Box-based Refinement for Weakly Supervised and Unsupervised Localization Tasks

Eyal Gomel
Tel Aviv University
eyalgomel12@gmail.com

Tal Shaharbany
Tel Aviv University
shaharabany@mail.tau.ac.il

Lior Wolf
Tel Aviv University
wolf@cs.tau.ac.il

Abstract

It has been established that training a box-based detector network can enhance the localization performance of weakly supervised and unsupervised methods. Moreover, we extend this understanding by demonstrating that these detectors can be utilized to improve the original network, paving the way for further advancements. To accomplish this, we train the detectors on top of the network output instead of the image data and apply suitable loss backpropagation. Our findings reveal a significant improvement in phrase grounding for the "what is where by looking" task, as well as various methods of unsupervised object discovery. Our code is available at https://github.com/eyalgomel/box-based-refinement.

1. Introduction

In the task of unsupervised object discovery, one uses clustering methods to find a subset of the image in which the patches are highly similar, while being different from patches in other image locations. The similarity is computed using the embedding provided, e.g., by a transformer f that was trained using a self-supervised loss. The grouping in the embedding space does not guarantee that a single continuous image region will be selected, and often one region out of many is selected, based on some heuristic.

It has been repeatedly shown [47, 58, 5] that by training a detection network, such as faster R-CNN[39], one can improve the object discovery metrics. This subsequent detector has two favorable properties over the primary discovery method: it is bounded to a box shape and shares knowledge across the various samples.

In this work, we show that such a detector can also be used to improve the underlying self-supervised similarity. This is done by training a detector network h not on top of the image features, as was done previously, but on the output map of network f. Once the detector network h is trained, we freeze it and use the same loss that was used to train the detector network to refine the underlying representation of f.

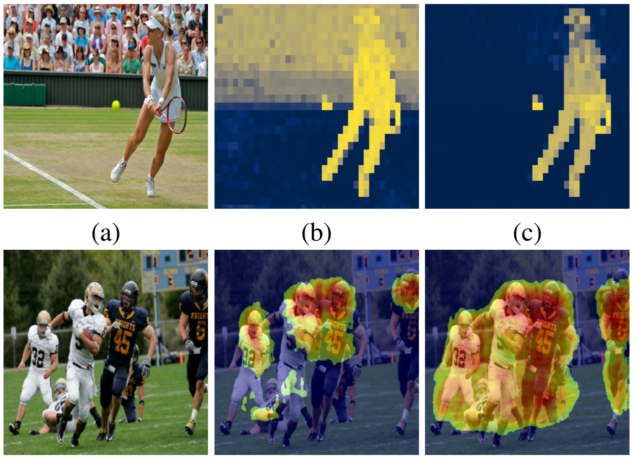

Figure 1. Examples of refining localization networks. The top row depicts an example of unsupervised object discovery. (a) the input image (b) the normalized cut eigenvector using the original DINO [9] network f, as extracted with the TokenCut[58] method. (c) the same eigenvector using the refined DINO network f^h our method produces. The bottom row contains phrase grounding results (d) the original input corresponding to the phrase "two football teams", (e) the localization map using the image-text network g of [42], and (f) the localization map using the refined g^h.

At this point, the detector network serves as a way to link a recovered set of detection boxes to an underlying feature map of f. Without it, deriving a loss would be extremely challenging, since the process used for extracting the detection box from f is typically non-differentiable.

The outcome of this process is a refined network f^h, obtained by fine-tuning f using network h. The finetuned network produces a representation that leads to a spatially coherent grouping of regions, as demonstrated in Fig. 1(a-c).

A similar process is used for the phrase grounding problem. In this case, given a textual phrase, a network g is trained to mark a matching image region. Supervision is performed at the image level, without localization information, a process known as weakly supervised training. In this case, the same loss is used to train a network h on a set of extracted regions, and then to refine g.

Our method exhibits remarkable versatility, as demonstrated through extensive testing on multiple benchmarks, two phrase grounding tasks, and various unsupervised object discovery methods. In all cases, our method consistently achieves significant improvements across all metrics, surpassing the performance of state-of-the-art methods. The move approach introduced trains a detector on the network output rather than the image data. This strategy, distinct from previous work, allows us to refine the primary network independently and further enhance its performance.

2. Related work

Our method is tested on two localization tasks that are not fully supervised: unsupervised object discovery (detection) and phrase grounding. Numerous studies have been introduced in the realm of unsupervised object discovery, alongside akin tasks involving detection and segmentation, using different techniques and methods to discover and localize objects in images (and videos) without requiring explicit object annotations. In particular, deep learning-based approaches have been combined with clustering-based methods [64, 49, 45, 57], generative models [56, 4, 33], and object-level grouping [46, 3]. Two of the methods we build upon in our experiments, LOST [47] and TokenCUT [58], employ clustering methods on top of the DINO network [9], while MOVE [5] uses a segmentation head on top of DINO representation.

In the phrase grounding task, text phrases are associated with specific image locations [62, 26]. When relying on weakly supervised learning, the locations are not given during training, only during test time [1]. A common way to link the phrase to the image is to embed both the text and image patches in a shared embedding space [14, 41, 27]. Recent contributions employ CLIP [38] for linking text with image locations since it has powerful text and image encoders and relies on weakly supervised training [31, 42]. It can, therefore, be used both to represent the text and to obtain a training signal for the phrase grounding network.

We are not aware of other work in which one network f trains another network h, which in turn is used to refine the first network. There are contributions in which two networks are trained symbiotically at the same time. For example, for the task of semi-supervised semantic segmentation, two differently initialized networks were trained jointly, with each network creating pseudo-labels for the other [13]. The DINO unsupervised representation learning method [9] employs a self-distillation process in which the teacher is a combination of frozen student networks.

