Evaluating Weakly Supervised Object Localization Methods Right

Junsuk Choe1,3* Sanghyuk Chun3
Seong Joon Oh2* Seungho Lee1
Zeynep Akata4
Hyunjung Shim1†

1 School of Integrated Technology, Yonsei University
2 Clova AI Research, LINE Plus Corp.
3 Clova AI Research, NAVER Corp.
4 University of Tuebingen

Abstract

Weakly-supervised object localization (WSOL) has gained popularity over the last years for its promise to train localization models with only image-level labels. Since the seminal WSOL work of class activation mapping (CAM), the field has focused on how to expand the attention regions to cover objects more broadly and localize them better. However, these strategies rely on full localization supervision to validate hyperparameters and for model selection, which is in principle prohibited under the WSOL setup. In this paper, we argue that WSOL task is ill-posed with only image-level labels, and propose a new evaluation protocol where full supervision is limited to only a small held-out set not overlapping with the test set. We observe that, under our protocol, the five most recent WSOL methods have not made a major improvement over the CAM baseline. Moreover, we report that existing WSOL methods have not reached the few-shot learning baseline, where the full-supervision at validation time is used for model training instead. Based on our findings, we discuss some future directions for WSOL. Source code and dataset are available at https://github.com/clovaai/wsolevaluation.

1. Introduction

As human labeling for every object is too costly and weakly-supervised object localization (WSOL) requires only image-level labels, the WSOL research has gained significant momentum [58, 56, 57, 6, 25, 55] recently.

Among these, class activation mapping (CAM) [58] uses the intermediate classifier activations focusing on the most discriminative parts of the objects to localize the objects of the target class. As the aim in object localization is to cover the full extent of the object, focusing only on the most discriminative parts of the objects is a limitation. WSOL

*Equal contribution. Work done at Clova AI Research.
†Hyunjung Shim is a corresponding author.

Figure 1. WSOL 2016-2019. Recent improvements in WSOL are illusory due to (1) different amount of implicit full supervision through validation and (2) a fixed score-map threshold (usually $\tau = 0.2$) to generate object boxes. Under our evaluation protocol with the same validation set sizes and oracle $\tau$ for each method, CAM is still the best. In fact, our few-shot learning baseline, i.e. using the validation supervision (10 samples/class) at training time, outperforms existing WSOL methods. Results on ImageNet.
tion supervision. Towards a well-posed setup, we propose a new WSOL setting where a small held-out set with full supervision is available to the learners.

Our contributions are as follows. (1) Propose new experimental protocol that uses a fixed amount of full supervision for hyperparameter search and carefully analyze six WSOL methods on three architectures and three datasets. (2) Propose new evaluation metrics as well as data, annotations, and benchmarks for the WSOL task at https://github.com/clovaai/wsolevaluation. (3) Show that WSOL has not progressed significantly since CAM, when the calibration dependency and the different amounts of full supervision are factored out. Moreover, searching hyperparameters on a held-out set consisting of 5 to 10 full localization supervision per class often leads to significantly lower performance compared to the few-shot learning (FSL) baselines that use the full supervision directly for model training. Finally, we suggest a shift of focus in future WSOL research: consideration of learning paradigms utilizing both weak and full supervisions, and other options for resolving the ill-posedness of WSOL (e.g. background-class images).

2. Related Work

By model output. Given an input image, semantic segmentation models generate pixel-wise class predictions [11, 31], object detection models [11, 13] output a set of bounding boxes with class predictions, and instance segmentation models [27, 7, 18] predict a set of disjoint masks with class and instance labels. Object localization [41], on the other hand, assumes that the image contains an object of single class and produces a binary mask or a bounding box around that object coming from the class of interest.

By type of supervision. Since bounding box and mask labels cost significantly more than image-level labels, e.g. categories [2], researchers have considered different types of localization supervision: image-level labels [35], gaze [34], points [2], scribbles [26], boxes [8], or a mixture of multiple types [21]. Our work is concerned with the object localization task with only image-level category labels [33, 58].

By amount of supervision. Learning from a small amount of labeled samples per class is referred to as few-shot learning (FSL) [53]. We recognize the relationship between our new WSOL setup and the FSL paradigm; we consider FSL methods as baselines for future WSOL methods.

