We acquire similarity maps by calculating the cosine similarity between the seed vectors and the rest pixels in the high-level feature space. To increase the robustness, we apply multiple object seeds to generate similarity maps, which are employed to produce the final self-enhanced localization maps with pixel-wise maximum aggregation. Despite the simplicity of our SEM, it turns out to be good at increasing both localization ability and visual effect.

Figure 2b depicts the workflow of the proposed SEM method. Given an input image  $I$  with its class label  $y \in [0, Y - 1]$ , e.g., *car*, we forward  $I$  through the classification network yielding the high-level feature maps  $F$ , where  $Y$  is the total number of categories. In the first stage, we adopt the method in [4] to obtain the first-stage maps which are proved to be the same with CAM [4]. Particularly, we apply a convolution operation as the last layer to produce class-specific feature maps which are denoted as  $M' \in \mathbb{R}^{Y \times W \times H}$ , where  $W$  and  $H$  are the width and height of the maps. We select the  $y_{th}$  feature map  $M'_y$  from the feature maps to get the localization map of the category  $y$ . We obtain the normalized maps  $M_y \in [0, 1]^{W \times H}$  by following  $M_y(i, j) = \frac{M'_y(i, j) - \min(M'_y)}{\max(M'_y) - \min(M'_y)}$ , where  $i$  and  $j$  are the indices of the map. To distinguish the localization map produced by CAM from SEM, we denote the normalized localization map of category  $y$  as  $M_{cam}$  from here. In the second stage, we choose object seed pixels whose scores in localization maps are ranked at the top  $K$ , where  $K \in [1, W \times H]$  is the number of the seeds. The corresponding feature vectors of the seeds can be extracted from the feature maps  $F$  according to the positions of the seed pixels. For each seed vector, we calculate the cosine similarity with every pixel in  $F$ , getting  $K$  similarity maps  $\{S_k | k = 0, 1, \dots, K - 1\}$ . We obtain the final self-enhancement localization maps  $M_{sem}$  by getting the maximum values among the  $K$  maps at each position, i.e.,  $M_{sem}(i, j) = \max(S_k(i, j))$ ,  $k = 0, 1, \dots, K - 1$ . Figure 9 compares the localization maps between the proposed SEM and CAM.

Figure 3 depicts the code snippet of the implementation of SEM in PyTorch [49]. We obtain the improved localization map of category  $y$ , only using a few lines of codes. Line 9-10 are for the first stage and they obtain the localization map of class  $y$  by selecting the  $y_{th}$  feature map and normalizing the selected map. Line 13-26 are for the second stage. We first extract the feature vectors of the seeds, i.e.,  $F_{seeds}$ , after getting the indices of pixels with top  $K$  scores. Then, we obtain the SEM maps by computing

---

```

1 def SEM(M, F, y, K):
2     #M is top-layer feature maps of size (Y, W, H)
3     #F is high-level feature maps of size (C, W, H)
4     #K is the seed number
5     #y is the category label
6
7     c, w, h = F.shape
8     #Stage one
9     cam = M[y, :, :]
10    cam = (cam - cam.min()) / (cam.max() - cam.min())
11
12    #Stage two
13    _, topk_indices = torch.topk(
14        cam.view(-1),
15        K,
16        largest=True)
17
18    F_seeds = F.view(c, -1)[:, topk_indices]
19    simi = torch.nn.functional.cosine_similarity(
20        F.view(c, -1, 1),
21        F_seeds.view(c, 1, -1),
22        dim=0)
23
24    sem, _ = simi.max(dim=1)
25    sem = sem.view(w, h)
26    sem = (sem - sem.min()) / (sem.max() - sem.min())
27
28    return sem

```

---

Fig. 3. Implementation of SEM in PyTorch [49].

the cosine similarity between the seed features  $F_{seeds}$  and the rest features in  $F$ . In our implementation, the height  $H$  and width  $W$  of the feature maps  $M$  and  $F$  are one-eighth of the input images and the number  $K$  of seeds is experimentally set as small values ( $< 150$ ), which makes the computation of cosine similarity quite efficient.

### 3.3 Beyond Localization Map

Previous works [3]–[5], [7] only aim at roughly locating the target objects rather than acquiring their fine details, i.e., object boundaries. Here, we take a step further to explore if we can obtain much finer details of objects under the supervision of only image-level labels. Different from fully supervised edge detection approaches [42], [44] that require a large number of human-annotated edges for supervision, such detailed supervision information can hardly be obtained on such huge dataset (1.2M images). Khoreva *et al.* [45] proposes a weakly supervised method for detecting object boundaries. However, this method requires bounding boxes of objects rather than only applying image-level labels. It also uses intensive computational resources to infer the object boundaries using GrabCut [50], MCG [51] and object detector [52], which prevents it from applying to a large-scale dataset. To the best of our knowledge, our method is the first feasible attempt to conduct object boundary detection on such a large-scale dataset in a weakly supervised manner.

The framework of our SEM-based boundary detector is shown in Figure 4. Localization maps act as a vital bridge for learning boundaries. We fix the network parameters of producing localization maps, and borrow the multi-scale features to predict boundaries. Therefore, the quality of predicted boundaries can also reflect the fineness of discovering object details in localization maps. Given an input image  $I$ , we generate  $M_{sem}$  with the description in Section 3.2. The maps can highlight the area of the target objects but fail to grasp the object boundaries. To solve this issue, we first employ the map  $M_{sem}$  as the unary potentials of ConvCRF [53] to locate coarse object boundaries together with

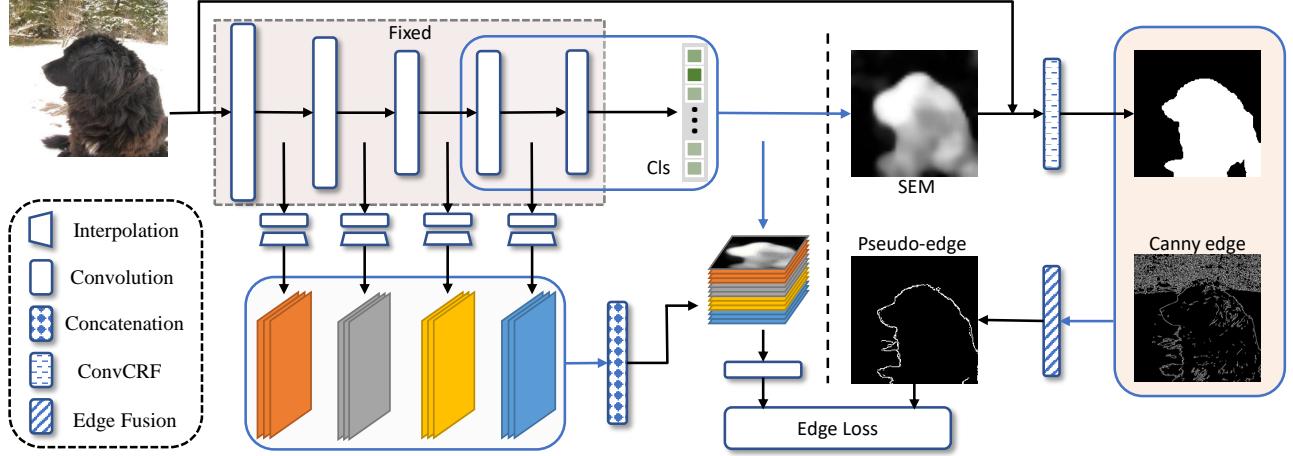


Fig. 4. The network structure for training object boundaries. Pseudo-edges are generated based on the localization maps produced by SEM, and then employed as the supervision to train the boundaries with multiscale feature maps.

the input RGB image  $I$ . Then, the located boundaries are fused with the Canny [41] edges by finding the longest contour to generate pseudo-boundaries. Finally, we use the pseudo-boundaries as supervision to train the boundary detector. The boundary detector shares the backbone network with SEM and utilizes the multi-scale feature maps as the input. Multi-scale feature maps are firstly fed into a convolution layer to adapt the channels, and then enlarged to predict boundaries.

Suppose  $B$  denotes the ground-truth boundary mask,  $B_{i,j}$  is 1 if a pixel at  $(i, j)$  is on boundaries, otherwise  $B_{i,j}$  is 0. The typical loss function [42], [43] to optimize object boundaries is a cross-entropy function. Due to the imbalance in the number between the boundary pixels and the background pixels, we apply the weights  $\alpha = \frac{|B^+|}{|B^+| + |B^-|}$  and  $\beta = \frac{|B^-|}{|B^+| + |B^-|}$  to balance the costs in the training loss, where  $B^+$  and  $B^-$  denote the positive and negative boundary pixels, respectively. Therefore, the vanilla loss is as the second part in Eq. (1).

$$\begin{aligned} \mathcal{L}^e(I) &= \frac{1}{|B^+| + |B^-|} \sum_{i,j} [\underbrace{\alpha(1 - B_{i,j})P_{i,j} \log(1 - P_{i,j})}_{HNS} \\ &\quad + \underbrace{\beta B_{i,j} \log P_{i,j} + \lambda \alpha(1 - B_{i,j}) \log(1 - P_{i,j})}_{vanilla}], \end{aligned} \quad (1)$$

where  $\lambda$  is a hyper-parameter.  $P_{i,j}$  is the predicted probability of being edge point at the pixel  $(i, j)$ .

