

Unsupervised Object Localization with Representer Point Selection

Yeonghwan Song¹Seokwoo Jang¹Dina Katabi²Jeany Son¹¹AI Graduate School, GIST²MIT CSAIL

{yeonghwan.song, jangseokwoo}@gm.gist.ac.kr

dina@csail.mit.edu

jeany@gist.ac.kr

Abstract

We propose a novel unsupervised object localization method that allows us to explain the predictions of the model by utilizing self-supervised pre-trained models without additional finetuning. Existing unsupervised and self-supervised object localization methods often utilize class-agnostic activation maps or self-similarity maps of a pre-trained model. Although these maps can offer valuable information for localization, their limited ability to explain how the model makes predictions remains challenging. In this paper, we propose a simple yet effective unsupervised object localization method based on representer point selection, where the predictions of the model can be represented as a linear combination of representer values of training points. By selecting representer points, which are the most important examples for the model predictions, our model can provide insights into how the model predicts the foreground object by providing relevant examples as well as their importance. Our method outperforms the state-of-the-art unsupervised and self-supervised object localization methods on various datasets with significant margins and even outperforms recent weakly supervised and few-shot methods. Our code is available at: <https://github.com/yeonghwansong/UOLwRPS>

1. Introduction

Object localization is one of the most fundamental problems in computer vision which aims to find a bounding box of a particular object category in a given image. Recent object localization methods achieve impressive performances thanks to advances in deep neural networks (DNNs) [19, 13] and large-scale datasets [33, 37]. Despite their successes, the biggest concern in this task is that collecting datasets with precise box-level annotations is labor-intensive and time-consuming. Recently, object localization methods with less supervision, *i.e.* weakly-supervised methods [25, 52, 48, 12, 53, 46, 31, 44, 3], have been well-studied to mitigate those problems, nevertheless, they require image-level class labels.

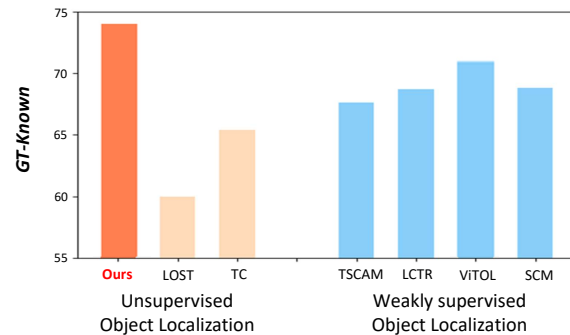


Figure 1. *GT-Known Loc* scores of various methods with respect to different levels of supervision on ImageNet-1K dataset. All methods use ViT-S [13] as a backbone and TC denotes TokenCut [39].

Beyond learning with less supervision, most recent works [42, 41, 51, 1, 21, 27, 36, 45, 35, 39] focus on the object localization task using self-supervised or unsupervised learning that does not require any human annotated labels. These works address the problem of identifying which regions are more likely to contain the *foreground object*, which is a salient object in an image. In order to discover foreground object regions, several methods [1, 21, 27, 36] attempt to use the magnitude of feature vectors as a clue for class-agnostic activation maps (CAAM) [1]. Most of these methods rely heavily on pre-trained models designed for the image classification task. However transferring these models to the object localization task is challenging, since foreground features are not easily distinguished from the background. To address this problem, these works have been focusing on learning to discriminate foreground and background representations from the pre-trained model.

Recent approaches for unsupervised object localization face a limitation in terms of the explainability of the predictions. DNNs are powerful predictors across various domains, but their complicated structure often makes them black-box functions. Class activation maps (CAM) [56] are frequently used to provide visually interpretable information about a specific class the model has learned, but a detailed explanation of how the model makes its predictions

remains unclear. This limitation becomes even more severe for CAAM, which is commonly used in self-supervised object localization methods [1, 21, 27, 36].

In this paper, we propose a simple yet effective unsupervised object localization method based on the representer theorem using self-supervised representation learning models. The proposed method formulates the solution to the object localization task based on representer point selection [47]. This strategy is derived from the representer theorem, where the predictions of a neural network for a given test point find expression through a linear combination of the activations originating from training examples. The importance of each feature, which measures the significance of training examples in contributing to the final prediction, is computed by its norm. By selecting representer points that have the most influence on the model's predictions, our model gains valuable insights into how the model identifies foreground regions. In addition, we introduce a simple extension to weakly supervised object localization, demonstrating zero-shot transferability across different datasets without the need for additional training.

Our experiments are conducted on various datasets for the object localization task to prove the effectiveness of our method and show outstanding performances compared to the state-of-the-art methods with substantial margins. Furthermore, as shown in Figure 1, our method even outperforms some recent weakly supervised and few-shot object localization methods without any fine-tuning or post-processing using additional refinement networks.

We summarize our contributions as follows:

- We propose a novel unsupervised object localization method that leverages the representer theorem on self-supervised representation learning models. The proposed method helps to enhance the interpretability of predictions by providing relevant examples as well as their importance.
- We show that the representer point selection, which has not been extensively explored in previous research on deep neural networks, can be effectively applied to object localization.
- Our method is simple to implement and works well with a variety of self-supervised visual representation learning models. It is also an effective solution due to its good transferability across datasets and easy scalability to weakly supervised object localization.
- Our method shows outstanding results on various datasets for the object localization task and also achieves comparable performances to the state-of-the-art weakly-supervised and few-shot methods.

2. Related Works

Weakly Supervised Object Localization Weakly supervised object localization (WSOL) is a challenging task, aiming to localize an object with only image-level annotations. CAM [56] is the first work to introduce a method of generating a class activation map by projecting the weights of the classifier on the feature maps. However, CAM has a limitation in finding parts of the objects that are not relevant for classification. To address this problem, various CAM-based WSOL methods have been proposed [25, 52, 12, 53, 46, 31, 45, 44, 3]. HaS [25] used the random masking of image patches to encourage the model to observe more object regions. ACoL [52] adopted two adversarial classifiers to catch the relevant parts. ADL [12] and AE [43] progressively erased the discriminative parts on the feature map of each convolutional layer, ensuring that the focus was not solely on these discriminative areas. SPG [53] and I²C [54] utilized shallow convolutional feature maps and pixel-level correlations between two different images to adjust an activation map respectively. PSOL [50] proposed a new task of pseudo-supervised object localization which consists of two independent tasks: class-agnostic object localization and object classification. PSOL [50] and SEM [55] trained a localizer with an object bounding box generated by DDT [42] and edges of an input image extracted from the canny edge detector as pseudo labels, respectively. Recently, most CAM-based methods [46, 40, 31, 45, 44, 3] have focused on learning discriminative features for the foreground and background, while vision transformer-based methods [14, 17, 10, 2] have emerged to exploit long-range dependency.

