Unsupervised Object Localization with Representer Point Selection
- *Supplementary Materials* -

Yeonghwan Song[1] Seokwoo Jang[1] Dina Katabi[2] Jeany Son[1]
[1]AI Graduate School, GIST [2]MIT CSAIL
{yeonghwan.song, jangseokwoo}@gm.gist.ac.kr dina@csail.mit.edu jeany@gist.ac.kr

A. Further Analysis of Threshold τ

In this supplementary section, we offer theoretical and empirical support to determine the threshold value, τ. As defined in Eq. (5) of our original manuscript, $\mathbf{w}^*$, which is used as a foreground predictor, can be rewritten with the sample global importance α from Eq. (9) as follows:

$$\mathbf{w}^* = \sum_{n=1}^{N_D}\sum_{i=1}^{HW} \alpha_{n,i}\hat{\mathbf{f}}_{n,i} = \sum_{i=1}^{N} \alpha_i \hat{\mathbf{f}}_i \tag{13}$$

$$= \frac{1}{C}\sum_{i=1}^{N}(\|\mathbf{f}_i\| - \tau)\hat{\mathbf{f}}_i \tag{14}$$

$$= \frac{1}{C}\underbrace{\sum_{i=1}^{N}\mathbf{f}_i}_{\mathbf{v}} - \frac{\tau}{C}\underbrace{\sum_{i=1}^{N}\hat{\mathbf{f}}_i}_{\mathbf{u}}, \tag{15}$$

where $\mathbf{f}_i$ is a feature vector, $\hat{\mathbf{f}}_i = \frac{\mathbf{f}_i}{||\mathbf{f}_i||}$, $N = N_D HW$ for simplicity and C denotes a constant. In Eq. (15), τ is a threshold that is used to determine soft pseudo labels for the training examples using the norm of the feature vector.

A straightforward approach to determine the threshold is to calculate the expected value of feature vector norms across the training set, given by $\tau = E[||\mathbf{f}_i||] = \sum_i^N ||\mathbf{f}_i||/N$. However, using a uniform probability distribution for expected value calculations does not provide information on the directions and similarities between feature vectors. Therefore, we propose a joint probability distribution that considers the correlations among feature vectors to compute the expected value. Let us denote two independent random variables, X and X', which share the sample space of feature vector norms, and XX' is a joint random variable. To utilize the relationships between all pairs of feature vectors, we employ the cosine similarity to compute the joint probability mass function of XX'. The joint probability mass function of XX' is then expressed as follows:

$$P(X = \|\mathbf{f}_i\|, X' = \|\mathbf{f}_j\|) \propto \hat{\mathbf{f}}_i^\top \hat{\mathbf{f}}_j, \tag{16}$$

and the expectation of XX' is given by

$$E[XX'] = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\|\mathbf{f}_i\|\|\mathbf{f}_j\|P(X = \|\mathbf{f}_i\|, X' = \|\mathbf{f}_j\|)}{\sum_{i'=1}^{N}\sum_{j'=1}^{N} P(X = \|\mathbf{f}'_i\|, X' = \|\mathbf{f}'_j\|)} \tag{17}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} \|\mathbf{f}_i\|\|\mathbf{f}_j\| \frac{\hat{\mathbf{f}}_i^\top \hat{\mathbf{f}}_j}{\sum_{i'=1}^{N}\sum_{j'=1}^{N} \hat{\mathbf{f}}_{i'}^\top \hat{\mathbf{f}}_{j'}}, \tag{18}$$

where $\hat{\mathbf{f}}_i^\top \hat{\mathbf{f}}_j > 0, \forall i, j$, because the last layer of pre-trained encoder $\Phi(\cdot)$ contains a ReLU operation.