The vanilla loss function can successfully recall edge pixels. However, it fails in depressing false positive points. The reason is that  $\alpha$  is usually a very small value because the number of positive points only accounts for a very small portion of images. If a false positive point occurs, the cost of this point will be negligible compared to the total cost. Enlarging the value of  $\lambda$  could be a feasible way to address this problem, but it could harm the ability to recovering edge pixels. To tackle this problem, we propose to add a Hard-Negative Suppression (HNS) item to eliminate the edges in background regions and only preserve the edges around target objects. The proposed HNS item is shown in Eq.(1). If the point  $(i, j)$  mismatches with the pseudo-edge point, the score  $P_{i,j}$  is expected to be 0. For the hard-negative points with high scores  $P_{i,j}$ , they will cause high contributions to the total cost. The higher hard-negative scores will induce higher costs, and therefore, their scores will be gradually suppressed with the training process.

## 4 EXPERIMENTS

### 4.1 Experiment Setup

#### 4.1.1 Datasets

We mainly perform the experiments on two popular benchmarks, *i.e.*, ILSVRC [1] and CUB-200-2011 [2], following the previous state-of-the-art methods [3]–[5]. ILSVRC is a widely acknowledged localization dataset including 1.2 million images of 1,000 categories for training and 50,000 images for validation. CUB-200-2011 includes 11,788 images (5,994 for training and 5,794 for testing) of 200 different species of birds. In our experiments, the proposed method is learned on the training sets using only image-level labels as supervision.

#### 4.1.2 Evaluation Metrics

**Direct Evaluation** We mainly utilize the proposed *direct* metric to evaluate the localization performance by performing a pixel-wise comparison between the produced localization map and its corresponding ground-truth mask. For ILSVRC, we annotate the ground-truth masks for the images on the validation set. Particularly, we first manually exclude 5,729 images that have ambiguous object pixels and annotate the left 44,271 images with an interactive object segmentation approach [54]. Then, the annotated masks are split for validation (23,151 images) and test (21,120 images), respectively. Please see the supplementary material for more details of the annotated masks. For CUB-200-2011, the ground-truth masks have already been provided in the original dataset. Given the produced localization maps, each pixel is with probability value bounded in  $[0, 1]$ , and the desired object-related pixels are expected to have high activation values. For the convenience of visualization and evaluation, we map the values to the range of  $[0, 255]$ . During evaluation, we binarize the localization maps using a threshold  $T \in [0, 255]$ , and calculate IoU scores of the binarized masks against ground-truth masks *w.r.t.* different thresholds. We mainly compare different algorithms under the IoU-Threshold curve and its two key properties, *i.e.*, Peak-IoU and Peak-T. In addition, we also report the Precision-Recall curve and the Average-Precision (AP) scores.

**Indirect Evaluation** Following previous approaches [3]–[5], we also conduct the evaluation using the *indirect* measurement to illustrate the superiority of the proposed SEM method. To be specific, this metric calculates the percentage of the images that can satisfy the following two conditions simultaneously. 1) The

predicted classification labels match the ground-truth categories; 2) The predicted bounding boxes have over 50% IoU with at least one of the ground-truth boxes. This criterion involves two aspects, *i.e.*, classification accuracy and localization accuracy. We also evaluate the localization maps *w.r.t.* the ground-truth labels, which is dubbed as Gt-known accuracy.

#### 4.1.3 Implementation Details

To understand the true localization ability of localization maps produced by the current methods, we first implement several recent benchmark methods, *i.e.*, CAM [3], ACoL [4], SPG [5], ADL [6] and CutMix [10]. We apply the same backbone network, *i.e.*, InceptionV3 [55], and the same training strategy for a fair comparison. We train the networks following the way in the original papers [3]–[7], [10]. During testing, We use their own localization maps as the fist-stage guidance and apply the proposed SEM method to enhance localization maps. If there is no specific mentioning, SEM is applied on the CAM [3] method using InceptionV3 by default in this paper. To fully understand the proposed SEM in terms of different backbone networks, we further implement the proposed SEM based on various popular backbone networks, including VGG16 [56], InceptionV3 [55] and ResNets [57]. We follow the baseline methods, *e.g.*, CAM [3], ACoL [4] and SPG [5], to adapt the backbone classification networks for producing decent localization maps. In particular, we remove the fully connected layer on the top and change the last two  $stride = 2$  to  $stride = 1$  for enlarging the size of feature maps. We add a block of three convolution layers to adjust channel numbers, which follows ACoL [4] to obtain final localization maps. For the prediction of SEM-Edge, we simply borrow the feature maps from the backbone networks. Explicitly, the feature maps from the first to fourth blocks are adjusted to the channel size of 128 by a  $Conv3 \times 3$ -BN-ReLU layer, and then upsampled to the size of (160, 160). We concatenate the multi-scale feature maps and the localization maps produced by SEM, and then feed these features into three convolution layers to predict the final edge maps. We follow the training practices of the previous works [4], [5]. The networks are initialized by the weights trained on ILSVRC. For the training process on ILSVRC, the networks are fine-tuned for five epochs with the initial learning rate of 0.001. The learning rate is decreased by a factor of 10 after the second epoch. The learning rate of the added top three convolutional layers is 10 times of the above configuration.

## 4.2 Comparison with the state-of-the-arts

First, we evaluate and compare the recent baseline methods in the proposed *direct* metric. We adapt our training-free SEM method into these baselines to illustrate the effectiveness. We also conduct various backbone networks to verify the robustness of SEM. Second, we show that the proposed SEM can also boost the accuracy of the inferred bounding boxes in the traditional *indirect* measurement. Our improved localization maps achieve the best accuracy and significantly surpass the existing methods.

**Peak-IoU** Table 1 shows the recent weakly localization approaches under the proposed *direct* evaluation metric. We use “w/” and “w/o” to indicate whether the proposed SEM is applied to enhance the quality of localization maps. The proposed SEM successfully surpasses the counterparts in the proposed direct metric of Peak-IoU and Peak-T. We can observe that Peak-IoU values of CAM are 50.58% and 50.13% on our annotated

validation and test sets, respectively. The other methods, *i.e.*, HaS, ACoL, CutMix, SPG and ADL, which are proposed to improve the quality of localization maps based on CAM, have increased the Peak-IoU value only by 1.01% and 1.77% (ACoL) at most. Not all of these methods have really succeeded in improving localization maps. By contrast, the proposed SEM module achieves consistent improvement over every approach. The Peak-IoU values increase to 57.02% and 56.92% (SPG w/ SEM) on the validation and testing sets, respectively. Our SEM method successfully achieves a new state-of-the-art accuracy, surpassing the best values w/o SEM by 5.43% and 5.02% on the two splits. Compared to the method w/o SEM, SEM achieves the most significant improvement on SPG [5], increasing Peak-IoU by 5.74% and 5.16% on the two splits, respectively. ACoL [4] and ADL [6], which apply erasing operations on high-level feature maps to force the networks to find more object regions, obtain the least increase. The cause for the limited improvements of these two methods is that the erasing operation on feature maps may include background noises. We show the included noises in Figure 8.

We also validate the robustness of the proposed SEM method on various backbone networks. In Table 2, we choose four different popular backbone networks, and compare our SEM model with the CAM method. SEM obtains the smallest improvement on VGG16 of increasing the Peak-IoU values by 2.19% and 2.64%, while achieves the largest improvement on ResNet50 by 6.08% and 6.03% on the validation and test splits, respectively. Also, we validate the SEM method on the CUB-200 dataset upon different backbone networks. All of the Peak-IoU results surpass the CAM counterparts. ResNet50 also achieves the largest gain of rising by 7.14% from 47.02% to 54.16%.

We draw the IoU-Threshold curves to give an even clear comparison of various baseline methods in Figure 5a and 5b. It is easy to read from the curves that 1) the baseline methods using CAM to extract localization maps have lower Peak-IoU values, while the Peak-IoU values obviously increase after applying the proposed SEM module; 2) the Peak-T values drift towards bigger values after applying SEM, reflecting that the localization maps have better visual effects. Figure 6a and 6b illustrate the IoU-Threshold curves regarding various backbone networks. We can also observe that the SEM method can not only improve Peak-IoU values but promote the Peak-T to produce better localization maps for visualization.