WSOL works. Class activation mapping (CAM) [58] turns a fully-convolutional classifier into a score map predictor by considering the activations before the global average pooling layer. Vanilla CAM has been criticized for its focus on the small discriminative part of the object. Researchers have considered dropping regions in inputs at random [25, 55] to diversify the cues used for recognition. Adversarial erasing techniques [56, 6] drop the most discriminative part at the current iteration. Self-produced guidance (SPG) [57] is trained with auxiliary foreground-background masks generated by its own activations. Other than object classification in static images, there exists work on localizing informative video frames for action recognition [37, 28, 54], but they are beyond the scope of our analysis.

Relation to explainability. WSOL methods share similarities with the model explainability [17], specifically the input attribution task: analyzing which pixels have led to the image classification results [16]. There are largely two streams of work on visual input attribution: variants of input gradients [50, 45, 42, 48, 43, 49, 24, 36] and counterfactual reasoning [39, 12, 59, 40, 15, 20]. While they can be viewed as WSOL methods, we have not included them in our studies because they are seldom evaluated in WSOL benchmarks. Analyzing their possibility as WSOL methods is an interesting future study.

Our scope. We study the WSOL task, rather than weakly-supervised detection, segmentation, or instance segmentation. The terminologies tend to be mixed in the earlier works of weakly-supervised learning [44, 14, 9, 47]. Extending our analysis to other weakly-supervised learning tasks is valid and will be a good contribution to the respective communities.

3. Problem Formulation of WSOL

We define and formulate the weakly-supervised object localization (WSOL) task as an image patch classification and show the ill-posedness of the problem. We will discuss possible modifications to resolve the ill-posedness in theory.

3.1. WSOL Task as Multiple Instance Learning

Given an image $X \in \mathbb{R}^{H \times W}$, object localization is the task to identify whether or not the pixel belongs to the object of interest, represented via dense binary mask $T = (T_{i1}, \ldots, T_{iHW})$ where $T_{ij} \in \{0, 1\}$ and $(i, j)$ indicate the pixel indices. When the training set consists of precise image-mask pairs $(X, T)$, we refer to the task as fully-supervised object localization (FSOL). In this paper, we consider the case when only an image-level label $Y \in \{0, 1\}$ for global presence of the object of interest is provided per training image $X$. This task is referred to as the weakly-supervised object localization (WSOL).

One can treat an input image $X$ as a bag of stride-1 sliding window patches of suitable side lengths, $h$ and $w$: Figure 2. WSOL as MIL. WSOL is interpreted as a patch classification task trained with multiple-instance learning (MIL). The score map $s(X)$ is thresholded at $\tau$ to estimate the mask $T$. The current iteration. Self-produced guidance (SPG) [57] is trained with auxiliary foreground-background masks generated by its own activations. Other than object classification in static images, there exists work on localizing informative video frames for action recognition [37, 28, 54], but they are beyond the scope of our analysis.
(X_{11}, \ldots, X_{HW}) with X_{ij} \in \mathbb{R}^{h \times w}$. The object localization task is then the problem of predicting the object presence \( T_{ij} \) at the image patch \( X_{ij} \). The weak supervision imposes the requirement that each training image \( X \), represented as \((X_{11}, \ldots, X_{HW})\), is only collectively labeled with a single label \( Y \in \{0, 1\} \) indicating whether at least one of the patches represents the object. This formulation is an example of the multiple-instance learning (MIL) [22], as observed by many traditional WSOL works [35, 44, 14, 47].

Following the patch classification point of view, we formulate WSOL task as a mapping from patches \( X \) to the binary labels \( T \) (induces dropped). We assume that the patches \( X_i \), image-level labels \( Y_i \), and the pixel-wise labeling \( T \) in our data arise in an i.i.d. fashion from the joint distribution \( p(X, Y, T) \). See Figure 2 for an overview. The aim of WSOL is to produce a scoring function \( s(X) \) such that thresholding it at \( \tau \) closely approximates binary label \( T \). Many existing approaches for WSOL, including CAM [58], use the scoring rules based on the posterior \( s(X) = p(Y|X) \). See Appendix §A.1 for the interpretation of CAM as pixel-wise posterior approximation.