Unsupervised Object Localization Unsupervised object discovery is strongly related to co-localization [42, 1, 21] and co-segmentation [49, 28] tasks. Early work for unsupervised object discovery [11] proposed a part-based region matching using off-the-shelf region proposals. With the advance of deep learning, DDT [42] and SCDA [41] introduced methods to mine the statistical properties of object regions in an image using features from pre-trained CNNs. MO [51] mined a frequently activated pattern from multi-scale feature maps. These methods achieved impressive results not only in object discovery but also in co-localization, which aims to localize objects in a dataset that consists of only one coarse category. PsyNet [1] introduced a simple and practical framework for the co-localization task using self-supervised learning and class-agnostic activation mapping. Ki *et al.* [21] employed contrastive learning with the assumption of consistency between image-level and activation-level transformations. Su *et al.* [36] proposed a joint graph partition method splitting foreground and background regions in a paired image feature graph. C²AM [45] proposed to apply contrastive learning to disentangle the

feature map into foreground and background with an additional trainable layer generating a class-agnostic activation map. LOST [35] and TokenCut [39] utilize a pre-trained vision transformer model [13] by DINO [5], which has the ability to produce fine segmentation masks using attention maps with [CLS] token. LOST [35] suggested a heuristic seed selection and expansion method, and TokenCut [39] employed a normalized cut algorithm on the graph using the similarity between patches.

Self-supervised Visual Representation Learning Self-supervised visual representation learning (SSL) methods have been proposed to learn meaningful visual representation without manual annotations. The network learns visual representation by training pretext tasks rather than the task intended to be solved. For example, Noroozi *et al.* [32] trained CNN to solve a jigsaw puzzle, Larsson *et al.* [26] reconstructed color images from gray-scale images, and Gidaris *et al.* [15] predicted discrete rotation angles of rotated images. Recently, contrastive learning-based SSL methods [18, 6, 8, 16] have emerged with comparable performances to the supervised learning methods. They focus on modeling image similarity and dissimilarity between pairs of images [6, 18] or only consider image similarity [16, 8] based on data augmentation. DINO [5] proposed a knowledge distillation strategy to train the vision transformer model. These pre-trained SSL models are utilized in various downstream tasks of computer vision with fine-tuning. On the contrary, in this work, we focus on leveraging the pre-trained SSL models for the unsupervised object localization task without any fine-tuning.

3. Background

Representer Theorem. Representer theorem [34] provides a mathematical basis for traditional machine learning methods. Let $L(\mathbf{x}_n, y_n, \mathbf{w})$ be the loss function, and $\frac{1}{N} \sum_{n=1}^N L(\mathbf{x}_n, y_n, \mathbf{w})$ be the empirical risk over a reproducing kernel Hilbert space (RKHS) H_k . Then the representer theorem states that an optimal solution of a regularized empirical risk minimization (ERM) problem can be represented as a linear combination of a positive definite kernel k on the input set \mathcal{X} over H_k . The optimal parameter \mathbf{w}^* can be expressed through ERM as follows:

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w} \in H_k} \left\{ \frac{1}{N} \sum_{n=1}^N L(\mathbf{x}_n, y_n, \mathbf{w}) + \lambda \|\mathbf{w}\|^2 \right\}. \quad (1)$$

can be rewritten by the representer theorem

$$\mathbf{w}^*(\cdot) = \sum_{i=1}^n \alpha_i k(\cdot, \mathbf{x}_i) = \sum_{i=1}^n \alpha_i \langle \varphi(\cdot), \varphi(\mathbf{x}_i) \rangle, \quad (2)$$

where $\alpha_i \in \mathbb{R}$, and φ denotes a mapping function into H_k . For further details for the Representer theorem, please refer

to the work by [34, 47].

Representer Point Selection for DNNs. The representer theorem is originally developed for non-parametric predictors, where the parameters lie in a reproducing kernel Hilbert space. In deep neural networks, however, it is difficult to find a global solution for empirical risk minimization. To address this issue, [47] proposes a technique to decompose the pre-activation predictions into a linear combination of the activation values of the training points. The neural network can be represented as

$$\hat{y}_t = \sigma(\phi(\Phi(\mathbf{x}_t, \mathbf{W}_\Phi), \mathbf{w}_\phi)) = \sigma(\mathbf{w}_\phi^\top \mathbf{f}_t), \quad (3)$$

where σ is the activation function, $\mathbf{f}_n = \Phi(\mathbf{x}_n, \mathbf{W}_\Phi)$ is the last intermediate layer feature of a training example \mathbf{x}_n in DNNs, ϕ denotes the last prediction layer for a specific task, and \mathbf{W}_Φ and \mathbf{w}_ϕ denote the parameters of Φ and ϕ , respectively. Let \mathbf{W}^* be a stationary point of the objective (1). Then we have decomposition as follows:

$$\mathbf{w}_\phi^* \top \mathbf{f}_t = \sum_{n=1}^N \alpha_n k(\mathbf{x}_t, \mathbf{x}_n) = \sum_{n=1}^N \alpha_n \mathbf{f}_n^\top \mathbf{f}_t, \quad (4)$$

$$\text{where } \alpha_n = \frac{1}{-2\lambda N} \frac{\partial L(\mathbf{x}_n, y_n, \mathbf{W})}{\partial \phi(\mathbf{f}_n, \mathbf{w}_\phi)}, \quad (5)$$

and $\alpha_n \mathbf{f}_n^\top \mathbf{f}_t$ is a representer value of a training sample \mathbf{x}_n given a test sample \mathbf{x}_t . This representer value is assigned to each training point reflect their importance on the learned parameters and α_n is the global sample importance which is used to evaluate the importance of the training data \mathbf{x}_n . Consequently, this approach enables the selection of representer points – training instances with either large or small representer values – to enhance comprehension of the model’s predictions.

4. Method

In this section, we discuss how we leverage the representer point selection [47] to solve unsupervised object localization (UOL) task, and our extension to weakly supervised object localization (WSOL) and zero-shot transferring.

4.1. Representer Point Selection for UOL

Inspired by the representer point selection, we formulate the object localization task as identifying the important regions in training examples, called representer points, using a pre-trained self-supervised model. Unlike other tasks such as classification, regression, and retrieval that are covered in [47], the object localization task requires dense predictions within a single image. To address this, we address all elements within image feature maps rather than relying solely on a single image-level feature vector.