**Precision-Recall** Beyond the proposed IoU-Threshold metric, we also evaluate the predicted localization maps against the ground-truth masks using Precision-Recall curves and Average Precision (AP) values. In Table 1, we first see the typical CAM method achieves AP of 70.68% and 71.06% on the annotated validation and test sets, respectively. It is noticeable that only HaS and SPG obtain consistent improvement on the two splits, while other methods, *i.e.*, ACoL, CutMix and ADL, do not have better results than CAM. These methods have claimed to produce better localization maps, and it turns out that not all of them have succeeded in this direct measurement. We further apply our SEM model to enhance the extracted localization maps, and it successfully increases the AP scores of four baseline methods. We notice that SEM decreases the AP scores on ACoL and ADL, which is also caused by high-level erasing operation as we have mentioned. In Table 2, we compare the usefulness of SEM on different backbone networks regarding the AP scores. We find that SEM can consistently make obvious improvements using every backbone network. InceptionV3 [55] achieves the best

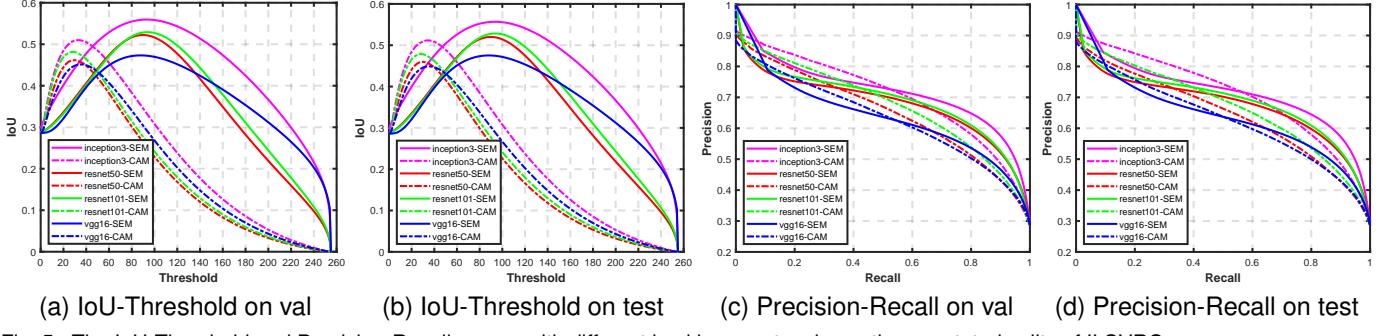


Fig. 5. The IoU-Threshold and Precision-Recall curves with different backbone networks on the annotated splits of ILSVRC.

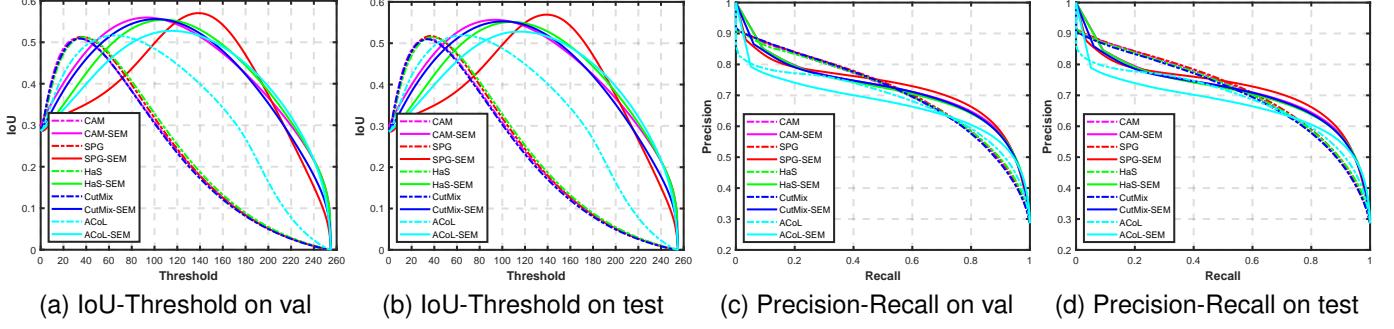


Fig. 6. The IoU-Threshold and Precision-Recall curves of different baseline methods on the annotated splits of ILSVRC.

TABLE 1

Comparison of different methods for extracting localization maps under the amended *direct* evaluation metric on ILSVRC. w/o and w/ indicate whether the proposed SEM is applied for enhancement. Better results are in **bold**.

Method	SEM	AP		Peak-IoU		Peak-T	
		val	test	val	test	val	test
CAM [3]	w/o	70.68	71.06	50.58	50.13	33	38
	w/	<b>72.78</b> (+2.10)	<b>72.54</b> (+1.48)	<b>56.05</b> (+5.05)	<b>55.67</b> (+5.54)	<b>93</b>	<b>93</b>
HaS [7]	w/o	70.79	71.07	51.20	51.44	37	38
	w/	<b>72.46</b> (+1.67)	<b>72.24</b> (+1.17)	<b>55.43</b> (+4.23)	<b>55.15</b> (+3.71)	<b>109</b>	<b>108</b>
ACoL [4]	w/o	<b>68.15</b>	<b>68.64</b>	51.59	51.90	62	65
	w/	67.89 (-0.26)	67.92 (-0.72)	<b>52.74</b> (+0.84)	<b>52.80</b> (+0.90)	<b>115</b>	<b>115</b>
CutMix [10]	w/o	70.78	70.45	50.89	50.97	34	34
	w/	<b>72.12</b> (+1.34)	<b>71.78</b> (+1.33)	<b>55.59</b> (+4.70)	<b>55.28</b> (+4.31)	<b>101</b>	<b>101</b>
SPG [5]	w/o	71.18	71.53	51.28	51.76	36	37
	w/	<b>72.77</b> (+1.59)	<b>72.59</b> (+1.06)	<b>57.02</b> (+5.74)	<b>56.92</b> (+5.16)	<b>139</b>	<b>139</b>
ADL [6]	w/o	<b>69.89</b>	<b>70.19</b>	50.01	49.96	42	42
	w/	67.32 (-2.57)	67.26 (-2.93)	<b>51.10</b> (+1.09)	<b>50.98</b> (+1.02)	<b>144</b>	<b>144</b>

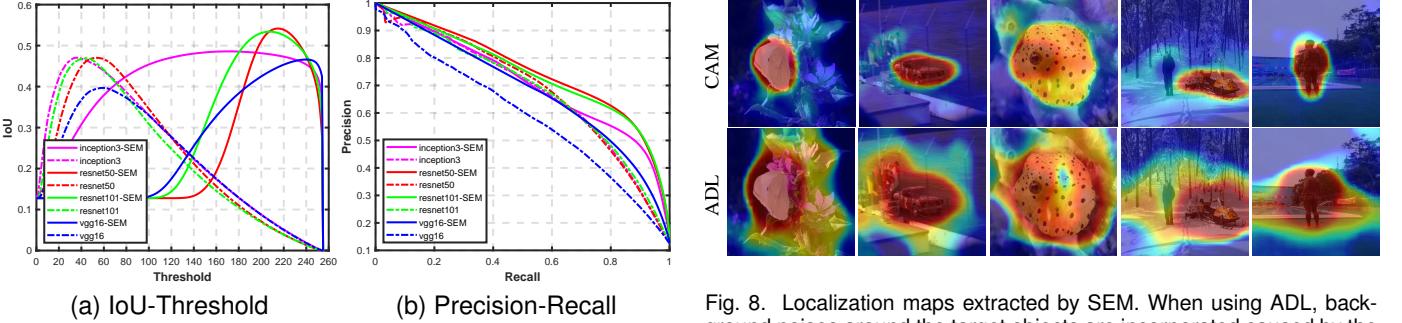


Fig. 7. The IoU and Precision-Recall curves of different backbone networks on the test set of CUB.

scores among the methods without the SEM module, obtaining the AP scores of 70.68% and 71.06% on the validation and test sets, respectively. The proposed SEM method can further increase the

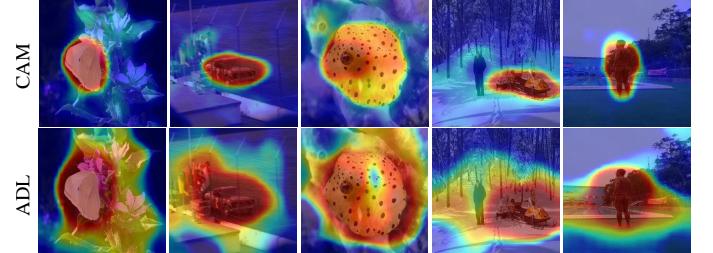


Fig. 8. Localization maps extracted by SEM. When using ADL, background noises around the target objects are incorporated caused by the erasing operation on feature maps.

AP scores by 2.10% and 1.48% and finally obtain the state-of-the-art AP of 72.78% and 72.54% on both splits. SEM can also make consistent improvement on the CUB dataset using various backbone networks, and the improvement of SEM on CUB is even

TABLE 2

Comparison of different backbone networks for extracting localization maps with CAM [3] under the amended *direct* evaluation metric on ILSVRC. w/o and w/ indicate whether the proposed SEM is applied for enhancement. Better results are in **bold**.