3.2. When is WSOL ill-posed?

We show that if background cues are more strongly associated with the target labels \( T \) than some foreground cues, the localization task cannot be solved, even when we know the exact posterior \( p(Y|X) \) for the image-level label \( Y \). We will make some strong assumptions in favor of the learner, and then show that WSOL still cannot be perfectly solved.

We assume that there exists a finite set of cue labels \( \mathcal{M} \) containing all patch-level concepts in natural images. For example, patches from a duck image are one of \{duck's head, duck's feet, sky, water, \cdots\} (see Figure 3). We further assume that every patch \( X \) is equivalently represented by its cue label \( M(X) \in \mathcal{M} \). Therefore, from now on, we write \( M \) instead of \( X \) in equations and examine the association arising in the joint distribution \( p(M, Y, T) \). We write \( M^{fg}, M^{bg} \in \mathcal{M} \) for foreground and background cues.

We argue that, even with access to the joint distribution \( p(Y, M) \), it may not be possible to make perfect predictions for the patch-wise labels \( T(M) \) (proof in Appendix §A.2).

Lemma 3.1. Assume that the true posterior \( p(Y|M) \) with a continuous pdf is used as the scoring rule \( s(M) = p(Y|M) \). Then, there exists a scalar \( \tau \in \mathbb{R} \) such that \( s(M) \geq \tau \) is identical to \( T \) if and only if the foreground-background posterior ratio \( \frac{p(Y=1|M^{fg})}{p(Y=1|M^{bg})} \geq 1 \) almost surely, conditionally on the event \( \{T(M^{fg}) = 1 \text{ and } T(M^{bg}) = 0\} \).

In other words, if the posterior likelihood for the image-level label \( Y \) given a foreground cue \( M^{fg} \) is less than the posterior likelihood given background \( M^{bg} \) for some foreground and background cues, no WSOL method can make a correct prediction. This pathological scenario is described in Figure 3: Duck’s feet are less seen in duck images than the water background. Such cases are abundant in user-collected data (Appendix Figure 1).

This observation implies a data-centric solution towards well-posed WSOL: we can augment (1) positive samples \((Y = 1)\) with more less-represented foreground cues \((e.g.\, duck\, images\, with\, feet)\) and (2) negative samples \((Y = 0)\) with more target-correlated background cues \((e.g.\, non-duck\, images\, with\, water\, background)\). Such data-centric approaches are promising future directions for WSOL.

How have WSOL methods addressed the ill-posedness?

Previous solutions to the WSOL problem have sought architectural modifications [56, 57, 6] and data augmentation [25, 55] schemes that typically require heavy hyperparameter search and model selection, which are a form of implicit localization supervision. For example, [25] has found the operating threshold \( \tau \) via “observing a few qualitative results”, while others have evaluated their models over the test set to select reasonable hyperparameter values (Table 1 of [25], Table 6 of [56], and Table 1 of [6]). [57] has performed a “grid search” over possible values. We argue that certain level of localization labels are inevitable for WSOL. In the next section, we propose to allow a fixed number of fully labeled samples for hyperparameter search and model selection for a more realistic evaluation.

4. Evaluation Protocol for WSOL

We reformulate the WSOL evaluation based on our observation of the ill-posedness. We define performance metrics, benchmarks, and the hyperparameter search procedure.

4.1. Evaluation metrics

The aim of WSOL is to produce score maps, where their pixel value \( s_{ij} \) is higher on foreground \( T_{ij} = 1 \) and lower on background \( T_{ij} = 0 \) (§3.1). We discuss how to quantify the above conditions and how prior evaluation metrics have failed to clearly measure the relevant performance. We then propose the MaxBoxAcc and P75P metrics for bounding box and mask ground truths, respectively.

The localization accuracy [41] metric entangles classification and localization performances by counting the num-
number of images where both tasks are performed correctly. We advocate the measurement of localization performance alone, as the goal of WSOL is to localize objects (§3.1) and not to classify images correctly. To this end, we only consider the score maps \(s_{ij}\) corresponding to the ground-truth classes in our analysis. Metrics based on such are commonly referred to as the \textit{GT-known} metrics [25, 56, 57, 6].