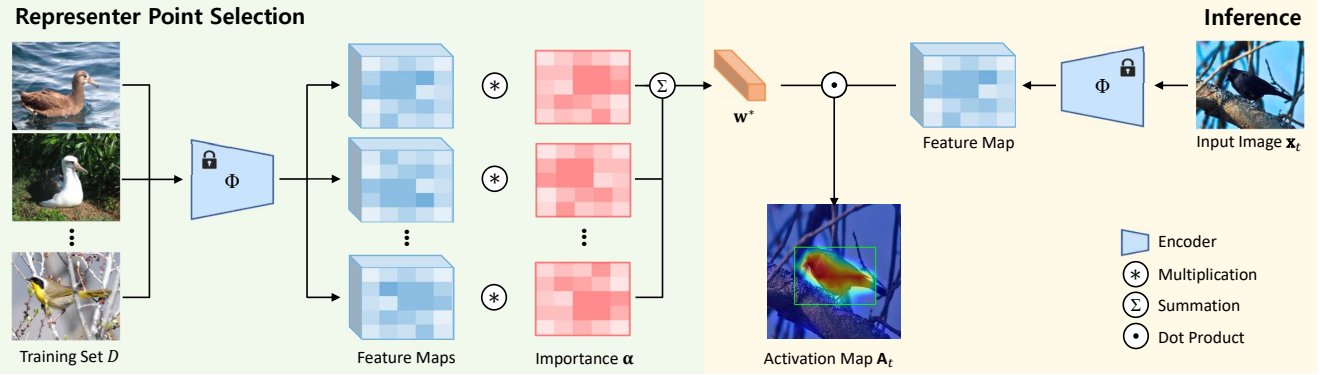


Figure 2. Pipelines of our unsupervised object localization method, which consists of two separate stages: representer point selection and inference. In the representer point selection stage, we compute a foreground predictor \mathbf{w}^* using images in the training dataset. In the inference stage, the activation maps are computed by a dot product between the feature map of the test image and \mathbf{w}^* .

As our focus lies in unsupervised learning, the empirical risk or the loss function can be defined as $\frac{1}{N_D} \sum_{n=1}^{N_D} L(\mathbf{x}_n, \mathbf{W}_\Phi)$ without a label y_n given pre-trained encoder Φ and training set D . Then the activation maps \mathbf{A}_t for foreground regions can be computed by aggregating feature maps using the global sample importance, which indicate how much each training example contributes to the prediction of being foreground:

$$\text{vec}(\mathbf{A}_t)_j = \underbrace{\sum_{n=1}^{N_D} \sum_{i=1}^{WH} \alpha_{n,i} \hat{\mathbf{f}}_{n,i}^\top \hat{\mathbf{f}}_{t,j}}_{\mathbf{w}^*} = \mathbf{w}^{*\top} \hat{\mathbf{f}}_{t,j}, \quad (6)$$

where $\mathbf{f}_{n,i} = \Phi(\mathbf{x}_n, \mathbf{W}_\Phi)_i$ denotes i^{th} feature vector in the feature map of an image \mathbf{x}_n , which has been encoded using a self-supervised pre-trained encoder Φ , $\hat{\mathbf{f}} = \frac{\mathbf{f}}{\|\mathbf{f}\|}$ indicates the normalized feature vector, α is the global sample importance, and N_D, W, H denote the number of images in the training dataset, the width of the feature map, and the height of the feature map, respectively. Here, we refer to $\alpha_{n,i} \hat{\mathbf{f}}_{n,i}^\top \hat{\mathbf{f}}_{t,j}$ as the *representer value* for each training image patch $\mathbf{x}_{n,i}$ given a testing image patch $\mathbf{x}_{t,j}$, corresponding to receptive fields of $\mathbf{f}_{n,i}$ and $\mathbf{f}_{t,j}$, respectively. In Figure 2, we illustrate the entire structure of the proposed representer point selection method to compute a foreground predictor \mathbf{w}^* and inference pipelines using it.

Global Sample Importance. α_n is referred to as the global sample importance since it is independent of the testing examples but reliant on the empirical risk of the training set. In order to tackle the unsupervised localization task, we first design the loss function which is suitable for an unsupervised setting to compute α . Let \mathbf{w} be the parameters of a binary classifier for foreground predictions given feature vectors, then the loss function can be written as follows:

$$L(\mathbf{x}_{n,i}, y_{n,i}, \mathbf{W}_\Phi) = -y_{n,i} \mathbf{w}^\top \hat{\mathbf{f}}_{n,i}, \quad (7)$$

where $y_{n,i} \in \{-1, 1\}$ denotes a label of $\mathbf{x}_{n,i}$, which is an image patch corresponding to a receptive field of $\mathbf{f}_{n,i}$. In the unsupervised object localization task, however, the goal is to find foreground regions in images without the use of labels. Thus, instead of using $y_{n,i}$, we use the norm of the feature vector to determine a soft pseudo label of the example for the objective function as follows:

$$L(\mathbf{x}_{n,i}, \mathbf{W}_\Phi) = -(\|\mathbf{f}_{n,i}\| - \tau) \mathbf{w}^\top \hat{\mathbf{f}}_{n,i}, \quad (8)$$

where τ denotes a foreground threshold.

We estimate soft labels of inputs based on the norm of feature vectors because the norm can serve as an important indicator of the features that the network emphasized during training with the self-supervised contrastive loss. In other words, a higher norm of a feature vector indicates that the corresponding feature had more weight in the model training. These features are extracted from regions of the image that contain rich information to distinguish them from other instances, and these regions are more likely to be associated with significant objects. Likewise, prior self-supervised colocalization studies [1, 21, 27, 36] also use the magnitude of feature vectors as a clue to discover object regions.

Hence, the global sample importance is computed as:

$$\alpha_{n,i} = -\frac{1}{2\lambda N_D H W} \frac{\partial L(\mathbf{x}_{n,i}, \mathbf{W}_\Phi)}{\partial \mathbf{w}^\top \hat{\mathbf{f}}_{n,i}} \quad (9)$$

$$= \frac{1}{2\lambda N_D H W} (\|\mathbf{f}_{n,i}\| - \tau) = \frac{1}{C} (\|\mathbf{f}_{n,i}\| - \tau). \quad (10)$$

Here, the sign of the global sample importance is related to whether to *representer points* – training examples that have large or small representer values – contribute to the foreground or background. In order to have large representer values, both α and similarity between two features, expressed as $\mathbf{f}_n^\top \mathbf{f}_t$, should have large values. By analyzing the representer values, we can gain insight into how the model predicts foreground regions by providing relevant examples