Backbone	SEM	AP		Peak-IoU		Peak-T	
		val	test	val	test	val	test
VGG16 [3]	w/o	63.30	62.82	45.13	44.89	35	35
	w/	<b>65.03 (+1.73)</b>	<b>65.14 (+2.32)</b>	<b>47.32 (+2.19)</b>	<b>47.53 (+2.64)</b>	<b>89</b>	<b>88</b>
ResNet50 [7]	w/o	65.18	64.51	46.14	45.98	31	30
	w/	<b>68.12 (+2.94)</b>	<b>67.83 (+1.69)</b>	<b>52.22 (+6.08)</b>	<b>52.01 (+6.03)</b>	<b>90</b>	<b>90</b>
ResNet101 [4]	w/o	67.56	66.99	48.16	47.91	28	28
	w/	<b>69.08 (+1.52)</b>	<b>68.87 (+1.88)</b>	<b>52.91 (+4.75)</b>	<b>52.84 (+4.93)</b>	<b>94</b>	<b>94</b>
Inception3 [10]	w/o	70.68	71.06	51.00	51.17	33	38
	w/	<b>72.78 (+2.10)</b>	<b>72.54 (+1.48)</b>	<b>56.05 (+5.05)</b>	<b>55.67 (+4.50)</b>	<b>93</b>	<b>93</b>

TABLE 3

Comparison with different backbone networks on CUB. w/o of SEM refers to using CAM [3] for extracting localization maps, otherwise using the proposed SEM.

Backbone	SEM	AP	Peak-IoU	Peak-T
VGG16 [56]	w/o	59.17	39.69	49
	w/	<b>76.29(+17.12)</b>	<b>46.60(+6.91)</b>	<b>240</b>
ResNet50 [57]	w/o	68.90	47.02	53
	w/	<b>77.46(+8.56)</b>	<b>54.16(+7.14)</b>	<b>215</b>
ResNet101 [57]	w/o	68.42	46.91	46
	w/	<b>77.45(+9.03)</b>	<b>53.40(+6.49)</b>	<b>208</b>
InceptionV3 [55]	w/o	68.11	47.06	37
	w/	<b>75.63(+7.52)</b>	<b>48.59(+1.53)</b>	<b>171</b>

TABLE 4

Localization accuracy of the inferred bounding boxes on the ILSVRC validation set. †: The results of our re-implemented model.

Backbone	Method	Loc. Acc.		Cls. Acc.
		Top-1	Gt-known	
VGG16 [56]	†CAM [3]	45.15	60.06	71.2
		42.80 [3]	-	68.8 [3]
	ADL [6]	44.92	-	67.8
	ACoL [4]	45.83	62.96	67.5
	DFM [17]	47.41	-	68.6
ResNet50 [57]	SEM CAM	<b>47.53</b>	<b>63.47</b>	71.2
	†CAM [3]	51.13	62.71	75.1
	DFM [17]	49.71	-	77.8
ResNet101 [57]	SEM CAM	<b>53.84</b>	<b>67.00</b>	75.1
	†CAM [3]	52.94	64.13	79.1
	DFM [17]	50.67	-	77.2
	SEM CAM	<b>54.88</b>	<b>67.15</b>	79.1
InceptionV3 [55]	†CAM [3]	50.20	65.05	73.3
	SPG [5]	48.60	64.69	69.7
	ADL [6]	48.71	-	72.8
	†HaS	50.08	65.66	71.7
	†CutMix	47.99	64.46	69.8
	SEM HaS	51.59	68.38	71.7
	SEM CutMix	50.79	68.98	69.8
	SEM SPG	50.79	<b>69.26</b>	69.7
	SEM CAM	<b>53.04</b>	69.04	73.3

more significant. In Table 3, the most dramatic increase is made on VGG16, where the AP scores rise by 17.12% from 59.17% to 76.29% via merely using SEM. SEM can finally boost the AP scores to 77.46% after rising by 8.56% when using ResNet50 as the backbone network.

Figure 5c and 5d compare the Precision-Recall curves between SEM and CAM for extracting localization maps based on recent methods. We observe that localization maps extracted by SEM generally have higher precision when the recall rate is relatively large ( $> 0.6$ ). The explanation for this is that the higher recall rates are corresponding to the higher thresholds for

TABLE 5

Top-1 localization accuracy of the inferred bounding boxes on CUB based on InceptionV3 [55].

CAM [3]	SPG [5]	ADL [6]	SEM
43.67	46.64	53.04	<b>61.57</b>

splitting foreground and background during testing. Therefore, localization maps extracted by SEM have higher precision of foreground pixels over the higher brightness regions, which is highly consistent with the observation from the proposed IoU-Threshold curves and the visualizations of localization maps. We can also observe such situations in Figure 6c and 5d, where various backbone networks are implemented for testing robustness of SEM in different circumstances. In Figure 7b, such situations are even more obvious on various backbone settings when testing on the CUB dataset. However, the proposed IoU-Threshold curves are more convenient and easier to observe different properties of localization maps.

**Localization Accuracy** Following the common practice of previous approaches, we also evaluate the proposed SEM using the *indirect* evaluation metric, *i.e.*, calculating the percentage of post-inferred bounding boxes that have larger than 50% IoU with at least one of the ground-truth boxes. Table 4 compares SEM with various baseline methods. We implement SEM on different backbone networks and various recent WSOL methods, *i.e.*, HaS [7], CutMix [10], CAM [3] and SPG [5]. It is clear that our SEM overtakes all of the recent baseline algorithms on different backbone networks under both Top-1 and Gt-known criteria. We also report the classification error as a reference. Particularly, ResNet101-SEM achieves the best Top-1 localization accuracy of 54.88%, surpassing the other methods and achieving a new state-of-the-art accuracy. For the Gt-known localization accuracy, SEM SPG based on InceptionV3 achieves the best of 69.26% among all the methods, surpassing the current best-performing method, *i.e.*, HaS, by 3.6%. It is also notable that localization accuracy of bounding boxes increases on all the four baselines via simply using SEM to extract localization maps. The largest increase of Top-1 localization accuracy is acquired by using SEM on CAM, raising by 2.84% from 50.20% to 53.04%. The corresponding Gt-known localization accuracy also significantly increases by 3.99% from 65.05% to 69.04%. We also compare the localization accuracy on CUB shown in Table 5 with InceptionV3 as the backbone network. The proposed SEM method surpasses the ADL [6] method by a large margin of 8.53%, obtaining the Top-1 localization accuracy of 61.57% on CUB.

**Visualization** Figure 9 compares the localization maps and the

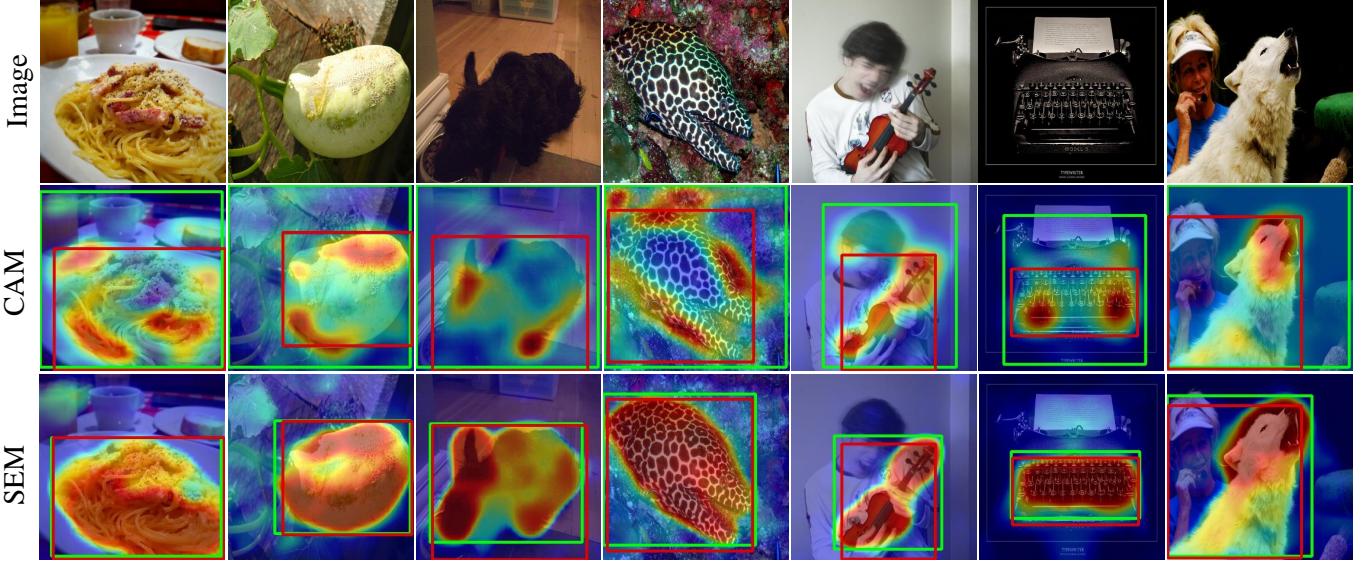


Fig. 9. Visualization of the localization maps and the predicted bounding boxes with SEM and CAM [3]. The ground-truth boxes are in red, and the predicted boxes are in green. The object regions are more intact and have higher brightness in the maps produced by SEM.