A common practice in WSOL is to normalize the score maps per image because the score statistics differ vastly across images. Either max normalization (divide through by \(\max_{ij} s_{ij}\)) or min-max normalization (additionally map \(\min_{ij} s_{ij}\) to zero) has been used; see Appendix §B.1 for the full summary. We always use the min-max normalization.

After normalization, WSOL methods threshold the score map at \(\tau\) to generate a tight box around the binary mask \(\{(i, j) \mid s_{ij} \geq \tau\}\). WSOL metrics then measure the quality of the boxes. \(\tau\) is typically treated as a fixed value [58, 56, 55] or a hyperparameter to be tuned [25, 57, 6]. We argue that the former is misleading because the ideal threshold \(\tau\) depends heavily on the data and model architecture and fixing its value may be disadvantageous for certain methods.

To fix the issue, we propose new evaluation metrics that are independent of the threshold \(\tau\).

\textbf{Masks: PxAP.} When masks are available for evaluation, we measure the pixel-wise precision and recall [1]. Unlike single-number measures like mask-wise IoU, those metrics allow users to choose the preferred operating threshold \(\tau\) that provides the best precision-recall trade-off for their downstream applications. We define the \textbf{pixel precision and recall at threshold} \(\tau\) as:

\[
\text{PxPrec}(\tau) = \frac{|\{s_{ij}^{(n)} \geq \tau\} \cap \{T_{ij}^{(n)} = 1\}|}{|\{s_{ij}^{(n)} \geq \tau\}|} \quad (1)
\]

\[
\text{PxRec}(\tau) = \frac{|\{s_{ij}^{(n)} \geq \tau\} \cap \{T_{ij}^{(n)} = 1\}|}{|\{T_{ij}^{(n)} = 1\}|} \quad (2)
\]

For threshold independence, we define and use the \textbf{pixel average precision, PxAP}:

\[
\text{PxAP} := \sum_{\tau} \text{PxPrec}(\tau) \cdot (\text{PxRec}(\tau) - \text{PxRec}(\tau-1)),
\]

where \(\tau\) is the area under curve of the pixel precision-recall curve. We use PxAP as the final metric in this paper.

\textbf{Bounding boxes: MaxBoxAcc.} Pixel-wise masks are expensive to collect; many datasets only provide box annotations. Since it is not possible to measure exact pixel-wise precision and recall with bounding boxes, we suggest a surrogate in this case. Given the ground truth box \(B\), we define the \textbf{box accuracy at score map threshold} \(\tau\) and \textbf{IoU threshold} \(\delta\), \(\text{MaxBoxAcc} (\tau, \delta)\) [58, 41], as:

\[
\text{BoxAcc}(\tau, \delta) = \frac{1}{N} \sum_{n} \text{1}_{\text{IoU}(\text{box}(s(X^{(n)}), \tau), B^{(n)}) \geq \delta}
\]

where \(\text{box}(s(X^{(n)}), \tau)\) is the tightest box around the largest-area connected component of the mask \(\{(i, j) \mid s(X^{(n)}_{ij}) \geq \tau\}\). In datasets where more than one bounding box are provided (e.g. ImageNet), we count the number of images where the box prediction overlaps with at least one of the ground truth boxes with \(\text{IoU} \geq \delta\). When \(\delta = 0.5\), the metric is identical to the commonly-called \textit{GT-known localization accuracy} [25] or \textit{CorLoc} [10], but we suggest a new naming to more precisely represent what is being measured. For score map threshold independence, we report the box accuracy at the optimal threshold \(\tau\), the \textbf{maximal box accuracy} \(\text{MaxBoxAcc}(\delta) := \max_{\tau} \text{BoxAcc}(\tau, \delta)\), as the final performance metric. In this paper, we set \(\delta = 0.5\), following the prior works [58, 25, 56, 57, 6, 55].

\textbf{Better box evaluation: MaxBoxAccV2.} After the acceptance at CVPR 2020, we have developed an improved version of \text{MaxBoxAcc}. It is better in two aspects. (1) \text{MaxBoxAcc} measures the performance at a fixed IoU threshold \((\delta = 0.5)\), only considering a specific level of fineness of localization outputs. We suggest averaging the performance across \(\delta \in \{0.3, 0.5, 0.7\}\) to address diverse demands for localization fineness. (2) \text{MaxBoxAcc} takes the \textbf{largest} connected component for estimating the box, assuming that the object of interest is usually large. We remove this assumption by considering the best match between the set of all estimated boxes and the set of all ground truth boxes. We call this new metric as \text{MaxBoxAccV2}.