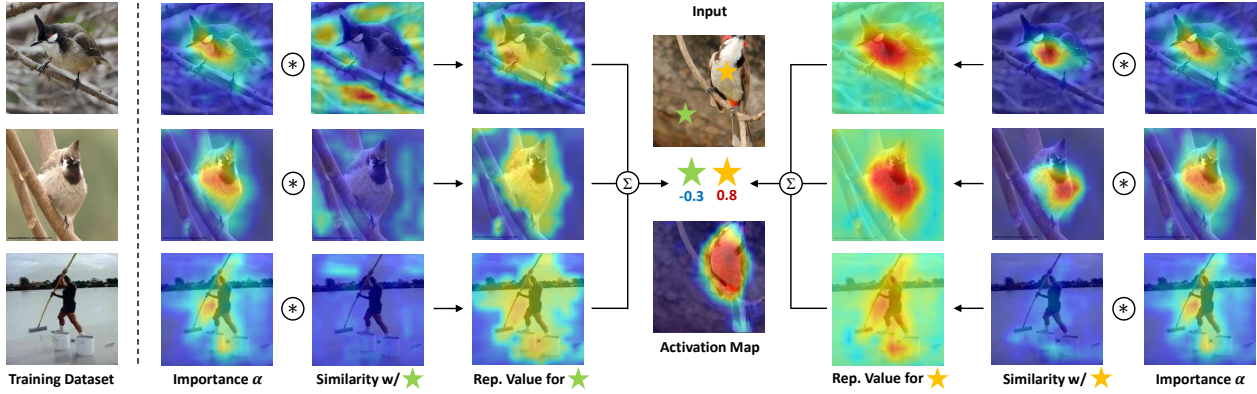


Figure 3. Illustration of how our method computes activation maps using two example points, yellow and green stars. To predict the point as a foreground region, two factors should be considered: the global sample importance and the similarity of features. the representer values are obtained by multiplying these two factors, and a high representer value indicates a stronger contribution to the foreground prediction.

Algorithm 1: Representer Point Selection

```

# model: pre-trained encoder  $\Phi$ 
# const: the constant value for  $\alpha$ 
# th: foreground threshold of norm  $\tau$ 
for x in data loader:
    f = model(x) # f: [N, C, H, W]
    v += sum(f, d=(N, H, W))
    u += sum(normalize(f, d=C), d=(N, H, W))
w = v/const - th*u/const
return w

```

as well as their importance. Figure 3 shows an example of global sample importance, the similarity between features, and representer values for given points in a given test image.

Straightforward Computation of \mathbf{w}^* We then compute a foreground predictor \mathbf{w}^* using $\alpha_{n,i}$ in Eq (10) as follows:

$$\mathbf{w}^* = \frac{1}{C} \underbrace{\sum_{n=1}^{N_D} \sum_{i=1}^{HW} \mathbf{f}_{n,i}}_{\mathbf{v}} - \tau \underbrace{\sum_{n=1}^{N_D} \sum_{i=1}^{HW} \hat{\mathbf{f}}_{n,i}}_{\mathbf{u}}, \quad (11)$$

The optimal \mathbf{w}^* can be computed by taking the difference between the mean feature vector of all images in the dataset and the mean of the normalized feature vectors of all images in the dataset, multiplied by two constant values. Algorithm 1 illustrates the Python-style pseudo code of the proposed representer point selection method to compute the foreground predictor \mathbf{w}^* and it clearly shows the simplicity and reproducibility of our method.

4.2. Towards WSOL and Zero-shot Transferring

Our method can be easily extended to weakly supervised and zero-shot transferring settings with a simple modification, since activation maps are computed by the linear combination of activations of training points directly without

learning. Given a class label c , the activation map \mathbf{A}^c is computed as follows:

$$\text{vec}(\mathbf{A}_t^c)_j = \sum_{n \in D_c} \sum_{i=1}^{WH} \alpha_{n,i} \hat{\mathbf{f}}_{n,i}^\top \hat{\mathbf{f}}_{t,j} = \mathbf{w}_c^\top \hat{\mathbf{f}}_{t,j}, \quad (12)$$

where D_c denotes the subset of the training dataset D that contains only the images assigned to class c . Note that, although WSOL allows for the use of image-level class labels, we utilize an off-the-shelf classification model to determine the class of each image in D during our experiments in order to ensure a fair comparison with other methods [45, 50]. Extending our method to zero-shot transferring scenarios can be simply done by computing the weight vector \mathbf{w}^* on the different datasets in which the classes do not overlap with the test dataset.

5. Experiments

5.1. Implementation Details

We adopt ResNet50 [19] and ViT-S [13] as backbones for SSL pre-trained model, utilizing publicly available pre-trained weights. We use the input image size of 224×224 for the representer point selection. To evaluate object localization performances on fine-grained classification datasets, the input image is resized to 480×480 and then center-cropped to a size of 448×448 . In the case of ImageNet-1K, the input image is resized to 256×256 and then center-cropped to a size of 224×224 . We use $\|\mathbf{v}\|/\|\mathbf{u}\|$ for τ , and min-max normalization to estimate the bounding box from the generated activation map. We also set $\lambda = 0.001$ as in [47], but changing this value does not have an impact on the final results due to the use of min-max normalization.

5.2. Datasets and Evaluation Metrics

We conduct experiments on the four fine-grained datasets for unsupervised object localization to evaluate our

Table 1. Comparison between our method and the state-of-the-art self-supervised/unsupervised methods in terms of *GT-Known Loc* performance on ImageNet-1K and four fine-grained datasets: CUB200-2011, Stanford Cars, FGVC-Aircraft, and Stanford Dogs. * denotes our reproduced results from their official code. † and ‡ indicate a backbone initialized by MoCo v2 and DINO pre-trained weights, respectively.

Method	Backbone	\mathcal{T}	CUB-200-2011	Stanford Cars	FGVC-Aircraft	Stanford Dogs	ImageNet-1K
<i>Self-supervised methods</i>							
ORE '22 [27]	VGG16	✓	87.72	97.43	97.86	85.45	-
JGP '22 [36]	VGG16	✓	88.83	97.73	96.72	-	-
PsyNet '20 [1]	SE-ResNet50	✓	85.10	98.81	97.81	77.84	-
Ki <i>et al.</i> '21 [21]	SE-ResNet50	✓	85.93	98.95	98.75	80.32	-
PSOL '20 [50]	ResNet50	✓	90.00	-	-	-	65.44
C ² AM '22 [45]	ResNet50 [†]	✓	89.90	99.44*	98.65*	89.91*	66.51
C ² AM (+PSOL) '22 [45]	ResNet50 [†]	✓	91.54	-	-	-	68.07
<i>w/o finetuning</i>							
DDT '19 [42]	VGG16	✗	82.26	71.33	92.53	-	-
MO '20 [51]	VGG16	✗	80.45	92.51	94.94	80.70	-
LOST '21 [35]	ViT-S [‡]	✗	89.70	-	-	-	60.00
TokenCut '22 [39]	ViT-S [‡]	✗	91.80	-	-	-	65.40
Ours	VGG16	✗	90.47	98.31	96.88	89.67	59.07
Ours	ResNet50 [†]	✗	96.67	99.69	98.71	95.07	66.89
Ours	ViT-S [‡]	✗	91.61	99.84	99.22	94.71	73.65

Table 2. Comparison of localization performances in terms of *Top-1*, *Top-5* and *GT-known Loc* on CUB-200-2011 test set and ImageNet-1K validation set. Self-supervised and unsupervised methods including ours employ an additional classifier to evaluate in WSOL setup following PSOL [50]. \mathcal{T} and \mathcal{S} indicate whether each method uses training and image-level supervision (B for bounding box annotations), respectively. †, ‡, and \circ indicate the use of MoCo v2, BYOL, and DINO pre-trained weights, respectively.