The role of h in propagating a detection-based loss back to f is reminiscent of other cases in which a network is used for the purpose of supervising another, e.g., GANs [23]. In other cases, an auxiliary network can be trained in a supervised way to provide a differentiable approximation of an indifferentiable black box [35].

3. The Phrase Grounding Method

While we apply the same method for multiple applications, each application relies on a different configuration of baseline networks. Therefore, to minimize confusion, we first focus on phrase grounding. Applying our method to unsupervised object discovery is explored in Sec. 4.

In phrase grounding, we refine a pre-trained localization model (g) using a detection model (h) that we add. h is trained based on g and then the predictions of h, now serving as a teacher, are used to finetune network g, which becomes the student. This cyclic process is illustrated in Fig. 2 and serves to make g more spatially coherent, see Fig. 1(d-f).

The phrase grounding network g is based on an encoder-decoder architecture adapted to support text-based conditioning [42]. The input signals are (i) a text t and (ii) an RGB image $I \in R^{3\times W\times H}$. It outputs a localization heatmap M that identifies image regions in I that correspond to the part of the scene described by t.

$$M = g(I, Z_t(t))\,, \tag{1}$$

where $M \in R^{W\times H}$ contains values between 0 and 1, and $Z_t(t)$ is a text embedding of the input text t, given by the text encoder of CLIP [37]. Our refinement algorithm uses g with the pre-trained weights published by [43].

Our method trains a model h to generate a set of bounding boxes $\bar{B}$ that match the localization map M.

$$\bar{B} = h(M) \tag{2}$$

Thus h provides a feedforward way to generate bounding boxes from M. The alternative provided, for example, by [43] is a multi-step process in which M is first converted to a binary mask by zeroing out any pixel value lower than half the mask's max value [36, 17, 16]. Next, contours are extracted from the binary mask using the method of [51]. For each detected contour, a bounding box is extracted, whose score is given by taking the mean value of M for that bounding box. Finally, a non-maximal suppression is applied over the boxes with an overlap of at least 0.05 IOU, filtering out low-score boxes (0.5 of the maximal score).

h replaces this process with a single feed-forward pass. However, its main goal is to provide a training signal for refining g. This is done by considering the output of h as foreground masks and considering the values of g's output inside and outside these masks.

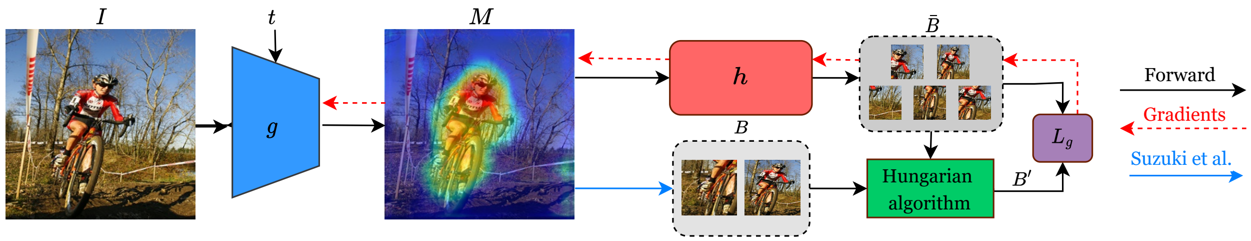

Figure 2. An illustration of our method. The phrased grounding network f is given the input image I and a text phrase t and produces a heatmap M. A heuristic (blue line) then produces a set of bounding boxes B from this map that are used to train a detection network h, which outputs a set of boxes $\bar{B}$. The loss that is used is applied after applying the optimal permutation.

3.1. Training h

The network h is trained to predict a fixed number k of bounding boxes $\bar{B}$. Each box is represented as a vector $b_i \in \mathbb{R}^6$ that contains the center coordinates of the box, its width, and its height. In addition, the network h contains a logit value, which denotes whether there is an expected object within each box.

Training is performed maintaining the semi-supervised nature of the phrase grounding method. The bounding boxes used for training h are extracted using network g and the method of Suzuki et al[51], as explained above. We call the set of resulting bounding boxes B.

Following Carion et al. [8], we train h using a loss L_h that has three terms: (1) a classification loss L_{cls}, (2) an $l1$ loss L_{box}, and (3) the GIoU[40] loss L_{giou}.

If the number of objects k returned by h is smaller than the number of target boxes $|B|$, the k boxes with the highest confidence are used. In the opposite case, B is padded with zero-coordinate vectors with a "no object" label.

For computing the loss, one assumes a one-to-one correspondence between the ground truth objects and the detected boxes. This matching is obtained by minimizing L_h over all possible permutations, using the Hungarian algorithm [30] for minimal cost bipartite matching. Denote as $B' = [b'_0, b'_1, ..., b'_{k-1}]$ the matrix that holds the set of boxes B ordered optimally.

The classification loss L_{cls} is a Negative log-likelihood loss

$$L_{\text{cls}} = \sum_{i=0}^{k-1} -\log \bar{p}_i \tag{3}$$

where $\bar{p}_i$ is the predicted box logit, representing the probability of the existence of an object.

L_{box} is applied directly to the coordinates of the centers of the bounding boxes, their height and width:

$$L_{\text{box}} = \sum_{i=0}^{k-1} \|b'_i - \bar{b}_i\|_1 \tag{4}$$

While the loss L_{box} is affected by the size of the box, the 2nd loss, L_{giou}, is a scale-invariant loss given by

$$L_{\text{giou}}(B', \bar{B}) = \sum_{i=0}^{k-1} 1 - \left(\frac{|\bar{b_i} \cap b'_i|}{|\bar{b_i} \cup b'_i|} - \frac{|c_i \setminus (\bar{b_i} \cup b'_i)|}{|C_i|} \right) \tag{5}$$

where c_i is the smallest box containing b'_i and $\bar{b}_i$. All losses are normalized by the number of boxes.