For future WSOL researches, we encourage using the \text{MaxBoxAccV2} metric. The code is already available in our repository. We show the evaluation results under the new metric in the in Appendix §C.7.

\subsection*{4.2. Data splits and hyperparameter search}

For a fair comparison of the WSOL methods, we fix the amount of full supervision for hyperparameter search. As shown in Table 1 we propose three disjoint splits for every dataset: \textit{train-weaksup, train-fullsup,} and \textit{test}. The \textit{train-weaksup} contains images with weak supervision (the image-level labels). The \textit{train-fullsup} contains images with full supervision (either bounding box or binary mask). It is left as freedom for the user to utilize it for hyperparameter search, model selection, ablative studies, or even model fitting. The \textit{test} split contains images with full supervision; it must be used only for the final performance report. For example, checking the \textit{test} results multiple times with different model configurations violates the protocol as the learner implicitly uses more full supervision than allowed.

As WSOL benchmark datasets, ImageNet [41] and

\begin{table}[h]
\centering
\begin{tabular}{lrrr}
\hline
Dataset & ImageNet & CUB & OpenImages \\
\hline
#Classes & 1000 & 200 & 100 \\
#images/class & \sim 1.2K & \sim 30 & \sim 300 \\
train-weaksup & 10 & \sim 5 & 25 \\
test & 10 & \sim 29 & 50 \\
\hline
\end{tabular}
\caption{\textbf{Dataset statistics.} “\~” indicates that the number of images per class varies across classes and the average value is shown.}
\end{table}
Caltech-UCSD Birds-200-2011 (CUB) [52] have been extensively used. For ImageNet, the 1.2M “train” and 10K “validation” images for 1000 classes are treated as our train-weaksup and test, respectively. For train-fullsup, we use the ImageNetV2 [38]. We have annotated bounding boxes on those images. CUB has 5 994 “train” and 5 794 “test” images for 200 classes. We treat them as our train-weaksup and test, respectively. For train-fullsup, we have collected 1 000 extra images (∼5 images per class) from Flickr, on which we have annotated bounding boxes. For ImageNet and CUB we use the oracle box accuracy BoxAcc.

We contribute a new WSOL benchmark based on the OpenImages instance segmentation subset [3]. It provides a fresh WSOL benchmark to which the models have not yet overfitted. To balance the original OpenImages dataset, we have sub-sampled 100 classes and have randomly selected 29 819, 2 500, and 5 000 images from the original “train”, “validation”, and “test” splits as our train-weaksup, train-fullsup, and test splits, respectively. We use the pixel average precision PxAP. A summary of dataset statistics is in Table 1. Details on data collection and preparation are in Appendix §B.2.

Hyperparameter search. To make sure that the same amount of localization supervision is provided for each WSOL method, we refrain from employing any source of human prior outside the train-fullsup split. If the optimal hyperparameter for an arbitrary dataset and architecture is not available by default, we subject it to the hyperparameter search algorithm. For each hyperparameter, its feasible range, as opposed to sensible range, is used as the search space, to minimize the impact of human bias.

We employ the random search hyperparameter optimization [4]; it is simple, effective, and parallelizable. For each WSOL method, we sample 30 hyperparameters to train models on train-weaksup and validate on train-fullsup. The best hyperparameter combination is then selected. Since running 30 training sessions is costly for ImageNet (1.2M training images), we use 10% of images in each class for fitting models during the search. We verify in Appendix §B.4 that the ranking of hyperparameters is preserved even if the training set is sub-sampled.

5. Experiments

5.1. Evaluated Methods

We evaluate six widely used WSOL methods published in peer-reviewed venues. We describe each method in chronological order and discuss the set of hyperparameters. The full list of hyperparameters is in Appendix §C.1.