Method	Backbone	\mathcal{S}	\mathcal{T}	CUB-200-2011			ImageNet-1K		
				<i>Top-1 Loc</i>	<i>Top-5 Loc</i>	<i>GT-Known</i>	<i>Top-1 Loc</i>	<i>Top-5 Loc</i>	<i>GT-Known</i>
<i>Few-shot method</i>									
SPNet '19 [29]	ResNet50	✓(B)	✓	-	-	95.80	-	-	71.50
<i>Weakly supervised methods</i>									
FAM '21 [31]	ResNet50	✓	✓	73.74	-	85.73	54.46	-	64.56
CREAM '22 [46]	ResNet50	✓	✓	76.03	-	89.88	55.66	-	69.31
BGC '22 [22]	ResNet50	✓	✓	73.16	86.68	91.60	53.76	65.75	69.89
DAOL '22 [58]	ResNet50	✓	✓	66.65	-	81.83	55.84	-	70.27
BAS '22 [44]	ResNet50	✓	✓	77.25	90.08	95.13	57.18	68.44	71.77
BagCAM '22 [57]	ResNet50	✓	✓	69.67	-	94.01	44.24	-	72.08
TS-CAM '21 [14]	ViT-S	✓	✓	71.30	83.80	87.70	53.40	64.30	67.60
LCTR '22 [10]	ViT-S	✓	✓	79.20	89.90	92.40	56.10	65.80	68.70
ViTOL '22 [17]	ViT-S	✓	✓	-	-	80.90	58.64	-	72.51
SCM '22 [2]	ViT-S	✓	✓	76.40	91.60	96.60	56.10	66.40	68.80
<i>Self-supervised methods</i>									
PSOL '20 [50]	ResNet50	✗	✓	70.68	86.64	90.00	53.98	63.08	65.44
C ² AM '22 [45]	ResNet50 [†]	✗	✓	-	-	89.90	-	-	66.51
C ² AM (+PSOL) '22 [45]	ResNet50 [†]	✗	✓	74.76	87.37	91.54	54.65	65.05	68.07
<i>w/o finetuning</i>									
LOST '21 [35]	ViT-S [◦]	✗	✗	71.30	-	89.70	49.00	-	60.00
TokenCut '22 [39]	ViT-S [◦]	✗	✗	72.90	-	91.80	52.30	-	65.40
Ours	ResNet50 [†]	✗	✗	79.57	92.60	96.67	55.60	66.05	69.10
Ours	ResNet50 [‡]	✗	✗	79.55	92.58	96.72	56.03	66.78	69.93
Ours	ViT-S [◦]	✗	✗	74.97	84.24	91.61	62.03	71.96	74.44

method: CUB-200-2011 [37], Stanford Cars [23], FGVC-Aircraft [30], and Stanford Dogs [20]. We also evaluate our method on ImageNet-1K to compare with other state-of-the-art methods in unsupervised and WSOL setups. In addition, we evaluate the segmentation quality of our method by comparing it with the WSOL state-of-the-art methods on CUB-200-2011 and OpenImages-30K [4] datasets.

Following [56], we use Top-1 (*Top-1 Loc*), Top-5 (*Top-5 Loc*) and GT-known localization (*GT-known Loc*) accuracy for evaluation. *GT-known Loc* computes the ratio of the images where the Intersection over Union (IoU) between the estimated bounding box and the known ground-truth boxes is greater than 50%. *Top-1 Loc* and *Top-5 Loc* compute the ratio of the images correctly classified with Top-1 and Top-

Table 3. Comparison between the proposed method and the state-of-the-art weakly supervised object localization methods in terms of $PxAP$ and $PloU$ on CUB-200-2011 and OpenImages-30K. * denotes that the results are obtained from their official code.

Method	CUB-200-2011		OpenImages-30K	
	$PloU$	$PxAP$	$PloU$	$PxAP$
<i>WSOL methods</i>				
BGC '22 [22]	-	-	-	63.70
CREAM '22 [46]	-	-	-	64.70
DAOL '22 [58]	56.18	74.70	49.68	65.42
BagCAM-22 [57]	74.51	90.38	52.17	67.68
<i>Self-supervised methods</i>				
C ² AM '22 [45]	69.65*	88.74*	47.75*	58.28*
<i>Ours with SSL/WSOL pre-trained backbone</i>				
MoCo v2 [7] + Ours	71.28	87.03	50.20	62.18
DenseCL [38] + Ours	70.11	87.74	54.68	68.32
DINO ViT-S [5] + Ours	78.97	94.37	61.24	73.14
C ² AM [45] + Ours	71.54	90.22	56.15	69.79
DAOL [58] + Ours	67.37	85.99	53.25	67.62

5 predictions, respectively, with the condition of *GT-Known Loc*. For segmentation evaluation, *Pixel-wise average precision (PxAP)* computes the area under the pixel precision-recall curve and *Peak-IoU (PloU)* is the best IoU score with various thresholds.

5.3. Results

Unsupervised Object Localization. Our method is primarily designed for the unsupervised object localization (UOL) task which does not focus on localizing objects for specific categories. In this task, w^* is computed using the entire dataset, regardless of the number of categories in the dataset. We report the performances of our method and the state-of-the-art methods in terms of *GT-known Loc* on five commonly used datasets in Table 1. We compare our method with recent self-supervised object localization methods including ORE [27], PsyNet [1], Ki *et al.* [21], JGP [36] and C²AM [45], as well as object localization methods without finetuning including DDT [42], MO [51], LOST [35], and TokenCut [39]. As shown in Table 1, our method significantly surpasses other methods on all benchmark datasets. Our method outperforms not only unsupervised methods but also the self-supervised methods where the models are finetuned on the training split of the target datasets. Even with a shallower architecture, such as VGG-16, our method shows better performance compared to the methods using ResNet50 on CUB-200-2011. For the VGG-16 backbone, we employ ImageNet pre-trained weights.