The final loss is a weighted sum of all three losses:

$$\begin{aligned} L_h(B', \bar{B}) = \lambda_1 * L_{\text{cls}}(B', \bar{B}) + \lambda_2 * L_{\text{box}}(B', \bar{B}) + \\ \lambda_3 * L_{\text{giou}}(B', \bar{B}) \end{aligned} \tag{6}$$

where $\lambda_1 = 2, \lambda_2 = 5, \lambda_3 = 2$. These weights are similar to those used in previous work, with an extra emphasis on λ_1 (using a value of 2 instead of 1), but there was no attempt to optimize them beyond inspecting a few training images.

3.2. Refining g

For finetuning g, we use the multiple loss terms, including the same loss terms that are used for training h, with a modification. Here, instead of just calculating the loss between two sets of boxes, we also compute the union box of ground truth boxes: $BU = Union(B)$. With probability 0.5 we use BU instead of B for calculating the loss (in this case, the matching is done with a single box only)

$$L_{h_{BU}} = \begin{cases} L_h(BU, \bar{B}), & \text{if } p \geq 0.5 \\ L_h(B, \bar{B}), & \text{otherwise} \end{cases}, p \sim \text{Uniform}[0, 1] \tag{7}$$

In addition to the bounding box loss, we use losses for the localization maps used by [43] to train g. This prevents the fine-tuned model from following h "blindly", without considering the underlying data.

The relevancy map loss, uses a CLIP-based relevancy [11] to provide rough estimation for the localization map

$$L_{\text{rmap}}(I, H) = \|H - g^h(I, Z^T)\|^2, \tag{8}$$

where H is the relevancy map and g^h is the refined network g. The foreground loss $L_{fore}(I,T)$ is given by

$$L_{\text{fore}}(I,t) = -CLIP(g^h(I,Z^T) \odot I, t), \quad (9)$$

where $\odot$ is the Hadamard product. The loss maximizes the similarity given by CLIP between the mask's foreground region and the input text t. On the other hand, the background loss $L_{back}(I,t)$ minimizes the similarity CLIP distance between the background and text t

$$L_{back}(I,t) = CLIP((1 - g^h(I,Z^T)) \odot I, t), \quad (10)$$

The overall loss is given by:

$$L_g = L_{h_{BU}} + \lambda_4 * L_{reg}(I,g^h) + \lambda_5 * L_{\text{rmap}}(I,H) + \\ \lambda_6 * L_{\text{back}}(I,T) + \lambda_7 * L_{\text{fore}}(I,T)$$

where $\lambda_4 = 1, \lambda_5 = 64, \lambda_6 = 2, \lambda_7 = 1$. These hyperparameters reflect the values assigned by previous work, multiplied by 4 in order to approximately balance the loss that arises from h with the other loss terms.

Architecture h is a VGG16 [48], pre-trained on the ImageNet[18] dataset. In order to apply it to the single channel heatmap $M \in R^{\times W \times H}$, this input is repeated three times across the channel dimension. The last layer of the classifier is replaced by a linear layer of dimensions $4096 \times (6k)$, k being the number of boxes predicted by h.

4. Unsupervised object discovery

For the task of unsupervised object discovery, a vision transformer f is pretrained in a self-supervised manner, using DINO [9]. It is then used to extract features F from an input image $I \in R^{3\times W\times H}$

$$F = \bar{f}(I) \quad (11)$$

where $\bar{f}$ denotes the latent variables from the transformer f. $F \in R^{d\times N}$, where d is the features dimension and N denotes the number of patches for f. For each patch p, we denoted by $f_p \in R^d$ the associated feature vector. Bounding boxes based on these features are extracted using unsupervised techniques, such as LOST [47], TokenCut [58] or MOVE [5].

LOST builds a patch similarities graph $\mathcal{G}$, with a binary symmetric adjacency matrix $A = (a_{pq})_{1\leq p,q\leq N} \in \{0,1\}^{N\times N}$ where

$$a_{pq} = \begin{cases} 1 & \text{if } f_p^\top f_q \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

An initial seed $p*$ is selected as the patch with the smallest number of connections to other patches.

$$p^* = \operatorname*{arg\,min}_{p\in\{1,\dots,N\}} d_p \quad \text{where} \quad d_p = \sum_{q=1}^{N} a_{pq}. \quad (13)$$

This is based on the assumptions that connectivity implies belonging to the same object, since patch embeddings are similar for the same object, and that each object occupies less area than the background.

Denote the list of a patches with the lowest degree d_p as $\mathcal{D}_a$. LOST then considers the subset of $\mathcal{D}_a$ that is positively correlated, in the embedding space, with p^*

$$\mathcal{S} = \{q \in \mathcal{D}_a | f_q^\top f_{p^*} \geq 0\} \quad (14)$$

This set is then expanded obtaining

$$\mathcal{S}^+ = \{q | \sum_{p\in\mathcal{S}} f_q^\top f_p \geq 0\} \quad (15)$$

We note that in the image itself, the patches of $\mathcal{S}^+$ can be part of multiple separate regions. The method selects the connected component (4-connectivity in the image space) in $\mathcal{S}^+$ that contains the seed p^* as its single discovered object.

TokenCut[58] employs a slightly different adjacency matrix, A, which employs the cosine similarity score between pairs of feature vectors.

$$Ap,q = \begin{cases} 1, & \text{if } \frac{f_p^\top f_q}{\|f_p\|_2 \|f_q\|_2} \geq \tau \\ \epsilon, & \text{else} \end{cases}, \quad (16)$$

where $\tau = 0.2$ and $\epsilon = 1e - 5$.

The normalized cut method [44] is applied to the graph to achieve object discovery. This method clusters all patches into two groups, based on the 2nd smallest eigenvector of the normalized adjacency matrix, and selects the group with the maximal absolute value in this eigenvector. The bounding box of the patches in this group is returned.