Class activation mapping (CAM) [58] trains a classifier of fully-convolutional backbone with the global average pooling (GAP) structure. At test time, CAM uses the logit outputs before GAP as the score map $s_{ij}$. CAM has the learning rate and the score-map resolution as hyperparameters and all five methods below use CAM in the background.

Hide-and-seek (HaS) [25] is a data augmentation technique that randomly selects grid patches to be dropped. The hyperparameters are the drop rate and grid size.

Adversarial complementary learning (ACoL) [56] proposes an architectural solution: a two-head architecture where one adversarially erases the high-scoring activations in the other. The erasing threshold is a hyperparameter.

Self-produced guidance (SPG) [57] is another architectural solution where internal pseudo-pixel-wise supervision is synthesized on the fly. Three tertiary pixel-wise masks (foreground, unsure, background) are generated from three different layers using two thresholding hyperparameters for each mask and are used as auxiliary supervisions.

Attention-based dropout layer (ADL) [6] has proposed a module that, like ACoL, adversarially produces drop masks at high-scoring regions, while not requiring an additional head. Drop rate and threshold are the hyperparameters.

CutMix [55] is a data augmentation technique, where patches in training images are cut and pasted to other images during training. The target labels are also mixed. The hyperparameters are the size prior $\alpha$ and the mix rate $r$.

Few-shot learning (FSL) baseline. The full supervision in train-fullsup used for validating WSOL hyperparameters can be used for training a model itself. Since only a few fully labeled samples per class are available, we refer to this setting as the few-shot learning (FSL) baseline.

As a simple baseline, we consider a foreground saliency mask predictor [29]. We alter the last layer of a fully convolutional network (FCN) into a $1 \times 1$ convolutional layer with $H \times W$ score map output. Each pixel is trained with the binary cross-entropy loss against the target mask, as done in [5, 31, 32]. For OpenImages, the pixel-wise masks are used as targets; for ImageNet and CUB, we build the mask targets by labeling pixels inside the ground truth boxes as foreground [23]. At inference phase, the $H \times W$ score maps are evaluated with the box or mask metrics.

Center-gaussian baseline. The Center-gaussian baseline generates isotropic Gaussian score maps centered at the images. We set the standard deviation to 1, but note that it does not affect the MaxBoxAcc and PxAP measures. This provides a no-learning baseline for every localization method.

5.2. Comparison of WSOL methods

We evaluate the six WSOL methods over three backbone architectures, i.e. VGG-GAP [46, 58], InceptionV3 [51], and ResNet50 [19], and three datasets, i.e. CUB, ImageNet and OpenImages. For each (method, backbone, dataset) tuple, we have randomly searched the optimal hyperparameters over the train-fullsup with 30 trials, totalling about 9 000 GPU hours. Since the sessions are paralleliz-
<table>
<thead>
<tr>
<th>Methods</th>
<th>ImageNet (MaxBoxAcc)</th>
<th>CUB (MaxBoxAcc)</th>
<th>OpenImages (PxAP)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VGG</td>
<td>Inception</td>
<td>ResNet</td>
<td>Mean</td>
</tr>
<tr>
<td>CAM [58]</td>
<td>61.1</td>
<td>65.3</td>
<td>64.2</td>
<td>63.5</td>
</tr>
<tr>
<td>HaS [25]</td>
<td>+0.7</td>
<td>+0.1</td>
<td>-1.0</td>
<td>-0.1</td>
</tr>
<tr>
<td>ACoL [56]</td>
<td>-0.8</td>
<td>-0.7</td>
<td>-2.5</td>
<td>-1.4</td>
</tr>
<tr>
<td>SPG [57]</td>
<td>+0.5</td>
<td>+0.1</td>
<td>-0.7</td>
<td>+0.0</td>
</tr>
<tr>
<td>ADL [6]</td>
<td>-0.3</td>
<td>-3.8</td>
<td>+0.0</td>
<td>-1.4</td>
</tr>
<tr>
<td>CutMix [55]</td>
<td>+1.0</td>
<td>+0.1</td>
<td>-0.3</td>
<td>+0.3</td>
</tr>
<tr>
<td>Best WSOL</td>
<td>62.2</td>
<td>65.5</td>
<td>64.2</td>
<td>63.8</td>
</tr>
<tr>
<td>FSL baseline</td>
<td>62.8</td>
<td>68.7</td>
<td>67.5</td>
<td>66.3</td>
</tr>
<tr>
<td>Center baseline</td>
<td>52.5</td>
<td>52.5</td>
<td>52.5</td>
<td>52.5</td>
</tr>
</tbody>
</table>