Let us denote $\tau = E[X]$, and then the expected value of a jointly distributed discrete random variables of two independent random variables is given by the product of the expected values of two random variables as follows:

$$\tau^2 = (E[X])^2 = E[X]E[X'] = E[XX'] \tag{19}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} \|\mathbf{f}_i\|\|\mathbf{f}_j\| \frac{\hat{\mathbf{f}}_i^\top \hat{\mathbf{f}}_j}{\sum_{i'=1}^{N}\sum_{j'=1}^{N} \hat{\mathbf{f}}_{i'}^\top \hat{\mathbf{f}}_{j'}} \tag{20}$$

$$= \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{f}_i^\top \mathbf{f}_j}{\sum_{i'=1}^{N}\sum_{j'=1}^{N} \hat{\mathbf{f}}_{i'}^\top \hat{\mathbf{f}}_{j'}}, \tag{21}$$

where the denominator and numerator in Eq. (21) can be expressed by $\mathbf{u}$ and $\mathbf{v}$ as in Eq. (15) as follows:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{\mathbf{f}}_i^\top \hat{\mathbf{f}}_j = (\sum_{i=1}^{N} \hat{\mathbf{f}}_i)^\top \sum_{i=1}^{N} \hat{\mathbf{f}}_i \tag{22}$$

$$= \|\sum_{i=1}^{N} \hat{\mathbf{f}}_i\|^2 = \|\mathbf{u}\|^2, \tag{23}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{f}_i^\top \mathbf{f}_j = \|\sum_{i=1}^{N} \mathbf{f}_i\|^2 = \|\mathbf{v}\|^2. \tag{24}$$

Therefore, τ is computed as follows:

$$\tau = \frac{\|\mathbf{v}\|}{\|\mathbf{u}\|}. \tag{25}$$

Table 8. Comparison between our method and several object localization methods that use the additional classifier, EfficientNetB7 [11], in terms of *Top-1*, *Top-5* and *GT-known Loc* on CUB-200-2011 test set and ImageNet-1K validation set. Loc. and Cls. denote the localization and classification backbones, respectively. † indicates MoCo v2 pre-trained backbone.

Method	Loc.	Cls.	$\mathcal{S}$	$\mathcal{T}$	CUB-200-2011			ImageNet-1K		
					Top-1 Loc	*Top-5 Loc*	*GT-Known*	*Top-1 Loc*	*Top-5 Loc*	*GT-Known*
Weakly supervised method										
SPOL '21 [12]	ResNet50	EfficientNetB7	✓	✓	80.12	93.44	96.46	59.14	67.15	69.02
Self-supervised methods										
PSOL '20 [16]	DenseNet161	EfficientNetB7	✗	✓	80.89	89.97	91.78	58.00	65.02	66.28
C^2AM '22 [14]	DenseNet161	EfficientNetB7	✗	✓	81.76	91.11	92.88	59.56	67.05	68.53
w/o finetuning										
Ours	ResNet50†	EfficientNetB7	✗	✗	**84.90**	**94.74**	**96.67**	**60.17**	**67.87**	**69.30**

Table 9. Comparison between the proposed method and the state-of-the-art weakly supervised object localization methods in terms of *MaxBoxAccV2* on CUB-200-2011 and ImageNet-1K.

Methods	Backbone	CUB-200-2011	ImageNet-1K
WSOL methods			
BGC '22 [9]	ResNet50	75.90	68.70
CREAM '22 [15]	ResNet50	73.50	67.40
DAOL '22 [18]	ResNet50	69.87	68.23
BagCAM '22 [17]	ResNet50	84.88	69.97
ViTOL '22 [6]	ViT-S	73.17	70.47
SCM '22 [1]	ViT-S	**89.90**	-
Self-supervised methods			
C^2AM '22 [14]	ResNet50	83.80	66.80
w/o finetuning			
MoCo v2 [3] + Ours	ResNet50	87.26	66.38
DINO [2] + Ours	ViT-S	88.83	**73.04**

B. Additional WSOL Results

Advanced Classifier In compare our method with other weakly supervised object localization methods [12, 16, 14] that utilize more advanced classifiers, such as EfficientNetB7 [11], we also evaluate our method using EfficientNetB7 for classification in a weakly supervised setting. As shown in Table 8, our method outperforms other self-supervised methods which utilize much deeper networks, such as DenseNet161 [8], instead of ResNet50 [7], by significant margins. Furthermore, our method exhibits superior performances compared to the weakly supervised method [12] which relies on explicit class labels for training, while our method and self-supervised methods solely use pre-trained classification networks for classification.