post-inferred bounding boxes of various objects on ILSVRC. We can easily and clearly observe that our SEM overtakes its counterpart, *i.e.*, CAM, in both visualization effects and predicted bounding boxes. As we have found in some previous literature [4], [5], [7], CAM is only able to highlight some discriminative and sparse regions of the target objects. By contrast, localization maps produced by SEM can accurately highlight entire target regions with high confidence. We show more examples of localization maps on CUB in the supplementary material where the proposed SEM method consistently outperforms the typical way of extracting localization maps, *i.e.*, CAM. The localization maps from SEM can generally depict the border information of target objects, which paves the way for us to explore more challenging applications, *i.e.*, predicting object boundary, in a weakly supervised manner. As we have discussed in Section 3.1, the IoU-Threshold curves and Peak-T values can describe the visualization effects of localization maps in a detailed and quantitative way. In all the IoU-Threshold curves, *i.e.*, Figure 5a, 5b, 6a, 6b and 7a, the peak points drift towards the upper-right corners after using SEM for extracting localization maps. The meanings of this drifting are 1) more of the target object regions are discovered; 2) brightness of the object regions in localization maps is increased. The Peak-T values can also be quantitatively compared as shown in Table 1, 2 and 3 where the Peak-T values of SEM can consistently outperform the CAM method.

**Discussion** From the above statistics and illustrations, we realize that the recent algorithms [4]–[7], [10] based on CAM [3] actually have not significantly improved the quality of localization maps. According to the proposed direct pixel-wise measurement, the quality value of localization maps has only achieved marginal improvements both in accuracy and visualization effects. By contrast, SEM shows its superiority for the localization map enhancement. It makes a significant improvement over various baselines and backbone networks in both accuracy and visualization brightness. It achieves the state-of-the-art performance on both ILSVRC and CUB benchmarks. We mainly attribute the success of SEM to 1) the seeds obtained in the first-stage maps reliably lie in the object regions; 2) their features are close to the rest object parts while distant to the background pixels. For a good performance of

TABLE 6  
Comparison of the inferred bounding boxes in Gt-known error rates using different number of object seeds. K is the number of object seeds.

K	1	20	40	60	80	100
Gt-known Acc	66.43	68.03	68.75	<b>69.04</b>	68.92	68.24
Peak-IoU	55.59	55.99	<b>56.05</b>	55.95	55.72	55.41

SEM, we may not like to implement it on ACoL [4] and ADL [6]. They erase object regions and force networks to minimize the costs, which brings background noises to high-level feature maps. The enhanced localization maps could thereby include background regions, and the accuracy of localization maps will decrease.

#### 4.3 Ablation Study for the Number of Seeds

K object pixels are utilized as seeds to calculate the similarity with the rest pixels for producing the similarity maps. We expect the K seeds to lie in the target object regions instead of the background areas. We employ InceptionV3 as the backbone network to study the impact of K. Table 6 compares the Gt-known localization accuracy of the inferred bounding boxes and the proposed Peak-IoU scores with K changing from 1 to 100. The Gt-known accuracy increases to 69.04% when K is 60, which achieves the state-of-the-art record in the weakly supervised localization task. When K is getting either larger or smaller, the localization maps will become worse. The Gt-known performance decreases to 66.43% and 68.24% when K changes to 1 and 100, respectively. The reasons behind this phenomena are that: 1) When K is too small, the seed pixels are not diverse enough to retrieve all the object pixels; 2) When K is too large, the background pixels might be included, especially if the target objects are too small. Besides, we notice that the best Peak-IoU is achieved when K is chosen as 40. The inconsistency between Peak-IoU and Gt-known accuracy demonstrates that better localization maps cannot guarantee better performance under the *indirect* evaluation metric, which can further prove the necessity of introducing the *direct* evaluation metric.

#### 4.4 Object Boundary Evaluation

Based on the localization maps of SEM, we further explore to predict object boundaries in a weakly supervised manner on a

TABLE 7

Comparison of the predicted object boundaries. SOBD [46] is learned and tested on subsets of ILSVRC in fully supervised manner. The SEM based methods are obtained with only image-level labels as supervision.

Method	ODS	OIS	AP
SOBD-class specific [46]	-	-	28.9
SOBD-subclass specific [46]	-	-	29.6
SOBD-class agnostic [46]	-	-	29.5
CAM-HNS	34.2	33.7	25.5
SEM-Vanilla	34.3	33.9	17.6
SEM-HNS	39.7	37.6	31.9
SEM-HNS CutMix	40.6	38.2	33.1
SEM-HNS HaS	<b>40.8</b>	<b>38.3</b>	<b>33.3</b>

very large-scale dataset. To examine the quality of the inferred object boundaries, we follow the common practice of fully supervised edge detection approaches [44] to conduct the evaluation. Particularly, we apply a standard non-maximal suppression (NMS) technique to the predicted edge maps to obtain thinned edges, and then report the performance with the most commonly used metrics, *i.e.*, Average Precision (AP), Optimal Dataset Scale (ODS) and Optimal Image Scale (OIS). We infer the ground-truth boundaries from the annotated masks of the validation set and employ the popular edge toolbox in [58], [59] for evaluation. The maximum tolerance allowed for correct matches between edge predictions, and the ground-truth edges is set to 0.0075.

We choose InceptionV3 [55] as the backbone network. In Table 7, we compare different methods in predicting object boundaries on ILSVRC. First, we compare SEM-Vanilla with SEM-HNS where the former one is learned using the vanilla loss function in Eq. (1), and the latter one is trained by adding the proposed HNS item. Surprisingly, the proposed HNS loss function can dramatically increase the AP score by 14.3% from 17.6% to 31.9%. By adopting useful data augmentation techniques in HaS [7] and CutMix [10], the AP scores can be further enhanced to 33.3% and 33.1%, respectively. Next, we verify the influence of localization maps between the proposed SEM and CAM [3]. By replacing localization maps from SEM to CAM during the learning object boundaries stage, the AP score decreases from 31.9% to 25.5%. This result indicates that our proposed SEM can discover more details in the produced localization maps than CAM. As a reference, we try our best and find only one method, *i.e.*, SOBD [46], which aims at predicting object boundaries on such a large scale dataset. Different from our method, SOBD adopts a fully supervised approach to learning multiple situational detectors. It is surprising to see that our weakly-supervised edge detector significantly outperforms SOBD by more than 3.8% in AP. Figure 10 compares the predicted object boundaries between the vanilla loss function and our modified HNS function. Caused by the lack of ground-truth boundaries, our edges include noises within the object regions. After applying the proposed hard negative depression loss function, we can reduce the background edges and weaken some unwanted edges.

## 5 CONCLUSION

In this paper, we make three contributions. We firstly analyze the deficiencies of the current indirect metric and propose to apply a more delicate and direct approach for evaluating localization maps. We annotate images on the ILSVRC validation set to fulfill the direct metric. Then, we propose a two-stage method for obtaining better localization maps, which can accurately cover target objects. The proposed SEM method outperforms all existing

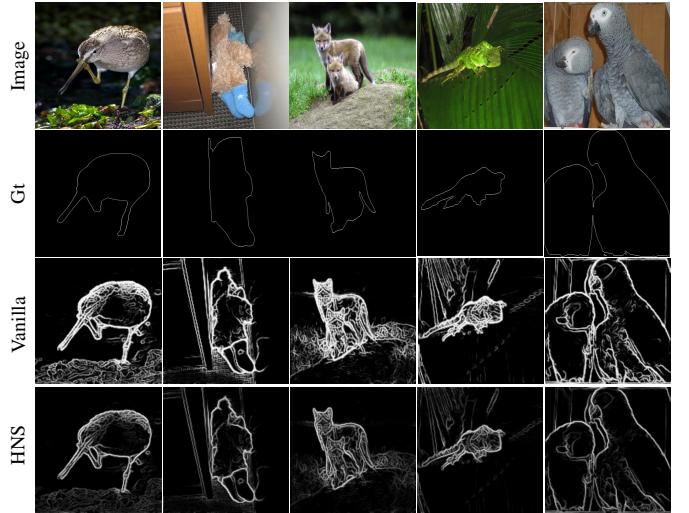


Fig. 10. The predict the object boundaries using only image categories as supervision. Our HNS loss is able to effectively suppress unimportant noises.

methods with various backbone networks. Based on SEM, we further explore the approach for generating quality object boundaries with only image-level labels as supervision. The proposed SEM-HNS method can accurately predict object boundary pixels and also outperform baseline methods in Average Precision. SEM-HNS applies the proposed Hard-Negative-Suppression loss to eliminate undesired edges in the background.