Weakly Supervised Object Localization. In addition, as mentioned in Section 4.2, our method can be extended to the weakly supervised object localization (WSOL) setting by computing w_c^* for each class c using a pre-trained classifier. To evaluate on *Top-1/5 Loc* metrics, we follow [50] which divides the WSOL task into two sub-tasks: object localiza-

Table 4. *GT-Known Loc* results of our method on various SSL models with ResNet50 backbones except DINO ViT.

Model	BYOL	MoCo v2 / v3	DINO Res / ViT	SimSiam
CUB	96.72	96.67 / 97.03	96.15 / 91.61	94.99
ImageNet	67.05	66.89 / 66.31	65.15 / 73.65	64.10
ImageNet (w_c^*)	69.93	69.10 / 69.34	69.15 / 74.44	67.57

Table 5. *GT-Known Loc* results of models with and without our inference using the representer point selection.

Method	CUB	ImageNet
MoCo v2 + CAAM	86.93	61.19
MoCo v2 + Ours	96.67 (+9.74)	69.10 (+7.91)
C ² AM [45]	89.90	66.51
C ² AM [45] + PSOL	91.54 (+1.64)	68.07 (+1.56)
C ² AM [45] + Ours	96.86 (+6.96)	69.80 (+3.29)
DAOL [58]	81.83	70.27
DAOL [58] + Ours	89.44 (+7.61)	70.97 (+0.70)

tion and classification. We first localize objects, and then we determine classes for localized objects using the pre-trained classifier, *i.e.* ResNet50 pre-trained on ImageNet.

Table 2 shows the performance of our method and the state-of-the-art weakly supervised object localization methods, including FAM [31], CREAM [46], BGC [22], BAS [44], DAOL [58], BagCAM [57], TS-CAM [14], LCTR [10], ViTOL [17], and SCM [2] on CUB-200-2011 and ImageNet-1K datasets. We also compare our method with PSOL [50] and C²AM [45] which are self-supervised methods. Our method shows outstanding results on CUB-200-2011 and comparable results on ImageNet-1K, compared to the advanced weakly supervised methods without any supervision and training. As shown in Table 2, our method outperforms all self-supervised methods with the additional classifier using the same architecture with localization backbone [19, 13]. For a fair comparison, we report C²AM initialized with MoCo v2 pre-trained weights as well as its refinements with PSOL. Note that, without any training and post-processing as PSOL, our method significantly surpasses C²AM (+PSOL) and PSOL on CUB-200-2011. For a fine-grained dataset such as CUB-200-2011, we utilize a class-agnostic foreground predictor w^* for comprehensive object localization rather than a class-specific one.

Our method shows outstanding performance on not only localization but also segmentation, as shown in Table 3, particularly with the model trained by self-supervised pixel-wise contrastive learning (*i.e.* DenseCL [38]) on the OpenImages-30K dataset. All the methods compared in this table employ ResNet50 as their backbone, with the exception of DINO. Upon utilizing the DINO ViT model as our backbone, we observe a noteworthy and substantial enhancement in performance. Exploiting our method as an inference method for feature extractors trained by the WSOL methods [45, 58] further enhances performance, yielding

Table 6. *GT-Known Loc* results of C^2 AM and ours using the model trained/computed on CIFAR₁₀, CIFAR₁₀₀, and CUB-200-2011 to show the zero-shot transferability of the methods.

Method	Training	CUB	Cars	Aircraft	Dogs	ImgNet
C^2 AM	CIFAR ₁₀	73.18	96.74	98.35	80.54	56.51
Ours	CIFAR ₁₀	94.24	99.04	98.26	92.39	63.87
C^2 AM	CIFAR ₁₀₀	65.12	93.40	95.68	78.75	57.29
Ours	CIFAR ₁₀₀	92.51	98.13	97.66	89.67	63.72
C^2 AM	CUB	89.90	90.85	89.80	90.44	59.08
Ours	CUB	96.67	98.20	98.32	93.14	65.24

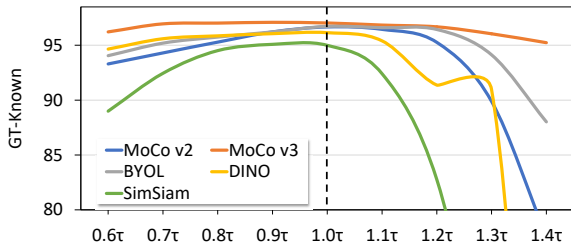


Figure 4. *GT-Known* scores with respect to different thresholds using various pre-trained models on the CUB-200-2011 dataset. The proposed threshold τ consistently achieves near-optimal scores.

significant improvements. It clearly shows that our method demonstrates consistently good performances on various datasets and diverse metrics.

5.4. Ablation Study

We conduct comprehensive ablation studies to verify the effectiveness of our method. We adopt the *GT-Known Loc* metric as a performance evaluation for all experiments.

Robustness across various SSL pre-trained models.

We report *GT-known Loc* scores of our method using popular self-supervised pre-trained models including BYOL [16], MoCo v2 [7], MoCo v3 [9], DINO [5] and SimSiam [8] to support compatibility of our method in Table 4. Our method consistently shows good performance with various pre-trained models. We believe that our method generates more precise activation map of the object, as long as good representations are provided by pre-trained models no matter how they are trained.

Effectiveness of representer point selection. Table 5 reports the results of our inference method using representer point selection on several models including C^2 AM [45] and DAOL [58] as well as class-agnostic activation map (CAAM) proposed in [1]. The backbones for our model and C^2 AM are initialized by MoCo v2 [7] pre-trained weights. Even with the state-of-the-art models trained for the object localization task, applying our inference improves their performance with significant margins on both CUB-200-2011 and ImageNet-1K datasets. In the case of C^2 AM, our inference significantly improves its performance more than us-

Table 7. Ablation study on the dataset sampling ratio for the representer point selection. \dagger denotes the use of a CIFAR₁₀ for the foreground predictor w^* , and \ddagger indicates the use of ResNet50 as the backbone; otherwise, ViT-s is employed.

Method (sampling ratio)	CUB	ImageNet
<i>w/o finetuning</i>		
Ours (100%)	91.62	73.65
Ours (10%)	91.70 ± 0.11	73.64 ± 0.03
Ours (1%)	91.65 ± 0.42	73.63 ± 0.03
Ours (0.1%)	91.08 ± 1.02	73.67 ± 0.06
Ours (0% \dagger)	92.73	73.10
<i>TokenCut [39] (100%)</i>		
TokenCut [39] (100%)	91.80	65.40
<i>Weakly supervised method</i>		
ViTOL [17] (100%)	80.90	72.51
<i>Few-shot method</i>		
SPNet \ddagger [29] ($\sim 10\%/\sim 1\%$)	95.80	71.50

ing post-processing by PSOL [50] which requires additional training using a ResNet50 network.