Table 2. Re-evaluating WSOL. How much have WSOL methods improved upon the vanilla CAM model? test split results are shown, relative to the vanilla CAM performance (increase or decrease). Hyperparameters have been optimized over the identical train-fullsup split for all WSOL methods and the FSL baseline: (10,5,5) full supervision/class for ImageNet,CUB,OpenImages. Reported results are in the Appendix Table 5; classification accuracies are in Appendix Table 4.

Contrary to the improvements reported in prior work (Appendix Table 5), recent WSOL methods have not led to major improvements compared to CAM, when validated in the same data splits and same evaluation metrics. On ImageNet, methods after CAM are generally struggling: only CutMix has seen a boost of +0.3pp on average. On CUB, ADL has attained a +2.0pp gain on average, but ADL fails to work well on other benchmarks. On the new WSOL benchmark, OpenImages, no method has improved over CAM, except for CutMix (+0.4pp on average). The best overall improvements over CAM (63.8% total mean) is a mere +0.2pp boost by HaS. In general, we observe a random mixture of increases and decreases in performance over the baseline CAM, depending on the architecture and dataset. An important result in the table to be discussed later is the comparison against the few-shot learning baseline (§5.5).

Some reasons for the discrepancy between our results and the reported results include (1) the confounding of the actual score map improvement and the calibration scheme, (2) different types and amounts of full supervision employed under the hood, and (3) the use of different training settings (e.g. batch size, learning rates, epochs). More details about the training settings are in Appendix §C.3.

Which checkpoint is suitable for evaluation? After the acceptance by CVPR 2020, we believe that it is inappropriate to use the best checkpoint for WSOL evaluation. This is because the best localization performances are achieved before convergence in many cases (Appendix §C.2). At early epochs, the localization performance fluctuates a lot, so the peak performance is noise rather than the real performance.
training. However, the maximal IoU performances do not differ as much. This is because WSOL methods exhibit different score distributions (Figure 5 and Appendix §C.4). Fixing the operating threshold \( \tau \) at a pre-defined value, therefore, can lead to an apparent increase in performance without improving the score maps.

Under our threshold-independent performance measures (MaxBoxAcc and \( \text{PxAP} \)) shown in Figure 5, we observe that (1) the methods have different optimal \( \tau \) on ImageNet and (2) the methods do not exhibit significantly different MaxBoxAcc or \( \text{PxAP} \) performances. This provides an explanation of the lack of improvement observed in Table 2. We advise future WSOL researchers to report the threshold-independent metrics.

5.4. Hyperparameter analysis

Different types and amounts of full supervision used in WSOL methods manifest in the form of model hyperparameter selection (§3). Here, we measure the impact of the validation on train-fullsup by observing the performance distribution among 30 trials of random hyperparameters. We then study the effects of feature-erasing hyperparameters, a common hyperparameter type in WSOL methods.

Performance with 30 hyperparameter trials. To measure the sensitivity of each method to hyperparameter choices, we plot the performance distribution of the intermediate models in the 30 random search trials. We say that a training session is non-convergent if the training loss is larger than 2.0 at the last epoch. We show the performance distributions of the converged sessions, and report the ratio of non-convergent sessions separately.

Our results in Figure 6 indicate the diverse range of performances depending on the hyperparameter choice. Vanilla CAM is among the least sensitive, with the smallest standard deviation \( \sigma = 1.5 \) on OpenImages. This is the natural consequence of its minimal use of hyperparameters. We thus suggest to use the vanilla CAM when absolutely no full supervision is available. ACoL and ADL tend to have greater variances across benchmarks (\( \sigma = 11.9 \) and 9.8 on CUB). We conjecture that the drop threshold for adversarial erasing is a sensitive hyperparameter.

WSOL on CUB are generally struggling: random hyperparameters often show worse performance than the center baseline (66\% cases). We conjecture that CUB is a disadvantageous setup for WSOL: as all images contain birds, the models only attend on bird parts for making predictions. We believe adding more non-bird images can improve the overall performances (§3.2).