## REFERENCES

- [1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009, pp. 248–255. [1, 4, 6](#)
- [2] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011. [1, 6](#)
- [3] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization.” *CVPR*, 2016. [1, 3, 4, 5, 6, 7, 8, 9, 10, 11](#)
- [4] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. Huang, “Adversarial complementary learning for weakly supervised object localization,” in *IEEE CVPR*, 2018. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10](#)
- [5] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T. Huang, “Self-produced guidance for weakly-supervised object localization,” in *ECCV*. Springer, 2018. [1, 3, 4, 5, 6, 7, 8, 9, 10](#)
- [6] J. Choe and H. Shim, “Attention-based dropout layer for weakly supervised object localization,” in *CVPR*, June 2019. [1, 3, 4, 7, 8, 9, 10](#)
- [7] K. K. Singh and Y. J. Lee, “Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization,” *arXiv preprint arXiv:1704.04232*, 2017. [1, 3, 5, 7, 8, 9, 10, 11](#)
- [8] H. Xue, C. Liu, F. Wan, J. Jiao, X. Ji, and Q. Ye, “Danet: Divergent activation for weakly supervised object localization,” in *ICCV*, October 2019. [1, 3](#)
- [9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *IJCV*, vol. 115, no. 3, pp. 211–252, 2015. [1](#)
- [10] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “Cutmix: Regularization strategy to train strong classifiers with localizable features,” in *ICCV*, 2019, pp. 6023–6032. [1, 3, 4, 5, 7, 8, 9, 10, 11](#)
- [11] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, “Top-down neural attention by excitation backprop,” in *ECCV*. Springer, 2016, pp. 543–559. [3, 4](#)
- [12] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks,” in *WACV*. IEEE, 2018, pp. 839–847. [3](#)
- [13] Y. Zhu, Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, “Soft proposal networks for weakly supervised object localization,” *arXiv preprint arXiv:1709.01829*, 2017. [3](#)

- [14] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, “Weakly supervised instance segmentation using class peak response,” in *CVPR*, 2018, pp. 3791–3800. [3](#)
- [15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *ICCV*, 2017, pp. 618–626. [3](#)
- [16] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach,” in *CVPR*, 2017. [3](#)
- [17] Y. Zhou, Z. Chen, H. Shen, Q. Liu, R. Zhao, and Y. Liang, “Dual-attention focused module for weakly supervised object localization,” *arXiv preprint arXiv:1909.04813*, 2019. [3, 9](#)
- [18] A. Khoreva, R. Benenson, M. Hein, and B. Schiele, “Simple does it: Weakly supervised instance and semantic segmentation,” in *CVPR*, vol. 1, no. 2, 2017, p. 3. [3](#)
- [19] J. Dai, K. He, and J. Sun, “Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation,” *ICCV*, 2015. [3](#)
- [20] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, “Weakly-and semi-supervised learning of a dcnn for semantic image segmentation,” *arXiv preprint arXiv:1502.02734*, 2015. [3](#)
- [21] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, “Scribblesup: Scribble-supervised convolutional networks for semantic segmentation,” *CVPR*, 2016. [3](#)
- [22] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, “Normalized cut loss for weakly-supervised cnn segmentation,” in *CVPR*, 2018. [3](#)
- [23] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov, “On regularized losses for weakly-supervised cnn segmentation,” in *ECCV*, 2018, pp. 507–522. [3](#)
- [24] R. Qian, Y. Wei, H. Shi, J. Li, J. Liu, and T. Huang, “Weakly supervised scene parsing with point-based distance metric learning,” in *AAAI*, vol. 33, 2019, pp. 8843–8850. [3](#)
- [25] W. Shimoda and K. Yanai, “Self-supervised difference detection for weakly-supervised semantic segmentation,” in *ICCV*, October 2019. [3](#)
- [26] Y. Zeng, Y. Zhuge, H. Lu, and L. Zhang, “Joint learning of saliency detection and weakly supervised semantic segmentation,” in *ICCV*, October 2019. [3](#)
- [27] Y. Shen, R. Ji, Y. Wang, Y. Wu, and L. Cao, “Cyclic guidance for weakly supervised joint detection and segmentation,” in *CVPR*, June 2019. [3](#)
- [28] C. Song, Y. Huang, W. Ouyang, and L. Wang, “Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation,” in *CVPR*, June 2019. [3](#)
- [29] R. G. Cinbis, J. Verbeek, and C. Schmid, “Weakly supervised object localization with multi-fold multiple instance learning,” *TPAMI*, vol. 39, no. 1, pp. 189–203, 2016. [3](#)
- [30] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu, “Deep self-taught learning for weakly supervised object localization,” in *CVPR*, 2018. [3](#)
- [31] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, “Weakly supervised object localization with progressive domain adaptation,” in *CVPR*, 2016, pp. 3512–3520. [3](#)
- [32] Y. Shen, R. Ji, S. Zhang, W. Zuo, and Y. Wang, “Generative adversarial learning towards fast weakly supervised detection,” in *CVPR*, 2018, pp. 5764–5773. [3](#)
- [33] H. Bilen and A. Vedaldi, “Weakly supervised deep detection networks,” in *CVPR*, 2016, pp. 2846–2854. [3](#)
- [34] V. Kantorov, M. Oquab, M. Cho, and I. Laptev, “Contextlocnet: Context-aware deep network models for weakly supervised localization,” in *ECCV*. Springer, 2016, pp. 350–365. [3](#)
- [35] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, “Weakly supervised cascaded convolutional networks,” *arXiv preprint*, 2017. [3](#)
- [36] Y. Wei, Z. Shen, B. Cheng, H. Shi, J. Xiong, J. Feng, and T. Huang, “Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection,” in *ECCV*, 2018, pp. 434–450. [3](#)
- [37] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, “Weakly supervised region proposal network and object detection,” in *ECCV*, 2018, pp. 352–368. [3](#)
- [38] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, “Hcp: A flexible cnn framework for multi-label image classification,” *TPAMI*, vol. 38, no. 9, pp. 1901–1907, 2016. [3](#)
- [39] F. Wan, P. Wei, J. Jiao, Z. Han, and Q. Ye, “Min-entropy latent model for weakly supervised object detection,” in *CVPR*, 2018, pp. 1297–1306. [3](#)
- [40] Z. Zeng, B. Liu, J. Fu, H. Chao, and L. Zhang, “Wsdod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection,” in *ICCV*, 2019, pp. 8292–8300. [3](#)
- [41] J. Canny, “A computational approach to edge detection,” *TPAMI*, no. 6, pp. 679–698, 1986. [3, 6](#)
- [42] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *ICCV*, 2015, pp. 1395–1403. [3, 5, 6](#)
- [43] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, “Richer convolutional features for edge detection,” in *CVPR*, 2017, pp. 3000–3009. [3, 6](#)
- [44] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, “Bi-directional cascade network for perceptual edge detection,” in *CVPR*, 2019, pp. 3828–3837. [3, 5, 11](#)
- [45] A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele, “Weakly supervised object boundaries,” in *CVPR*, 2016, pp. 183–192. [3, 5](#)
- [46] J. R. Uijlings and V. Ferrari, “Situational object boundary detection,” in *CVPR*, 2015, pp. 4712–4721. [3, 11](#)
- [47] L. Van Der Maaten, “Accelerating t-sne using tree-based algorithms,” *JMLR*, vol. 15, no. 1, pp. 3221–3245, 2014. [4](#)
- [48] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim, “Evaluating weakly supervised object localization methods right,” in *cvpr*, 2020. [4](#)
- [49] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS-W*, 2017. [5](#)
- [50] C. Rother, V. Kolmogorov, and A. Blake, “grabcut” interactive foreground extraction using iterated graph cuts,” *ACM TOG*, vol. 23, no. 3, pp. 309–314, 2004. [5](#)
- [51] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping for image segmentation and object proposal generation,” *TPAMI*, vol. 39, no. 1, pp. 128–140, 2016. [5](#)
- [52] R. Girshick, “Fast r-cnn,” in *arXiv preprint arXiv:1504.08083*, 2015. [5](#)
- [53] M. T. Teichmann and R. Cipolla, “Convolutional crfs for semantic segmentation,” *arXiv preprint arXiv:1805.04777*, 2018. [6](#)
- [54] S. Zhang, J. H. Liew, Y. Wei, S. Wei, and Y. Zhao, “Interactive object segmentation with inside-outside guidance,” in *CVPR*, 2020. [6](#)
- [55] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *CVPR*, 2016, pp. 2818–2826. [7, 9, 11](#)
- [56] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ICLR*, 2015. [7, 9](#)
- [57] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778. [7, 9](#)
- [58] P. Dollár and C. L. Zitnick, “Structured forests for fast edge detection,” in *ICCV*, 2013. [11](#)
- [59] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *ECCV*, 2014. [11](#)