Zero-shot transferability across datasets. We also examine the transferability of our model across different datasets in Table 6. We compare our method’s performance with C^2 AM [45], using our foreground predictor and C^2 AM’s trained model from CIFAR₁₀, CIFAR₁₀₀ [24] and CUB-200-2011 datasets, and then testing on other datasets without additional training. In this comparison, we ensure fairness by using class-agnostic weights and initializing the ResNet50 model with pre-trained weights from MoCo v2 [7]. The results in Table 6 highlight that our model, leveraging representer points from these datasets, can be transferred to various datasets with strong performance, while C^2 AM’s performance notably drops.

Impact of sampling rate on a dataset D .

In order to show the impact of the number of samples for selecting representer points, we randomly sample images on the training set. In Table 7, to illustrate the effect of sampling rate, we report the results of our method with DINO ViT-s using 100%, 10%, 1%, 0.1% and 0% of the dataset, where 0% uses CIFAR₁₀ dataset to establish representer points. For 10%, 1%, and 0.1% sampling rates, we compute mean and standard deviation with 10 random trials. As shown in Table 7, the number of samples to compute the foreground predictor has no significant effect on performance drops. In fact, in certain instances, it even produces better results compared to using the entire dataset. We also report results of ViT-s based models, such as TokenCut [39], ViTOL [17], and a few-shot method, SPNet [29]. SPNet uses 5-shots for 200 fine-grained classes on CUB-200-2011 and 10-shots for 1K classes on ImageNet-1K, with object localization annotations, while ViTOL and TokenCut uses a full dataset to train the model. Even with an extremely small amount of samples in the dataset, our method surpass SPNet, ViTOL,

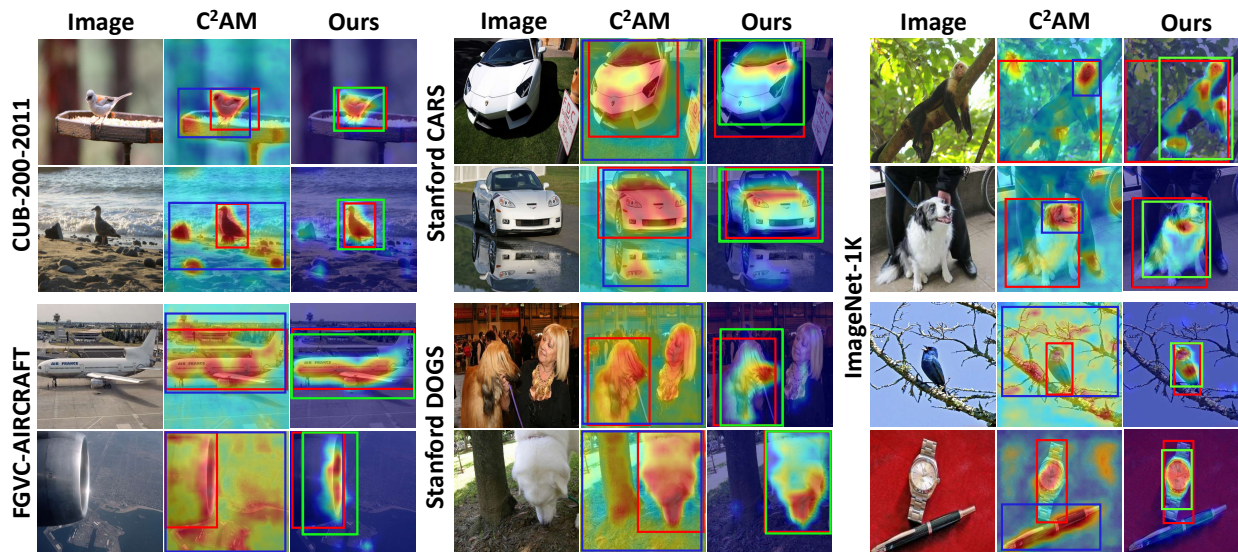


Figure 5. Qualitative results of our method compared to C^2AM . Ground truth bounding boxes are denoted in red, whereas predicted object localization bounding boxes are denoted in green or blue. Best viewed in color.

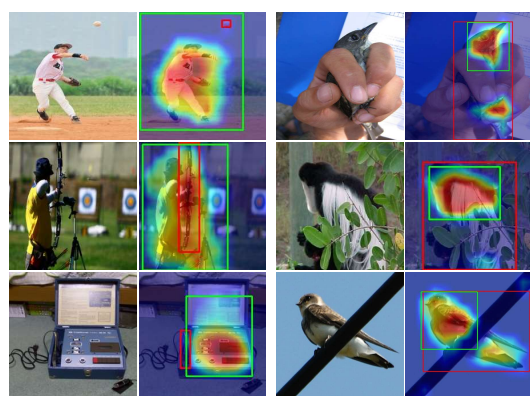


Figure 6. Failure cases of our method on ImageNet-1K (left) and CUB-200-2011 (right) datasets.

and TokenCut on ImageNet.

Impact of threshold τ . We conduct an empirical analysis to show the impact of a threshold $\tau = \|\mathbf{v}\|/\|\mathbf{u}\|$. As demonstrated by the results in Figure 4, we found that the proposed threshold τ consistently achieved nearly optimal scores, even when tested with various thresholds. Our defined threshold τ also consistently performs well across diverse models. Further detailed theoretical analysis for τ is provided in *our supplementary material*.

5.5. Qualitative Results and Failure Cases

We illustrate the qualitative results of our method and C^2AM in Figure 5. Both methods use the MoCo v2 model with the ResNet50 backbone. Without any training on the

dataset unlike C^2AM , our method successfully localizes objects on various datasets. Compared with the other method, our activation map shows the more precise area of the object, and activations in the background are suppressed.

We also demonstrate some failure cases in Figure 6. Since our method only relies on features from the pre-trained networks without additional training, it has limited ability to find the small target object particularly in scenarios where multiple objects coexist within the image. We observe that our method tends to find objects with a dominant size in the image. Moreover localizing the whole object is challenging in cases where the object is either occluded or fragmented. These failure cases, however, are not only our limitations but also the limitations of current unsupervised and weakly supervised object localization methods.

6. Conclusion

In this paper, we propose a simple and effective unsupervised object localization method without using additional training and supervision. We leverage the representative point selection in the object localization task, which allows us to explain the decision in terms of the activations of data points. The proposed method outperforms not only the state-of-the-art self-supervised and unsupervised object localization methods by a significant margin but also surpasses recent weakly supervised and few-shot methods.

Acknowledgement. This work was supported by the IITP grants (No.2019-0-01842, No.2021-0-02068, No.2022-0-00926) funded by MSIT, the ISTD program (No.20018334) funded by MOTIE, and the GIST-MIT Research Collaboration grant funded by GIST, Korea.