We show the non-convergence statistics in Figure 6d. Vanilla CAM exhibit a stable training: non-convergence rates are lowest on OpenImages and second lowest on ImageNet. ACoL and SPG suffer from many training failures, especially on CUB (43\% and 37\%, respectively).
In conclusion, vanilla CAM is stable and robust to hyperparameters. Complicated design choices introduced by later methods only seem to lower the overall performances rather than providing new avenues for performance boost.

Effects of erasing hyperparameters. Many WSOL methods since CAM have introduced different forms of erasing to encourage models to extract cues from broader regions (§5.1). We study the contribution of such hyperparameters in ADL, HaS, and ACoL in Figure 7. We observe that the performance improves with higher erasing thresholds (ADL drop threshold and ACoL erasing threshold). We also observe that lower drop rates leads to better performances (ADL and HaS). The erasing hyperparameters introduced since CAM only negatively impact the performance.

5.5. Few-shot learning baselines

Given that WSOL methods inevitably utilize some form of full localization supervision (§3), it is important to compare them against the few-shot learning (FSL) baselines that use it for model tuning itself.

Performances of the FSL baselines (§4.2) are presented in Table 2. Our simple FSL method performs better than the vanilla CAM at 10, 5, and 5 fully labeled samples per class for ImageNet, CUB, and OpenImages, respectively. The mean FSL accuracy on CUB is 92.0%, which is far better than that of the maximal WSOL performance of 70.8%.

We compare FSL against CAM at different sizes of train-fullsup in Figure 8. We simulate the zero-fullsup WSOL performance with a set of randomly chosen hyperparameters (§5.4); for FSL, we simulate the no-learning performance through the center-gaussian baseline.

FSL baselines surpass the CAM results already at 1-2 full supervision per class for CUB and OpenImages (92.4 and 70.9% MaxBoxAcc and PxAP). We attribute the high FSL performance on CUB to the fact that all images are birds; with 1 sample/class, there are effectively 200 birds as training samples. For OpenImages, the high FSL performance is due to the rich supervision provided by pixel-wise masks. On ImageNet, FSL results are not as great: they surpass the CAM result at 8-10 samples per class. Overall, however, FSL performances are strikingly good, even at a low data regime. Thus, given a few fully labeled samples, it is perhaps better to train a model with it than to search hyperparameters. Only when there is absolutely no full supervision (0 fully labeled sample), CAM is meaningful (better than the no-learning center-gaussian baseline).

6. Discussion and Conclusion

After years of weakly-supervised object localization (WSOL) research, we look back on the common practice and make a critical appraisal. Based on a precise definition of the task, we have argued that WSOL is ill-posed and have discussed how previous methods have used different types of implicit full supervision (e.g. tuning hyperparameters with pixel-level annotations) to bypass this issue (§3). We have then proposed an improved evaluation protocol that allows the hyperparameter search over a few labeled samples (§4). Our empirical studies lead to some striking conclusions: CAM is still not worse than the follow-up methods (§5.2) and it is perhaps better to use the full supervision directly for model fitting, rather than for hyperparameter search (§5.5).

We propose the following future research directions for the field. (1) Resolve the ill-posedness via e.g. adding more background-class images (§3.2). (2) Define the new task, semi-weakly-supervised object localization, where methods incorporating both weak and full supervision are studied.

Our work has implications in other tasks where learners are not supposed to be given full supervision, but are supervised implicitly via model selection and hyperparameter fitting. Examples include weakly-supervised vision tasks (e.g. detection and segmentation), zero-shot learning, and unsupervised tasks (e.g. disentanglement [30]).

Acknowledgements. The work is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the MSIP (NRF-2019R1A2C2006123) and ICT R&D program of MSIP/IITP [R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding]. This work was also funded by DFG-EXC-Nummer 2064/1-Projektnummer 390727645 and the ERC under the Horizon 2020 program (grant agreement No. 853489).
References


[32] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency


[34] Dim P Papadopoulos, Alasdair DF Clarke, Frank Keller, and Vittorio Ferrari. Training object class detectors from eye tracking data. In *ECCV*, pages 361–376, 2014.