MOVE[5], in contradistinction to the preceding two methodologies, employs a segmentation network that is trained atop the latent transformer features denoted as F. The resulting output of this network takes the form of a segmentation map denoted as $M \in R^{W\times H}$. Subsequently, this segmentation map undergoes binarization with a threshold set at 0.5, followed by the detection of connected components [7]. The most sizable bounding box is then selected to correspond to the most extensive connected component.

4.1. Training h and refining f

The training process of detector h follows the details described in Sec. 3.1, with a few minor changes. There is a single ground-truth bounding box B, extracted from an image I by model f using the unsupervised techniques described above. Using the same loss term L_h, h is optimized to minimize $L_h(B, \bar{B})$, where $\bar{B}$ are the k predicted boxes.

To maintain the unsupervised nature of the task, h is initialized with weights from the self-supervised method DINO[9], using a ResNet-50[25] backbone. In the phrase

Method	Backbone	VG trained			MS-COCO trained		
		VG	Flickr	ReferIt	VG	Flickr	ReferIt
Baseline	Random	11.15	27.24	24.30	11.15	27.24	24.30
Baseline	Center	20.55	47.40	30.30	20.55	47.40	30.30
GAE [10]	CLIP	54.72	72.47	56.76	54.72	72.47	56.76
FCVC [22]	VGG	-	-	-	14.03	29.03	33.52
VGLS [61]	VGG	-	-	-	24.40	-	-
TD [62]	Inception-2	19.31	42.40	31.97	-	-	-
SSS [26]	VGG	30.03	49.10	39.98	-	-	-
MG [1]	BiLSTM+VGG	50.18	57.91	62.76	46.99	53.29	47.89
MG [1]	ELMo+VGG	48.76	60.08	60.01	47.94	61.66	47.52
GbS [2]	VGG	53.40	70.48	59.44	52.00	72.60	56.10
WWbL [43]	CLIP+VGG	62.31	75.63	65.95	59.09	75.43	61.03
Ours	CLIP+VGG	**63.51**	**78.32**	**67.33**	**60.05**	**77.19**	**63.48**

Table 1. Phrase grounding results: "pointing game" accuracy on Visual Genome (VG), Flickr30K, and ReferIt. The methods in the first three rows do not train.

grounding case and MOVE [5], the input of h is the map M, and the analogue for non-trainable unsupervised object discovery is the map F where such map M is missing.

For refining the DINO-trained transformer model f, we use the same loss term L_h as is used in phrase grounding and add loss terms to prevent it from diverging too far. While in phrase grounding we used the loss terms that were used to train the phrase grounding network, here, for runtime considerations, we explicitly keep the transformer f in the vicinity of the DINO-pretrained network.

The loss term is defined as the distance between the output of f and that of the refined model f^h

$$L_f(I) = \|f(I) - f^h(I)\|^2, \qquad (17)$$

Both methods [47, 58] are improved by training a Class Agnostic Detector (CAD) on the extracted bounding boxes. Faster R-CNN [39] is used for CAD, with the *R50-C4* model of Detectron2 [60] based on a ResNet-50[25] backbone. This backbone is pre-trained with DINO self-supervision. Following this process, we train an identical CAD using the refined model f^h. Note that CAD and our method are complementary. While both train with the same pseudo-labels, CAD is trained on the original image and cannot backpropagate a loss to the underlying network f.

5. Experiments

We present our results for three tasks: weakly supervised phrase grounding (WSPG), "what is were by looking" (WWbL), and unsupervised single object discovery. The first two use the same phrase grounding network g, and the third one is based on one of two techniques, which both utilize the same pre-trained transformer f.

Datasets For WSPG and WWbL, the network g is trained on either MSCOCO 2014 [32] or the Visual Genome (VG) dataset [29]. Evaluation is carried out on the test splits of Flickr30k[34], ReferIt[12, 24] and VG [29].