# Supplementary Material

## APPENDIX A MASK ANNOTATION ON ILSVRC

In order to fulfill the purpose of directly evaluating localization maps with high precision and reliability, we annotate pixel-level masks of objects on the ILSVRC [1] validation dataset which is the most acknowledged dataset for studying localization maps. We first manually exclude 5,729 images that have ambiguous object pixels and annotate the left 44,271 images. We then divide the annotated masks into two groups, *i.e.*, the validation and test group. The validation group contains 23,151 images, while the test group contains 21,120 images. Figure 2 depicts some images with the labeled pixel-level masks. The target objects are chosen according to the image-level labels and bounding boxes provided on the ILSVRC CLS-LOC dataset. If multiple objects appear in a given image, *e.g.*, the man holding a rifle, we only label the object corresponding to the image-level label, *e.g.*, rifle. We choose such an annotation strategy because the annotated masks should be consistent with the category labels for training classification networks. Figure 1 shows the histogram of the area of target objects of all the 44,271 images. It is obvious that objects account for less than half of the area in most images. The scale of Region of Interests (RoIs) changes greatly, as the smallest objects only have about 10 pixels while the largest objects contain about 1M pixels. Also, we notice that there are about 400 images where target objects occupy nearly the entire images.

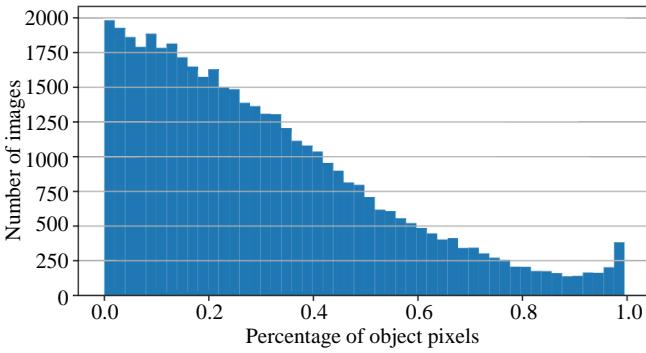


Fig. 1. Histogram of object area rates in images on ILSVRC.

## APPENDIX B SEM WITH DIFFERENT FEATURE MAPS

Feature maps of different depths are commonly acknowledged to have different semantic information. Features from low-level layers tend to involve more edge and texture information, while features from high-level layers are more abstract to express rough object locations. We employ feature maps of different depths to

TABLE 1  
Comparison of the inferred bounding boxes in Gt-known accuracy using different feature maps.  $feat_x$  denote the feature maps of  $x_{th}$  block.  $feat_5$  is extracted from the penultimate convolution layer in the  $5_{th}$  block.

feat3	feat4	feat5	Gt. known Acc	Peak-IoU
✓			42.55	40.98
	✓		58.56	52.41
		✓	<b>69.04</b>	56.05
✓	✓	✓	68.03	<b>56.09</b>

study the changes in localization abilities. In Table 1, we compare the Gt-known localization accuracy and Peak-IoU with feature maps of different depths.  $feat_3$  and  $feat_4$  are the output feature maps of Block3 and Block4.  $feat_5$  is the output feature maps of the penultimate convolutional layer. Please refer to the [released code](#) for more details. Gt-known localization accuracy is the best when using the feature maps of the penultimate convolutional layer (*labeled as feat5 in Table 1*), achieving the highest accuracy of 69.04%. The localization ability is dramatically getting worse when applying feature maps of lower levels. We also study the Gt-known localization accuracy of the combination of  $feat_4$  and  $feat_5$  by concatenation. By such a method of feature map concatenation, the localization accuracy decreases to 68.03%, which is worse than the result of only using  $feat_5$ . In terms of the proposed direct metric, *i.e.*, Peak-IoU, it achieves 56.06% when only using the features from the fifth block. Peak-IoU can be slightly increased to 56.09% when concatenating  $feat_3$  and  $feat_5$ . In our experiments, we only use  $feat_5$  to obtain localization maps using the proposed SEM approach.

## REFERENCES

- [1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009, pp. 248–255. [1](#)
- [2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *IJCV*, vol. 115, no. 3, pp. 211–252, 2015. [2](#)
- [3] L. Van Der Maaten, “Accelerating t-sne using tree-based algorithms,” *JMLR*, vol. 15, no. 1, pp. 3221–3245, 2014. [3](#)
- [4] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T. Huang, “Self-produced guidance for weakly-supervised object localization,” in *ECCV*. Springer, 2018. [3](#)
- [5] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization.” *CVPR*, 2016. [3, 4](#)
- [6] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011. [4](#)



Fig. 2. Illustration of our annotated object masks on the ILSVRC validation set. Numbers indicate the category IDs in ILSVRC [2]

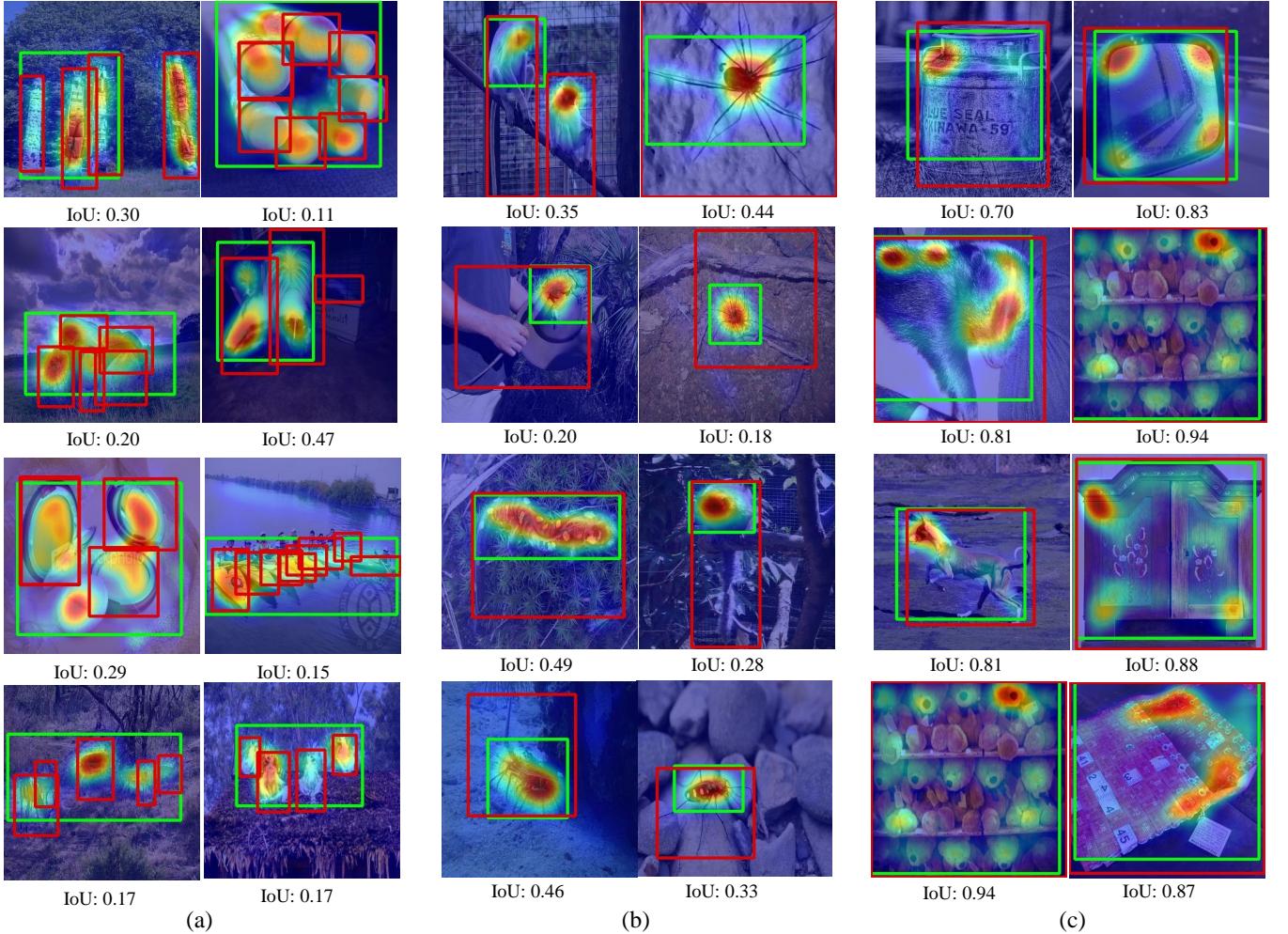


Fig. 3. The current metric treats an image as true positive if the post-inferred bounding boxes have over 50% IoU with at least one of the ground-truth boxes. Here, we show three scenarios where the current indirect evaluation metric fails to measure localization maps: (a) Localization maps accurately highlight target objects, while the metric considers them as false positive; (b) main parts of the target objects are highlighted, while the metric gives them zero credit; (c) although the predicted bounding boxes are accurate, the localization maps fail to highlight important areas of the target objects.