References

- [1] Kyunjune Baek, Minhyun Lee, and Hyunjung Shim. Psynet: Self-supervised approach to object localization using point symmetric transformation. In *AAAI*, 2020. 1, 2, 4, 6, 7, 8
- [2] Haotian Bai, Ruimao Zhang, Jiong Wang, and Xiang Wan. Weakly supervised object localization via transformer with implicit spatial calibration. In *ECCV*, 2022. 2, 6, 7
- [3] Soufiane Belharbi, Aydin Sarraf, Marco Pedersoli, Ismail Ben Ayed, Luke McCaffrey, and Eric Granger. F-cam: Full resolution class activation maps via guided parametric up-scaling. In *WACV*, 2022. 1, 2
- [4] Rodrigo others Benenson. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019. 6
- [5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 3, 7, 8
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 3
- [7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 7, 8
- [8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3, 8
- [9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 8
- [10] Zhiwei Chen, Changan Wang, Yabiao Wang, Guannan Jiang, Yunhang Shen, Ying Tai, Chengjie Wang, Wei Zhang, and Liujuan Cao. Lctr: On awakening the local continuity of transformer for weakly supervised object localization. In *AAAI*, 2022. 2, 6, 7
- [11] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, 2015. 2
- [12] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, 2019. 1, 2
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3, 5, 7
- [14] Wei Gao, Fang Wan, Xingjia Pan, Zhiliang Peng, Qi Tian, Zhenjun Han, Bolei Zhou, and Qixiang Ye. Ts-cam: Token semantic coupled attention map for weakly supervised object localization. In *ICCV*, 2021. 2, 6, 7
- [15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 3
- [16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. In *NeurIPS*, 2020. 3, 8
- [17] Saurav Gupta, Sourav Lakhota, Abhay Rawat, and Rahul Tallamraju. Vitot: Vision transformer for weakly supervised object localization. In *CVPR*, 2022. 2, 6, 7, 8
- [18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 3
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5, 7
- [20] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *FGVC*, 2011. 6
- [21] Minsong Ki, Youngjung Uh, Junsuk Choe, and Hyeran Byun. Contrastive attention maps for self-supervised colocalization. In *ICCV*, 2021. 1, 2, 4, 6, 7
- [22] Eunji Kim, Siwon Kim, Jungbeom Lee, Hyunwoo Kim, and Sungroh Yoon. Bridging the gap between classification and localization for weakly supervised object localization. In *CVPR*, 2022. 6, 7
- [23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013. 6
- [24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Univ. of Toronto, 2009. 8
- [25] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 1, 2
- [26] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017. 3
- [27] Huifang Li, Yidong Li, Yi Jin, and Tao Wang. Object representation enhancement for self-supervised colocalization. *IJIS*, 37:8277–8290, 2022. 1, 2, 4, 6, 7
- [28] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018. 2
- [29] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE TPAMI*, 33(2):353–367, 2010. 6, 8
- [30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6
- [31] Meng Meng, Tianzhu Zhang, Qi Tian, Yongdong Zhang, and Feng Wu. Foreground activation maps for weakly supervised object localization. In *ICCV*, 2021. 1, 2, 6, 7
- [32] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 3
- [33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1

- [34] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *COLT*, 2001. 3
- [35] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021. 1, 3, 6, 7
- [36] Yukun Su, Guosheng Lin, Yun Hao, Yiwen Cao, Wenjun Wang, and Qingyao Wu. Self-supervised object localization with joint graph partition. In *AAAI*, 2022. 1, 2, 4, 6, 7
- [37] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1, 6
- [38] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 7
- [39] Yangtao Wang et al. Self-supervised transformers for unsupervised object discovery using normalized cut. In *CVPR*, 2022. 1, 3, 6, 7, 8
- [40] Jun Wei, Qin Wang, Zhen Li, Sheng Wang, S Kevin Zhou, and Shuguang Cui. Shallow feature matters for weakly supervised object localization. In *CVPR*, 2021. 2
- [41] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE TIP*, 26(6):2868–2881, 2017. 1, 2
- [42] Xiu-Shen Wei, Chen-Lin Zhang, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Unsupervised object discovery and co-localization by deep descriptor transformation. *PR*, 88:113–126, 2019. 1, 2, 6, 7
- [43] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017. 2
- [44] Pingyu Wu, Wei Zhai, and Yang Cao. Background activation suppression for weakly supervised object localization. In *CVPR*, 2022. 1, 2, 6, 7
- [45] Jinheng Xie, Jianfeng Xiang, Junliang Chen, Xianxu Hou, Xiaodong Zhao, and Linlin Shen. C2am: Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation. In *CVPR*, 2022. 1, 2, 5, 6, 7, 8
- [46] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Rui-Wei Zhao, Tao Zhang, Xuequan Lu, and Shang Gao. Cream: Weakly supervised object localization via class re-activation mapping. In *CVPR*, 2022. 1, 2, 6, 7
- [47] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. In *NeurIPS*, 2018. 2, 3, 5
- [48] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 1
- [49] Chi Zhang, Guankai Li, Guosheng Lin, Qingyao Wu, and Rui Yao. Cyclesegnet: Object co-segmentation with cycle refinement and region correspondence. *IEEE TIP*, 30:5652–5664, 2021. 2
- [50] Chen-Lin Zhang, Yun-Hao Cao, and Jianxin Wu. Rethinking the route towards weakly supervised object localization. In *CVPR*, 2020. 2, 5, 6, 7, 8
- [51] Runsheng Zhang, Yaping Huang, Mengyang Pu, Jian Zhang, Qingji Guan, Qi Zou, and Haibin Ling. Object discovery from a single unlabeled image by mining frequent itemsets with multi-scale features. *IEEE TIP*, 29:8606–8621, 2020. 1, 2, 6, 7
- [52] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018. 1, 2
- [53] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *ECCV*, 2018. 1, 2
- [54] Xiaolin Zhang, Yunchao Wei, and Yi Yang. Inter-image communication for weakly supervised localization. In *ECCV*, 2020. 2
- [55] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Fei Wu. Rethinking localization map: Towards accurate object perception with self-enhancement maps. *arXiv preprint arXiv:2006.05220*, 2020. 2
- [56] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1, 2, 6
- [57] Lei Zhu, Qian Chen, Lujia Jin, Yunfei You, and Yanye Lu. Bagging regional classification activation maps for weakly supervised object localization. In *ECCV*, 2022. 6, 7
- [58] Lei Zhu, Qi She, Qian Chen, Yunfei You, Boyu Wang, and Yanye Lu. Weakly supervised object localization as domain adaptation. In *CVPR*, 2022. 6, 7, 8