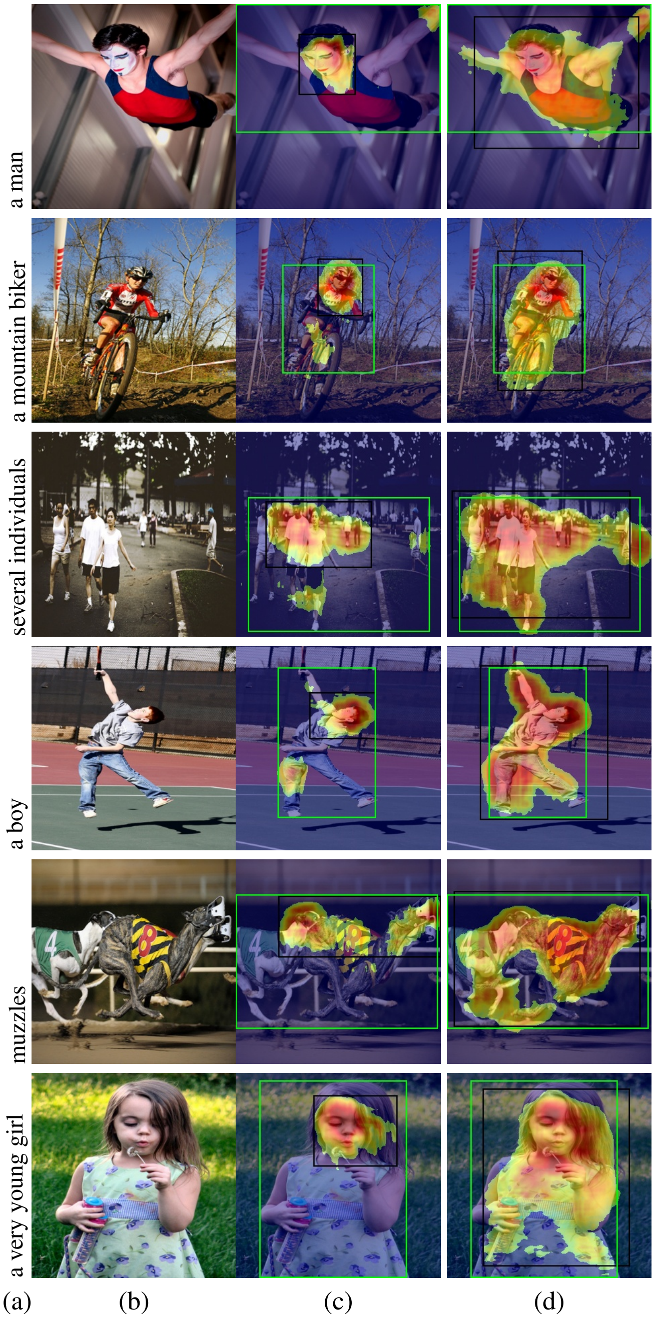

Figure 3. Sample phrase-grounding results. where (a) the phrase (b) the input image (c) results (black) for network g [43] compared to ground-truth box (green) (d) same for refined network g^h.

VG contains 77,398, 5,000, and 5000 training, validation, and test images, respectively. Each image is linked to natural-language text and annotated bounding boxes. During the training of MSCOCO2014 we use the training split defined by Akbari et al. [1]. It consists of 82,783 training samples and 40,504 validation samples, where each sam-

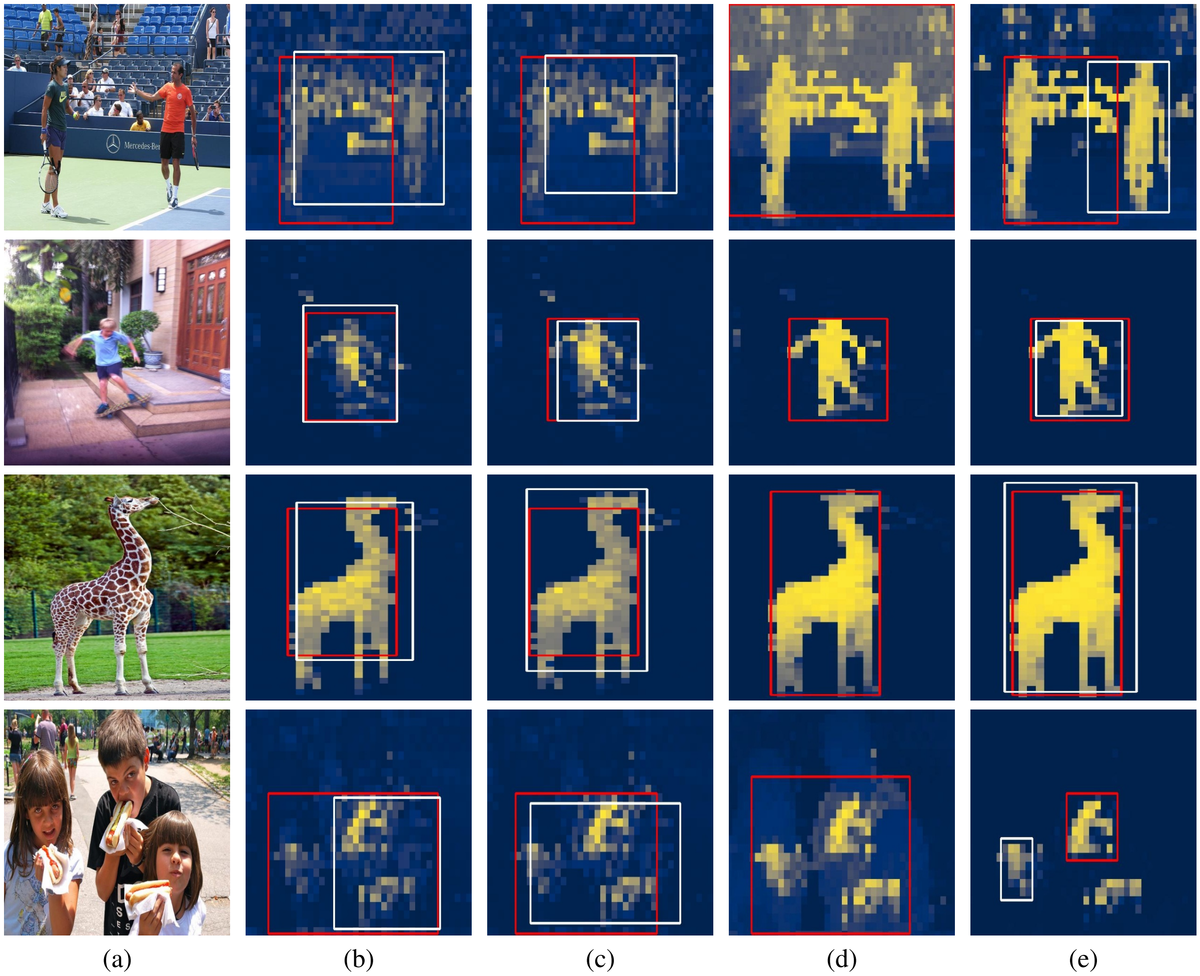

Figure 4. Single object discovery results. (a) the input image, (b) the inverse degree of the LOST [47] graph obtained over f (published model); the red bounding box is directly from LOST, the white is the prediction of CAD trained on top of it (c) same with our refined model f^h and LOST (d) same as b, but using f together with TokenCut[58], (using the published weights; the CAD model was not released and is not shown) (e) the results of f^h and TokenCut.