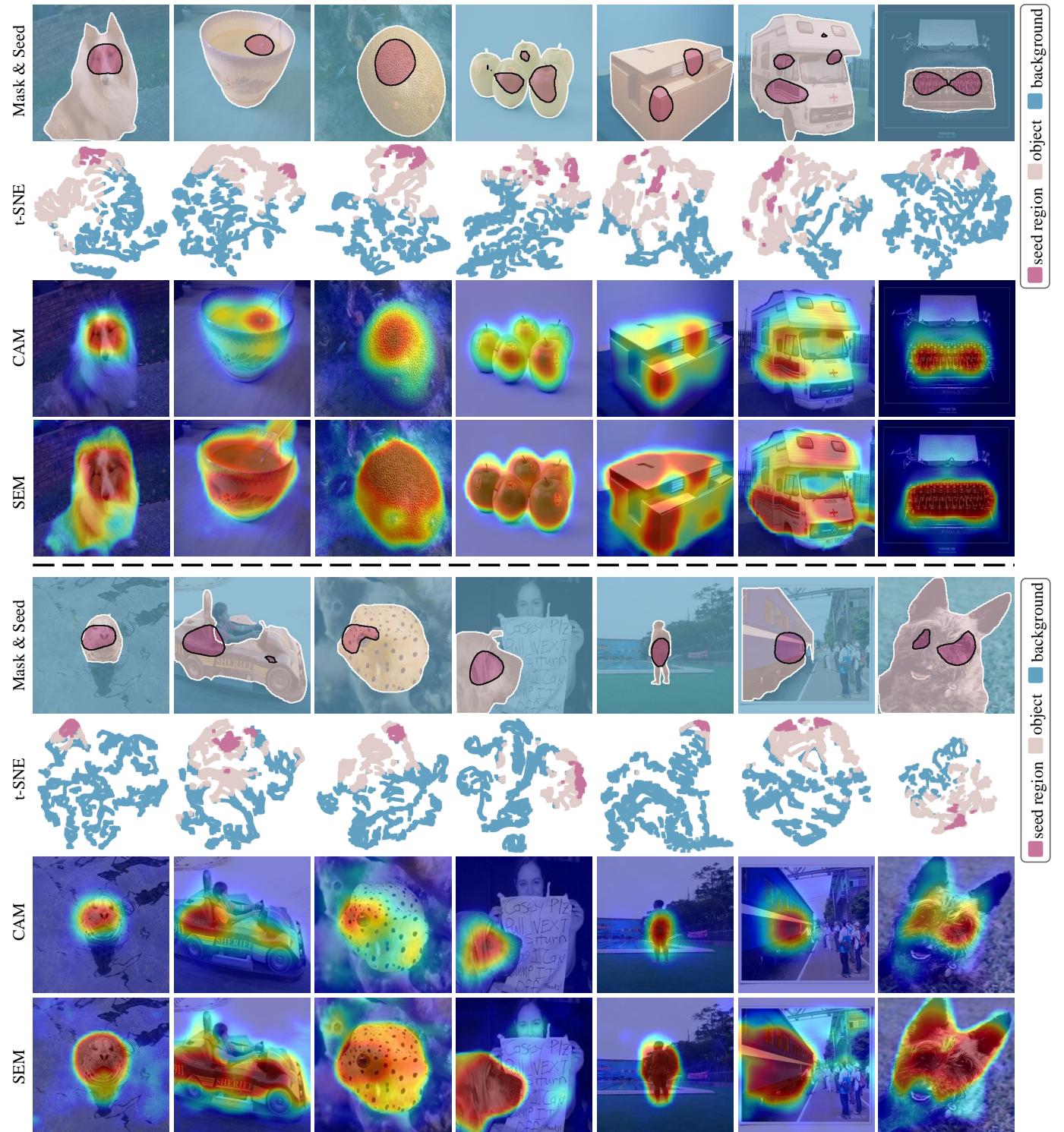


Fig. 4. t-SNE [3] of high-level features. Motivation illustration and visualization comparison of the proposed SEM approach. We illustrate the seed regions using the pixels with top 70% scores chosen in SPG [4]. The top two rows show the high-level features of seed regions are closer to the rest object regions than background areas. We implement similarities between pixels to enhance localization maps. The bottom two rows compare the proposed SEM and CAM [5], and SEM achieves better performance in accurately highlighting target objects.

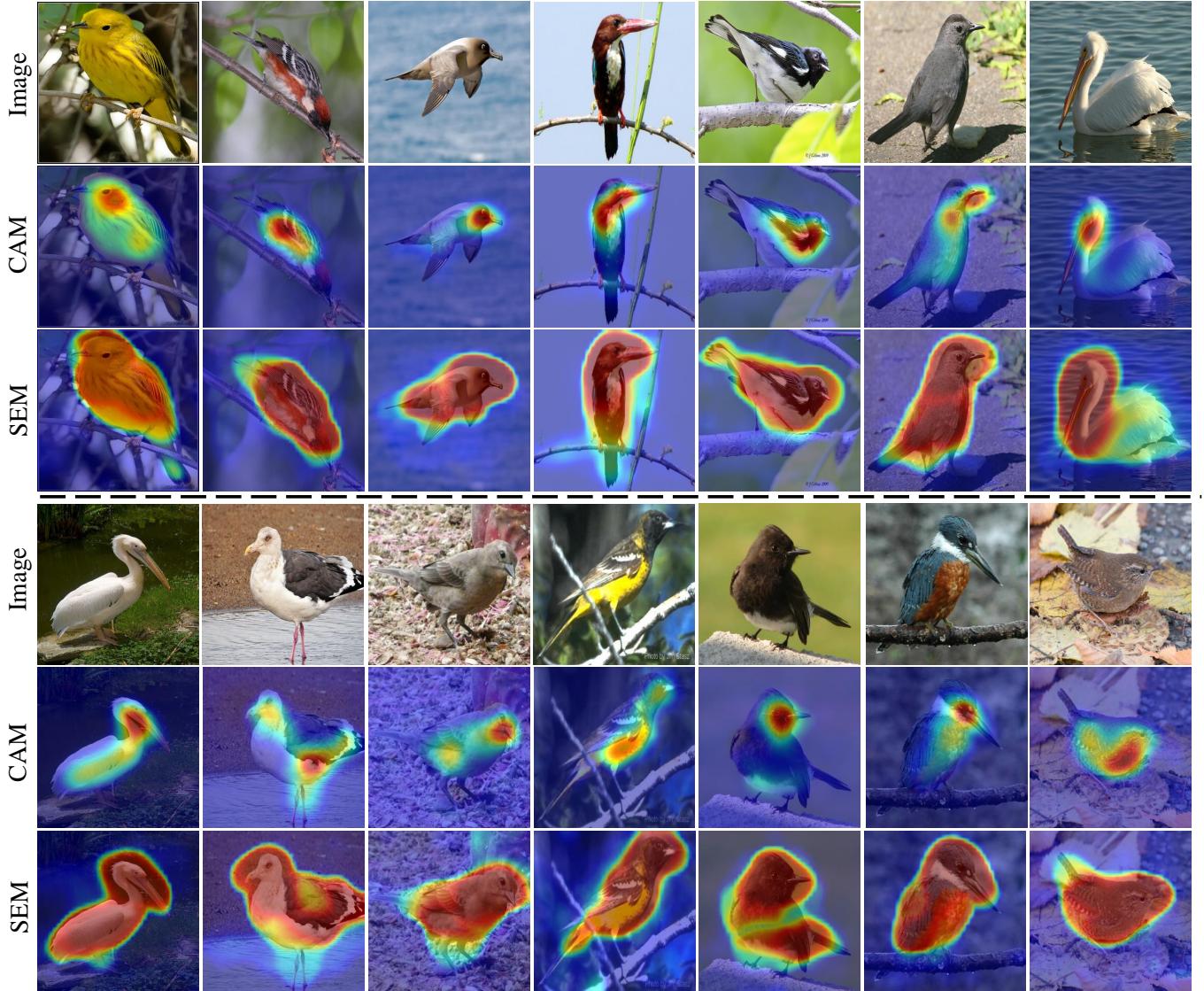


Fig. 5. Visualization of the localization maps produced by CAM [5] and SEM on CUB [6]. CAM only highlights the sparse discriminative regions, while SEM can highlight the entire object regions and present more distinctions between the objects and background.