MaxBoxAccv2 In cater various demands for localization accuracy, [4] proposed evaluating WSOL methods through *MaxBoxAccV2*. *MaxBoxAccV2* is calculated by averaging the *MaxBoxAcc* performance across various IoU threshold $\delta \in \{0.3, 0.5, 0.7\}$. As shown in Table 9, our method surpasses other self-supervised and weakly supervised methods on ImageNet. In the CUB-200-2011 dataset as well, our approach achieves performance with negligible differences from the state-of-the-art, independent of the architecture.

Table 10. Comparison of the performance of our method between Moco v2 and supervised pre-trained ResNet50 on UOL setup.

Pre-training	CUB	Cars	Aircraft	Dogs	ImageNet
Supervised	88.45	96.98	98.47	89.53	63.22
MoCo v2	**96.67**	**99.69**	**98.71**	**95.07**	**66.89**

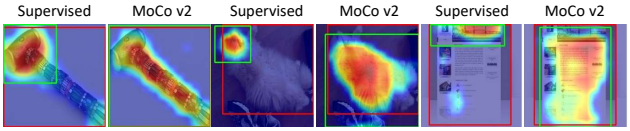

Figure 7. Visualization of activation maps using the supervised and self-supervised (MoCo v2) pre-trained models.

C. Advantage of SSL Pre-trained Backbone

We present the results of both supervised and self-supervised pre-trained models in Table 10. Interestingly, we found that the results of the supervised model were inferior to those of the self-supervised model. This disparity can be linked to a well-established challenge in object localization with class-level supervision [5, 10, 13]. Class-level supervised models often concentrate mainly on the most discriminative parts, as they are trained to learn features that have a substantial impact on classification. In the context of our method, which does not involve fine-tuning the model, this issue becomes more pronounced. To further illustrate this phenomenon, we include examples in Fig. 7. Here, the supervised model activates only the most visually prominent features, while the self-supervised model demonstrates a more comprehensive ability to localize the entire object.

D. More Qualitative Results

We also include further qualitative results to illustrate the operating process of our method for selecting representer points, as depicted in Figure 3 of the main manuscript. As shown in Figure 8, we present examples that highlight global sample importance, the similarity between features, and representer values for given points within an image. These examples reveal that that representer values tend to escalate when both the feature similarity and the importance α of each training example are pronounced. In the visualizations of representer value maps, red regions indicate