Training	Model	Test Bbox Accuracy		
		VG	Flickr	ReferIt
MS-COCO	MG [1]	15.77	27.06	15.15
	WWbL [43]	27.22	35.75	30.08
	Ours	**28.77**(27.1)	**47.26**(45.01)	**30.63**(29.05)
VG	MG [1]	14.45	27.78	18.85
	WWbL [43]	27.26	36.35	32.25
	Ours	**31.02**(29.23)	**42.40**(44.91)	**35.56**(34.56)

Table 2. Phrase grounding results: bounding box accuracy on Visual Genome (VG), Flickr30K, and ReferIt. The outcomes obtained from network h are presented within brackets.

Train set	Model	Test point Accuracy			Test Bbox Accuracy		
		VG	Flickr	ReferIt	VG	Flickr	ReferIt
COCO	MG [1]	32.91	50.154	36.34	11.48	23.75	13.31
	WWbL [43]	44.20	61.38	43.77	17.76	32.44	21.76
	Ours	**46.29**	**63.43**	**44.59**	**22.32**	**38.00**	**22.91**
VG	MG [1]	32.15	49.48	38.06	12.23	24.79	16.43
	WWbL [43]	43.91	58.59	44.89	17.77	31.46	18.89
	Ours	**46.77**	**61.75**	**44.9**	**22.40**	**35.23**	**23.44**

Table 3. WWbL results: bounding box accuracy on Visual Genome (VG), Flickr30K, and ReferIt.

ple contains an image and five captions describing the image. ReferIt[12, 24] consists of 130k expressions referring to 99,535 objects in 20k images. For evaluation, we use the test split of Akbari et al.[1]. The dataset Flickr30k Entities [34] consists of 224K phrases that depict objects present in more than 31K images, with each image having five corresponding captions. The evaluation is carried out on a the test split of Akbari et al.[1]. For unsupervised single object discovery, the network g is trained on either MSCOCO

Model	VOC07	VOC12	MS-COCO
Selective Search [52]	18.8	20.9	16.0
EdgeBoxes [65]	31.1	31.6	28.8
Kim et al. [28]	43.9	46.4	35.1
Zhang et al. [63]	46.2	50.5	34.8
DDT+ [59]	50.2	53.1	38.2
rOSD [54]	54.5	55.3	48.5
LOD [55]	53.6	55.1	48.5
DINO-seg [9]	45.8	46.2	42.1
LOST [47]	61.9	64.0	50.7
Ours using LOST	62.0(42.1)	66.2(53.5)	52.0(33.7)
TokenCut [58]	68.8	72.1	58.8
Ours using TokenCut	69.0(44.6)	72.4(54.1)	60.7(39.5)
MOVE [5]	76.0	78.8	66.6
Ours using MOVE	**77.5**(42.9)	**79.6**(54.9)	**67.2**(48.3)
LOD + CAD [47]	56.3	61.6	52.7
rOSD + CAD [47]	58.3	62.3	53.0
LOST + CAD [47]	65.7	70.4	57.5
Ours using LOST + CAD	66.1	71.0	58.7
TokenCut [58] +CAD	71.4	75.3	62.6
Ours using TokenCut + CAD	71.9	75.6	64.4
MOVE [5] +CAD	77.1	80.3	69.1
Ours using MOVE [5] +CAD	**78.7**	**81.3**	**69.3**

Table 4. Object Discovery results: CorLoc score on MS-COCO20K, VOC07 and VOC12. Network h was trained using pseudo labels from either LOST [47], TokenCut [58] or MOVE [5]. +CAD indicates training a second-phase class-agnostic detector with model pseudo-boxes as labels. Network h results are enclosed in brackets.

Ablation	Test point Accuracy			Test Bbox Accuracy		
	VG	Flickr	ReferIt	VG	Flickr	ReferIt
w/o Box Union	57.26	72.54	62.55	25.11	28.74	24.63
w/o reg.	53.49	68.47	61.92	26.45	42.79	29.74
k=1	56.84	70.74	62.15	27.75	32.35	24.73
Ours	**60.05**	**77.19**	**63.48**	**28.77**	**47.26**	**30.63**

Table 5. Ablation study for the phrase grounding task. See text for details. All models were trained on MS-COCO14[32] dataset

20K, PASCAL-VOC07[20] or PASCAL-VOC12[21]. MS-COCO20K has 19,817 images chosen at random from the MSCOCO 2014 dataset[32]. VOC07 and VOC12 contain 5,011 and 11,540 images respectively, with each image belonging to one of 20 categories. For evaluation, we follow common practice and evaluate the train/val datasets. This evaluation is possible since the task is fully unsupervised.

Implementation details For phrase grounding tasks, the proposed network h backbone is VGG16 [48], pre-trained on the ImageNet[18] dataset. For the object discovery task, we use h with ResNet-50[25] backbone, pre-trained with DINO[9] self-supervision on the ImageNet[18] dataset. For both tasks, h predicts $k = 10$ bounding boxes. Refining takes place using an Adam optimizer with a batch size of 36. The learning rate of h is 1e-5, while the learning rates of g^h and f^h are 1e-7 and 5e-7, respectively. The optimizer weight decay regularization is 1e-4. For the first 3000 iterations, network h is optimized, where g^h/f^h is fixed. Then, for the rest of the training (10k iterations), h is fixed while g^h/f^h is optimized.

Ablation	VOC07	VOC12	MSCOCO20K
w/o reg.	61.72	64.45	50.13
k=1	**62.54**	64.67	**52.00**
k=5	62.16	64.45	51.70
k=10	61.92	**66.16**	51.98
k=15	61.44	64.46	50.60

Table 6. Ablation study for the object discovery task.

Metrics Phrase grounding tasks are evaluated with respect to the accuracy of the pointing game[62], which is calculated based on the output map by finding the location of the maximum value, given a query, and checking whether this point falls within the object's region.

The "BBox accuracy" metric extracts a bounding box, given an output mask, and compares it with the ground-truth annotations. A prediction is considered accurate if IOU between the boxes is larger than 0.5. To extract the bounding box from an output map M, the procedure of Shaharabany et al. [43] is employed. First, M is binarized using a threshold of 0.5, then contours are extracted from M using the method of Suzuki et al. [51]. Based on the contours, a set of bounding boxes is derived by taking the smallest box containing each contour. These bounding boxes are scored by summing the values of M within the contour while ignoring boxes with low scores. Next, a non-maximal suppression process is applied and the minimal bounding box that contains the remaining bounding boxes is chosen.

The WWbL task is an open-world localization task, with only an image as input (no text input). Using this image, the goal is to both localize and describe all of the elements in the scene. To solve this task, a multi-stage algorithm was introduced by Shaharabany et al. [43], starting with obtaining object proposals using selective search [52]. Next, BLIP is used to caption these regions. Captions that are similar to each other are removed using the Community Detection (Cd) clustering method [6]. Using the learned phrase grounding model g, heatmaps are generated according to the extracted captions.

Similarly to the phrase grounding task, the WWbL task is evaluated using the same two metrics: pointing game accuracy and bounding box accuracy). For each ground-truth pair of bounding box and caption, the closest caption in CLIP space is selected from the list of automatically generated captions. The associated output map of the phrase grounding method is then compared to the ground truth bounding box using the pointing accuracy metric. In addition, bounding boxes are extracted for the output heatmaps M, as described above.

For single object discovery we use the Correct Localization (CorLoc) metric as used by [19, 54, 55, 53, 59, 15, 50]. A predicted bounding box is considered as correct if the IOU score between the predicted bounding box and one of the ground truth bounding boxes is above 0.5. We evaluate our model on the same datasets as [58, 47, 5].

Results Tab. 1 lists the results for Flickr30k, ReferIt, and VG for the weakly-supervised phrase grounding task. Evidently, our method is superior to all baselines, whether training takes place over VG or MS-COCO. In addition to the pointing game results, Tab. 2 presents bounding box accuracy for the phrase grounding task (this data is not available for most baselines). Here, too, our method outperforms the baseline methods by a wide margin.

Phrase grounding samples are provided in Fig. 3, comparing the results after the refinement process (those with g^h) to the results of the baseline g. As can be seen, our method encourages the localization maps to match the typical shape of image objects. As a result, the predicted bounding box after refining the model is often closer to the actual objects in the image.

The WWbL results are listed in Tab. 3, which depicts the performance obtained by our g^h, WWbL [43], and a baseline that employs the phrase grounding method MG [1] as part of the WWbL captioning procedure described above. Out of the three models, our refined model g^h achieves the best scores, for all benchmarks and both metrics.

Tab. 4 summarize the results on the VOC07, VOC12, and MS-COCO20K datasets for the single object discovery task. When utilizing the MOVE [5] model, our method achieves superior performance compared to all other models across all datasets. This superiority holds true when comparing all methods without CAD and when comparing all methods with CAD. Furthermore, our method consistently outperforms other approaches when refining the DINO model f using both TokenCut [58] boxes and LOST [47] boxes on all datasets.

Fig. 4 depicts typical samples of our results for the unsupervised object discovery task, when combining our method with either LOST [47] or TokenCut [58]. Evidently, our refining process improves object and background separation and produces a denser output mask, which covers the object more completely. Furthermore, the extracted bounding boxes become more accurate.

Ablation study In order to validate the individual components of our approach, we conducted an ablation study.

For the phrase grounding task, this study is reported in Tab. 5. The first ablation replaces the loss $L_{h_{BU}}$ with the loss L_h, i.e., no union of the detection boxes is performed.

Network	Phrase Grounding	Object discovery	
		LOST	TokenCut
f or g	28 x [4]	72.6 x [16]	72.6 x [16]
h	0.5 x [1]	0.5 x [1]	2.5 x [1]
f^h or g^h	3.2 x [4]	5.3 x [1]	20.5 x [1]

Table 7. Training time (hours) for phrase grounding and unsupervised object discovery. Within brackets is the number of GPUS used during training.

The second ablation employs only the loss of h, $L_{h_{BU}}$, and disregards the loss terms that were used to train network g. The third ablation employs a single detection box ($k = 1$) instead of the default of $k = 10$. As can be seen, these three variants reduce performance across all metrics and datasets. The exact reduction in performance varies across the datasets.

To extensively explore the task of unsupervised object discovery, we conducted a comprehensive ablation study by varying multiple values of k, see Tab. 4.1. This ablation was performed using LOST, which is quicker than TokenCut and without the extra overhead of training CAD. Evidently, removing the regularization term, leaving only the loss L_h (there is no box union in this task, since both LOST and TokenCut return a single box) hurts performance. However, as can be expected, using $k = 1$, instead of the value of $k = 10$ that is used throughout our experiments, better fits this scenario and leads to better performance on VOC07 (and virtually the same on MSCOCO20K).

Training time The time it takes to train our method on medium-sized datasets is reported in Tab. 7. For both original networks, f and g, we use pretrained networks and report the published values. Training h, f^h, g^h reflects our runs on GeForce RTX 2080Ti GPUs (f which is DINO, was trained on much more involved hardware, while g was trained on similar hardware). As can be seen, training h and refining f or g to obtain f^h or g^h is much quicker than the training of the f and g baselines. The difference in training time between LOST and TokenCut stems from the inference done during training, which is much quicker for LOST.

6. Conclusions

We present a novel method, in which a primary network is used in a symbiotic manner with a detection network. The first network is used to extract a feature map and detection boxes, which are used as the input and output of the second. The second network is then used to allow the first network to be refined using the boxes extracted from its output. All training phases are performed on the same training set, within the bounds of the allowed level of supervision. Tested on a wide variety of tasks and benchmarks, the proposed method consistently improves localization accuracy.