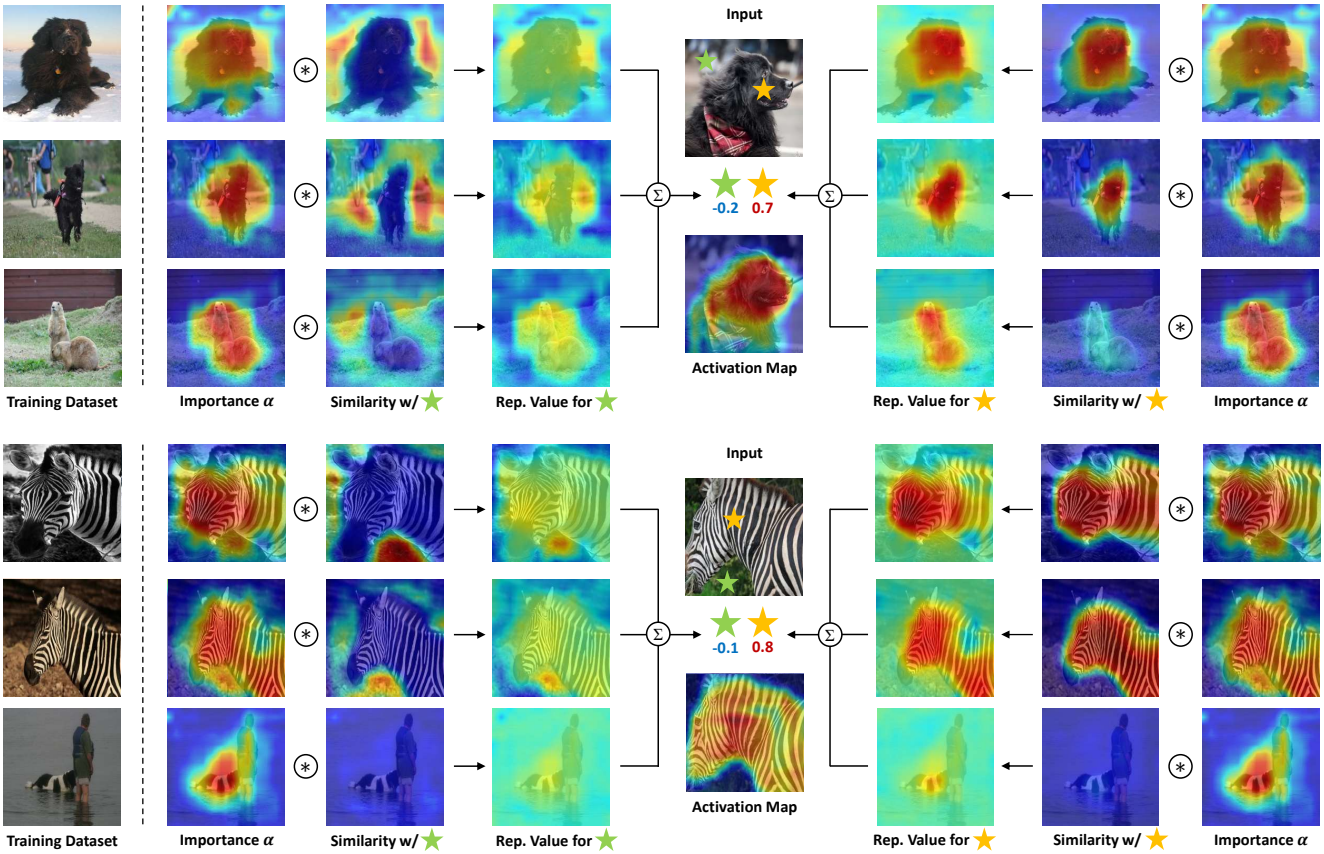

Figure 8. Illustration of how our method computes activation maps using two example points, yellow and green stars. Both importance maps and representer value maps are normalized to be centered at zero, unlike similarity maps' min-max normalization. Hence, red and blue regions denote positive and negative representer points, respectively. Green or yellow colored regions indicate very small absolute values of the representer value.

excitatory points for the foreground prediction, while blue regions indicate inhibitory points. By fostering a comprehensive understanding of model's predictions, it provides valuable insights into the reasoning behind the model's specific predictions and exclusions.

E. Limitations and Social Impacts

Since our method does not rely on ground-truth annotations, which reduces the risk of bias, but it increases the likelihood of errors in object localization when compared to supervised methods. In addition, our method shares common limitations with other unsupervised, self-supervised, or weakly supervised methods, such as difficulties in detecting and recognizing rare, small, or complexly appearing objects including objects with similar textures or shapes and those set against cluttered backgrounds. Additionally, since our approach utilizes training examples, it carries a potential risk of privacy violations if the dataset is not meticulously curated. However, despite these limitations, we believe our method offers a unique advantage: it provides explainability about how it discovers objects. This ability sets our approach apart from other methods and adds to its appeal.

References

[1] Haotian Bai, Ruimao Zhang, Jiong Wang, and Xiang Wan. Weakly supervised object localization via transformer with implicit spatial calibration. In *ECCV*, 2022. 2

[2] 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. 2

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[4] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, 2020. 2

[5] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, 2019. 2

[6] Saurav Gupta, Sourav Lakhotia, Abhay Rawat, and Rahul Tallamraju. Vitol: Vision transformer for weakly supervised object localization. In *CVPR*, 2022. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[9] 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. 2

[10] 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. 2

[11] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2

[12] 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

[13] 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

[14] 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. 2

[15] 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. 2

[16] Chen-Lin Zhang, Yun-Hao Cao, and Jianxin Wu. Rethinking the route towards weakly supervised object localization. In *CVPR*, 2020. 2

[17] Lei Zhu, Qian Chen, Lujia Jin, Yunfei You, and Yanye Lu. Bagging regional classification activation maps for weakly supervised object localization. In *ECCV*, 2022. 2

[18] Lei Zhu, Qi She, Qian Chen, Yunfei You, Boyu Wang, and Yanye Lu. Weakly supervised object localization as domain adaption. In *CVPR*, 2022. 2