References

[1] Hassan Akbari, Svebor Karaman, Surabhi Bhargava, Brian Chen, Carl Vondrick, and Shih-Fu Chang. Multi-level multimodal common semantic space for image-phrase grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12476–12486, 2019. 2, 5, 6, 8

[2] Assaf Arbelle, Sivan Doveh, Amit Alfassy, Joseph Shtok, Guy Lev, Eli Schwartz, Hilde Kuehne, Hila Barak Levi, Prasanna Sattigeri, Rameswar Panda, et al. Detector-free weakly supervised grounding by separation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1801–1812, 2021. 5

[3] Yutong Bai, Xinlei Chen, Alexander Kirillov, Alan Yuille, and Alexander C Berg. Point-level region contrast for object detection pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16061–16070, 2022. 2

[4] Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[5] Adam Bielski and Paolo Favaro. Move: Unsupervised movable object segmentation and detection. *arXiv preprint arXiv:2210.07920*, 2022. 1, 2, 4, 5, 7, 8

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008. 7

[7] Federico Bolelli, Stefano Allegretti, Lorenzo Baraldi, and Costantino Grana. Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling. *IEEE Transactions on Image Processing*, PP:1–1, 10 2019. 4

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 213–229, 2020. 3

[9] 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. 1, 2, 4, 7

[10] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, October 2021. 5

[11] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021. 3

[12] Kan Chen, Rama Kovvuri, and Ram Nevatia. Query-guided regression network with context policy for phrase grounding. In *ICCV*, 2017. 5, 6

[13] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021. 2

[14] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4:357–370, 2016. 2

[15] 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, 2015. 8

[16] Junsuk Choe, Dongyoon Han, Sangdoo Yun, Jung-Woo Ha, Seong Joon Oh, and Hyunjung Shim. Region-based dropout with attention prior for weakly supervised object localization. *Pattern Recognition*, 116:107949, 2021. 2

[17] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3133–3142, 2020. 2

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 4, 7

[19] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 452–466, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 8

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results. pascal-network.org/challenges/VOC/voc2007. 7

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012. 7

[22] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015. 5

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[24] Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International workshop ontoImage*, volume 2, 2006. 5, 6

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4, 5, 7

[26] Syed Ashar Javed, Shreyas Saxena, and Vineet Gandhi. Learning unsupervised visual grounding through semantic self-supervision. *arXiv preprint arXiv:1803.06506*, 2018. 2, 5

[27] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. 2

[28] Gunhee Kim and Antonio Torralba. Unsupervised detection of regions of interest using iterative link analysis. *Advances in neural information processing systems*, 22, 2009. 7

[29] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 5

[30] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. 3

[31] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. *arXiv preprint arXiv:2112.03857*, 2021. 2

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, volume 8693 of *LNCS*, pages 740–755, 2014. 5, 7

[33] Lanlan Liu, Michael Muelly, Jia Deng, Tomas Pfister, and Li-Jia Li. Generative modeling for small-data object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6073–6081, 2019. 2

[34] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 5, 6

[35] Adam Polyak, Yaniv Taigman, and Lior Wolf. Unsupervised generation of free-form and parameterized avatars. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):444–459, 2018. 2

[36] Zhenyue Qin, Dongwoo Kim, and Tom Gedeon. Rethinking softmax with cross-entropy: Neural network classifier as mutual information estimator. *arXiv preprint arXiv:1911.10688*, 2019. 2

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2

[38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 1, 5

[40] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[41] Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021. 2

[42] Tal Shaharabany, Yoad Tewel, and Lior Wolf. What is where by looking: Weakly-supervised open-world phrase-grounding without text inputs. *arXiv preprint arXiv:2206.09358*, 2022. 1, 2

[43] Tal Shaharabany, Yoad Tewel, and Lior Wolf. What is where by looking: Weakly-supervised open-world phrase-grounding without text inputs. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2, 3, 5, 6, 7, 8

[44] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 4

[45] Gyungin Shin, Samuel Albanie, and Weidi Wie. Unsupervised salient object detection with spectral cluster voting. *arXiv preprint arXiv:2203.12614*, 2022. 2

[46] Gyungin Shin, Weidi Xie, and Samuel Albanie. Namedmask: Distilling segmenters from complementary foundation models. In *CVPRW*, 2023. 2

[47] 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 *Proceedings of the British Machine Vision Conference (BMVC)*, November 2021. 1, 2, 4, 5, 6, 7, 8

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4, 7

[49] Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonin Vobecky, Éloi Zablocki, and Patrick Pérez. Unsupervised object localization: Observing the background to discover objects, 2023. 2

[50] Parthipan Siva, Chris Russell, Tao Xiang, and Lourdes Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3238–3245, 2013. 8

[51] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985. 2, 3, 7

[52] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013. 7

[53] Huy V. Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Perez, and Jean Ponce. Unsupervised image matching and object discovery as optimization, 2019. 8

[54] Huy V. Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections, 2020. 7, 8

[55] Huy V. Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery, 2021. 7, 8

[56] Andrey Voynov, Stanislav Morozov, and Artem Babenko. Object segmentation without labels with large-scale generative models. In *International Conference on Machine Learning*, pages 10596–10606. PMLR, 2021. 2

[57] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3124–3134, 2023. 2

[58] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 4, 5, 6, 7, 8

[59] Xiu-Shen Wei, Chen-Lin Zhang, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Unsupervised object discovery and co-localization by deep descriptor transformation. *Pattern Recognition*, 88:113–126, 2019. 7, 8

[60] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5

[61] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5945–5954, 2017. 5

[62] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. 2, 5, 7

[63] Tianshu Zhang, Buzhen Huang, and Yangang Wang. Object-occluded human shape and pose estimation from a single color image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7376–7385, 2020. 7

[64] Xiao Zhang, Yixiao Ge, Yu Qiao, and Hongsheng Li. Refining pseudo labels with clustering consensus over generations for unsupervised object re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3436–3445, 2021. 2

[65] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014. 7